System.IO.Compression Library Reference

January 5, 2016

Contents

Description ii

Namespaces iii

Description

The Compression library contains streams that provide compression and decompression services. The library is implemented using **zlib** (`http://wwww.zlib.net`) and **bzip2** (`http://www.bzip.org/`) libraries.

Namespaces

Namespace	Description
System.IO.Compression	The Compression namespace contains stream classes that provide data compression and decompression.

1 Usage

1.0.1 Referencing the System.IO.Compression library

Right-click a project node in IDE | Project References... | Add System Extension Library Reference... | enable *System.IO.Compression* check box

or add following line to your project's .cmp file:

```
reference <ext/System.IO.Compression/System.IO.Compression.cml>;
```

2 System.IO.Compression Namespace

The **Compression** namespace contains stream classes that provide data compression and decompression.

2.1 Classes

Class	Description
BZip2Exception	Exception class thrown when BZip2Stream cannot compress or decompress data.
BZip2Stream	A stream class that writes data to the underlying byte stream in bzip2 compression format, or reads data compressed in bzip2 format from the underlying byte stream and decompresses it.
DeflateException	Exception class thrown when DeflateStream cannot compress or decompress data.
DeflateStream	A stream class that writes data to the underlying byte stream in ZLIB compression format, or reads data compressed in ZLIB format from the underlying byte stream and decompresses it.

2.1.1 BZip2Exception Class

Exception class thrown when BZip2Stream cannot compress or decompress data.

Syntax

```
public class BZip2Exception;
```

Base Class

System.Exception

2.1.1.1 Member Functions

Member Function	Description
BZip2Exception()	Default constructor.
BZip2Exception(const System.IO.Compression.BZip2Exception&)	Copy constructor.
operator=(const System.IO.Compression.BZip2Exception&)	Copy assignment.
BZip2Exception(System.IO.Compression.BZip2Exception&&)	Move constructor.
operator=(System.IO.Compression.BZip2Exception&&)	Move assignment.
BZip2Exception(const System.String&, int)	Constructor. Initializes the **BZip2Exception** with the given error message and error code.
ErrorCode() const	Returns the error code returned by the bzip2 library.
~BZip2Exception()	Destructor.

BZip2Exception() Member Function

Default constructor.

Syntax

```
public BZip2Exception();
```

BZip2Exception(const System.IO.Compression.BZip2Exception&) Member Function

Copy constructor.

Syntax

```
public BZip2Exception(const System.IO.Compression.BZip2Exception& that);
```

Parameters

Name	Type	Description
that	const System.IO.Compression.BZip2Exception&	Argument to copy.

operator=(const System.IO.Compression.BZip2Exception&) Member Function

Copy assignment.

Syntax

```
public void operator=(const System.IO.Compression.BZip2Exception& that);
```

Parameters

Name	Type	Description
that	const System.IO.Compression.BZip2Exception&	Argument to assign.

BZip2Exception(System.IO.Compression.BZip2Exception&&) Member Function

Move constructor.

Syntax

```
public BZip2Exception(System.IO.Compression.BZip2Exception&& that);
```

Parameters

Name	Type	Description
that	System.IO.Compression.BZip2Exception&&	Argument to move from.

operator=(System.IO.Compression.BZip2Exception&&) Member Function

Move assignment.

Syntax

```
public void operator=(System.IO.Compression.BZip2Exception&& that);
```

Parameters

Name	Type	Description
that	System.IO.Compression.BZip2Exception&&	Argument to assign from.

BZip2Exception(const System.String&, int) Member Function

Constructor. Initializes the BZip2Exception with the given error message and error code.

Syntax

```
public BZip2Exception(const System.String& message_, int errorCode_);
```

Parameters

Name	Type	Description
message_	const System.String&	Error message.
errorCode_	int	Error code returned by bzip2 library.

ErrorCode() const Member Function

Returns the error code returned by the bzip2 library.

Syntax

```
public int ErrorCode() const;
```

Returns

int

Returns the error code returned by the bzip2 library.

~BZip2Exception() Member Function

Destructor.

Syntax

```
public ~BZip2Exception();
```

2.1.2 BZip2Stream Class

A stream class that writes data to the underlying byte stream in bzip2 compression format, or reads data compressed in bzip2 format from the underlying byte stream and decompresses it.

Syntax

`public class BZip2Stream;`

Base Class

System.IO.ByteStream

2.1.2.1 Example

```
using System;
using System.IO;
using System.IO.Compression;

int main()
{
    try
    {
        {
            FileByteStream in("bzip2/test.file", FileMode.open);
            FileByteStream out("bzip2/test.bz2", FileMode.create);
            BZip2Stream compressStream(out, CompressionMode.compress);
            in.CopyTo(compressStream);
        }
        {
            FileByteStream in("bzip2/test.bz2", FileMode.open);
            BZip2Stream decompressStream(in, CompressionMode.decompress);
            FileByteStream out("bzip2/test.decompressed", FileMode.create);
            decompressStream.CopyTo(out);
        }
    }
    catch (const Exception& ex)
    {
        Console.Error() << ex.ToString() << endl();
        return 1;
    }
    return 0;
}
```

2.1.2.2 Member Functions

Member Function	Description
BZip2Stream()	Default constructor.
BZip2Stream(System.IO.ByteStream&, System.IO.Compression.CompressionMode)	Constructor. Initializes the **BZip2Stream** class with the given compression mode and underlying byte stream.

BZip2Stream(System.IO.ByteStream&, System.IO.Compression.CompressionMode, int)	Constructor. Initializes the **BZip2Stream** class with the given compression mode, buffer size and underlying byte stream.
BZip2Stream(System.IO.ByteStream&, int)	Constructor. Initializes the **BZip2Stream** class using compression mode compress and given compression level.
BZip2Stream(System.IO.ByteStream&, int, int)	Constructor. Initializes the **BZip2Stream** class using compression mode compress, given compression level and given compression work factor.
BZip2Stream(System.IO.ByteStream&, int, int, int)	Constructor. Initializes the **BZip2Stream** class using compression mode compress, given compression level, given compression work factor and given buffer size.
Mode() const	Returns the compression mode.
Read(byte*, int)	Reads compressed data from the underlying byte stream and decompresses it to the given buffer. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.
ReadByte()	Reads compressed data from the underlying byte stream, decompresses it to an internal buffer and returns one byte of decompressed data. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.
Write(byte)	Writes one byte of data to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.
Write(byte*, int)	Writes given number of bytes from the given buffer to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.

~BZip2Stream()

Destructor. If the compression mode is compress compresses the rest of the data and writes it to the underlying byte stream. Releases occupied memory.

BZip2Stream() Member Function

Default constructor.

Syntax

```
public BZip2Stream();
```

BZip2Stream(System.IO.ByteStream&, System.IO.Compression.CompressionMode) Member Function

Constructor. Initializes the BZip2Stream class with the given compression mode and underlying byte stream.

Syntax

```
public BZip2Stream(System.IO.ByteStream& underlyingStream_, System.IO.Compression.CompressionMod
mode_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write to or read from.
mode_	System.IO.Compression.CompressionMode	Compression mode. Can be compress or decompress. When the compression mode is compress the stream supports writing, when the compression mode is decompress the stream supports reading.

Remarks

When the compression mode is compress, the stream uses default compression level defaultBZip2CompressionLevel, default work factor defaultBZip2WorkFactor and default buffer size 16K for internal input and output buffers.

BZip2Stream(System.IO.ByteStream&, System.IO.Compression.CompressionMode, int) Member Function

Constructor. Initializes the BZip2Stream class with the given compression mode, buffer size and underlying byte stream.

Syntax

```
public BZip2Stream(System.IO.ByteStream& underlyingStream_, System.IO.Compression.CompressionMod
mode_, int bufferSize_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write to or read from.
mode_	System.IO.Compression.CompressionMode	Compression mode. Can be compress or decompress. When the compression mode is compress the stream supports writing, when the compression mode is decompress the stream supports reading.
bufferSize_	int	Buffer size for internal input and output buffers.

Remarks

When the compression mode is compress, the stream uses default compression level defaultBZip2CompressionLevel and default work factor defaultBZip2WorkFactor.

BZip2Stream(System.IO.ByteStream&, int) Member Function

Constructor. Initializes the BZip2Stream class using compression mode compress and given compression level.

Syntax

```
public BZip2Stream(System.IO.ByteStream& underlyingStream_, int compressionLevel_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write the compressed data to.
compressionLevel_	int	Compression level. Can be in range 1..9. Compression level N sets bzip2 block size to N * 100K. Compression level 1 gives least compression and uses minimum memory. Compression level 9 gives the best compression but takes most memory.

Remarks

Sets the work factor defaultBZip2WorkFactor and uses the default buffer size 16K for internal input and output buffers.

BZip2Stream(System.IO.ByteStream&, int, int) Member Function

Constructor. Initializes the BZip2Stream class using compression mode compress, given compression level and given compression work factor.

Syntax

```
public BZip2Stream(System.IO.ByteStream& underlyingStream_, int compressionLevel_,
int compressionWorkFactor_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write the compressed data to.
compressionLevel_	int	Compression level. Can be in range 1 to 9 inclusive. Compression level N sets bzip2 block size to N * 100K. Compression level 1 gives least compression and uses minimal memory. Compression level 9 gives the best compression but takes most memory.
compressionWorkFactor_	int	Compression work factor. Can be in range 0 to 250 inclusive. Setting the work factor to 0 uses the default work factor 30. See http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.html for details.

Remarks

Uses the default buffer size 16K for internal input and output buffers.

BZip2Stream(System.IO.ByteStream&, int, int, int) Member Function

Constructor. Initializes the BZip2Stream class using compression mode compress, given compression level, given compression work factor and given buffer size.

Syntax

```
public BZip2Stream(System.IO.ByteStream& underlyingStream_, int compressionLevel_,
int compressionWorkFactor_, int bufferSize_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write the compressed data to.
compressionLevel_	int	Compression level. Can be in range 1 to 9 inclusive. Compression level N sets bzip2 block size to N * 100K. Compression level 1 gives least compression and uses minimal memory. Compression level 9 gives the best compression but takes most memory.
compressionWorkFactor_	int	Compression work factor. Can be in range 0 to 250 inclusive. Setting the work factor to 0 uses the default work factor 30. See `http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.html` for details.
bufferSize_	int	Size of internal input and output buffers.

Mode() const Member Function

Returns the compression mode.

Syntax

```
public System.IO.Compression.CompressionMode Mode() const;
```

Returns

System.IO.Compression.CompressionMode

Returns the compression mode.

Read(byte*, int) Member Function

Reads compressed data from the underlying byte stream and decompresses it to the given buffer. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public int Read(byte* buf, int count);
```

Parameters

Name	Type	Description
buf	byte*	A buffer to decompress the data to.
count	int	Maximum number of bytes to read.

Returns

int

Returns the number of bytes read. Can be less than the size requested but is always non-negative. The return value of 0 indicates end of stream.

Remarks

Throws BZip2Exception if an error in decompression process is encountered. If an error reading from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

ReadByte() Member Function

Reads compressed data from the underlying byte stream, decompresses it to an internal buffer and returns one byte of decompressed data. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public int ReadByte();
```

Returns

int

Returns one byte of decompressed data, or -1 if end of stream is encountered.

Remarks

Throws BZip2Exception if an error in decompression process is encountered. If an error reading from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

Write(byte) Member Function

Writes one byte of data to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public void Write(byte x);
```

Parameters

Name	Type	Description
x	byte	Byte to write.

Remarks

Throws BZip2Exception if an error in compression process is encountered. If an error writing from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

Write(byte*, int) Member Function

Writes given number of bytes from the given buffer to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public void Write(byte* buf, int count);
```

Parameters

Name	Type	Description
buf	byte*	A buffer of data to write.
count	int	Number of bytes to write.

Remarks

Throws BZip2Exception if an error in compression process is encountered. If an error writing from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

~BZip2Stream() Member Function

Destructor. If the compression mode is compress compresses the rest of the data and writes it to the underlying byte stream. Releases occupied memory.

Syntax

```
public ~BZip2Stream();
```

2.1.3 DeflateException Class

Exception class thrown when DeflateStream cannot compress or decompress data.

Syntax

```
public class DeflateException;
```

Base Class

System.Exception

2.1.3.1 Member Functions

Member Function	Description
DeflateException()	Default constructor.
DeflateException(const System.IO.Compression.DeflateException&)	Copy constructor.
operator=(const System.IO.Compression.DeflateException&)	Copy assignment.
DeflateException(System.IO.Compression.DeflateException&&)	Move constructor.
operator=(System.IO.Compression.DeflateException&&)	Move assignment.
DeflateException(const System.String&, int)	Constructor. Initializes the **DeflateException** with the given error message and error code.
ErrorCode() const	Returns the error code.
~DeflateException()	Destructor.

DeflateException() Member Function

Default constructor.

Syntax

```
public DeflateException();
```

DeflateException(const System.IO.Compression.DeflateException&) Member Function

Copy constructor.

Syntax

```
public DeflateException(const System.IO.Compression.DeflateException& that);
```

Parameters

Name	Type	Description
that	const System.IO.Compression.DeflateException&	Argument to copy.

operator=(const System.IO.Compression.DeflateException&) Member Function

Copy assignment.

Syntax

```
public void operator=(const System.IO.Compression.DeflateException& that);
```

Parameters

Name	Type	Description
that	const System.IO.Compression.DeflateException&	Argument to assign.

DeflateException(System.IO.Compression.DeflateException&&) Member Function

Move constructor.

Syntax

```
public DeflateException(System.IO.Compression.DeflateException&& that);
```

Parameters

Name	Type	Description
that	System.IO.Compression.DeflateException&&	Argument to move from.

operator=(System.IO.Compression.DeflateException&&) Member Function

Move assignment.

Syntax

```
public void operator=(System.IO.Compression.DeflateException&& that);
```

Parameters

Name	Type	Description
that	System.IO.Compression.DeflateException&&	Argument to assign from.

DeflateException(const System.String&, int) Member Function

Constructor. Initializes the DeflateException with the given error message and error code.

Syntax

```
public DeflateException(const System.String& message_, int errorCode_);
```

Parameters

Name	Type	Description
message_	const System.String&	Error message.
errorCode_	int	Error code returned by the zlib library.

ErrorCode() const Member Function

Returns the error code.

Syntax

```
public int ErrorCode() const;
```

Returns

int

Returns the error code.

~DeflateException() Member Function

Destructor.

Syntax

```
public ~DeflateException();
```

2.1.4 DeflateStream Class

A stream class that writes data to the underlying byte stream in ZLIB compression format, or reads data compressed in ZLIB format from the underlying byte stream and decompresses it.

Syntax

`public class DeflateStream;`

Base Class

System.IO.ByteStream

2.1.4.1 Example

```
using System;
using System.IO;
using System.IO.Compression;

int main()
{
    try
    {
        {
            FileByteStream in(”deflate/test.file”, FileMode.open);
            FileByteStream out(”deflate/test.compressed”, FileMode.create);
            DeflateStream compressStream(out, CompressionMode.compress);
            in.CopyTo(compressStream);
        }
        {
            FileByteStream in(”deflate/test.compressed”, FileMode.open);
            DeflateStream decompressStream(in, CompressionMode.decompress);
            FileByteStream out(”deflate/test.decompressed”, FileMode.create)
                ;
            decompressStream.CopyTo(out);
        }
    }
    catch (const Exception& ex)
    {
        Console.Error() << ex.ToString() << endl();
        return 1;
    }
    return 0;
}
```

2.1.4.2 Member Functions

Member Function	Description
DeflateStream()	Default constructor.
DeflateStream(System.IO.ByteStream&, System.IO.Compression.CompressionMode)	Constructor. Initializes the System.IO.Compression.DeflatStream class with the given compression mode and underlying byte stream.

DeflateStream(System.IO.ByteStream&, System.IO.Compression.CompressionMode, int)	Constructor. Initializes the **DeflateStream** class with the given compression mode, buffer size and underlying byte stream.
DeflateStream(System.IO.ByteStream&, int)	Constructor. Initializes the **DeflateStream** class using compression mode compress and given compression level.
DeflateStream(System.IO.ByteStream&, int, int)	Constructor. Initializes the **DeflateStream** class using compression mode compress, given compression level and given buffer size.
Mode() const	Returns the compression mode.
Read(byte*, int)	Reads compressed data from the underlying byte stream and decompresses it to the given buffer. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.
ReadByte()	Reads compressed data from the underlying byte stream, decompresses it to an internal buffer and returns one byte of decompressed data. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.
Write(byte)	Writes one byte of data to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.
Write(byte*, int)	Writes given number of bytes from the given buffer to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.
~DeflateStream()	Destructor. If the compression mode is compress compresses the rest of the data and writes it to the underlying byte stream. Releases occupied memory.

DeflateStream() Member Function

Default constructor.

Syntax

```
public DeflateStream();
```

DeflateStream(System.IO.ByteStream&, System.IO.Compression.CompressionMode) Member Function

Constructor. Initializes the System.IO.Compression.DeflatStream class with the given compression mode and underlying byte stream.

Syntax

```
public DeflateStream(System.IO.ByteStream& underlyingStream_, System.IO.Compression.CompressionM
mode_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write to or read from.
mode_	System.IO.Compression.CompressionMode	Compression mode. Can be compress or decompress. When the compression mode is compress the stream supports writing, when the compression mode is decompress the stream supports reading.

Remarks

When the compression mode is compress, the stream uses default compression level defaultDeflateCompressionLevel and default buffer size 16K for internal input and output buffers.

DeflateStream(System.IO.ByteStream&, System.IO.Compression.CompressionMode, int) Member Function

Constructor. Initializes the DeflateStream class with the given compression mode, buffer size and underlying byte stream.

Syntax

```
public DeflateStream(System.IO.ByteStream& underlyingStream_, System.IO.Compression.CompressionM
mode_, int bufferSize_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write to or read from.
mode_	System.IO.Compression.CompressionMode	Compression mode. Can be compress or decompress. When the compression mode is compress the stream supports writing, when the compression mode is decompress the stream supports reading.
bufferSize_	int	Buffer size for internal input and output buffers.

Remarks

When the compression mode is compress, the stream uses default compression level defaultDeflateCompressionLevel.

DeflateStream(System.IO.ByteStream&, int) Member Function

Constructor. Initializes the DeflateStream class using compression mode compress and given compression level.

Syntax

```
public DeflateStream(System.IO.ByteStream& underlyingStream_, int compressionLevel_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write the compressed data to.
compressionLevel_	int	Compression level. Can be in range -1 to 9. Compression level defaultDeflateCompressionLevel (-1) equals compression level 6. Compression level noDeflateCompression (0) uses no compression. Compression level fastestDeflateCompression (1) gives fastest operation but least compression. Compression level optimalDeflateCompression (9) gives slowest operation but best compression.

Remarks

Uses the default buffer size 16K for internal input and output buffers.

DeflateStream(System.IO.ByteStream&, int, int) Member Function

Constructor. Initializes the DeflateStream class using compression mode compress, given compression level and given buffer size.

Syntax

```
public DeflateStream(System.IO.ByteStream& underlyingStream_, int compressionLevel_,
int bufferSize_);
```

Parameters

Name	Type	Description
underlyingStream_	System.IO.ByteStream&	Underlying byte stream to write the compressed data to.
compressionLevel_	int	Compression level. Can be in range -1 to 9. Compression level defaultDeflateCompressionLevel (-1) equals compression level 6. Compression level noDeflateCompression (0) uses no compression. Compression level fastestDeflateCompression (1) gives fastest operation but least compression. Compression level optimalDeflateCompression (9) gives slowest operation but best compression.
bufferSize_	int	Buffer size for internal input and output buffers.

Mode() const Member Function

Returns the compression mode.

Syntax

```
public System.IO.Compression.CompressionMode Mode() const;
```

Returns

System.IO.Compression.CompressionMode

Returns the compression mode.

Read(byte*, int) Member Function

Reads compressed data from the underlying byte stream and decompresses it to the given buffer. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public int Read(byte* buf, int count);
```

Parameters

Name	Type	Description
buf	byte*	A buffer to decompress the data to.
count	int	Maximum bumber of bytes to read.

Returns

int

Returns the number of bytes read. Can be less than the size requested but is always non-negative. The return value of 0 indicates end of stream.

Remarks

Throws DeflateException if an error in decompression process is encountered. If an error reading from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

ReadByte() Member Function

Reads compressed data from the underlying byte stream, decompresses it to an internal buffer and returns one byte of decompressed data. Actually the decompression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public int ReadByte();
```

Returns

int

Returns one byte of decompressed data, or -1 if end of stream is encountered.

Remarks

Throws DeflateException if an error in decompression process is encountered. If an error reading from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

Write(byte) Member Function

Writes one byte of data to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public void Write(byte x);
```

Parameters

Name	Type	Description
x	byte	Byte to write.

Remarks

Throws DeflateException if an error in compression process is encountered. If an error writing from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

Write(byte*, int) Member Function

Writes given number of bytes from the given buffer to an internal buffer, compresses it and writes the compressed data to the underlying byte stream. Actually the compression is not done on per call basis but more efficient means, but the result is as described.

Syntax

```
public void Write(byte* buf, int count);
```

Parameters

Name	Type	Description
buf	byte*	A buffer of data to write.
count	int	Number of bytes to write.

Remarks

Throws BZip2Exception if an error in compression process is encountered. If an error writing from the underlying byte stream is encountered, can also throw System.IO.IOException if the underlying byte stream is System.IO.FileByteStream, or System.Net.Sockets.SocketError if the underlying byte stream is System.Net.Sockets.SocketByteStream.

∼DeflateStream() Member Function

Destructor. If the compression mode is compress compresses the rest of the data and writes it to the underlying byte stream. Releases occupied memory.

Syntax

```
public ~DeflateStream();
```

2.2 Functions

Function	Description

2.3 Enumerations

Enumeration	Description
CompressionMode	Compression mode for BZip2Stream and DeflateStream classes.

2.3.4.3 CompressionMode Enumeration

Compression mode for BZip2Stream and DeflateStream classes.

Enumeration Constants

Constant	Value	Description
compress	0	Writes compressed data to the underlying byte stream.
decompress	1	Reads compressed data from the underlying byte stream and decompresses it.

2.4 Constants

Constant	Type	Value	Description
defaultBZip2CompressionLevel	int	9	Default bzip2 compression level.
defaultBZip2WorkFactor	int	0	Default bzip2 work factor.
defaultDeflateCompressionLevel	int	-1	Default deflate compression level.
fastestDeflateCompression	int	1	Fastest deflate compression levvel.
maximumBZip2WorkFactor	int	250	Maximum bzip2 work factor.
minimumBZip2CompressionLevel	int	1	Minumum bzip2 compression level.
noDeflateCompression	int	0	No deflate compression.
optimalBZip2CompressionLevel	int	9	Best bzip2 compression.
optimalDeflateCompression	int	9	Best deflate compression.