



OffsetStream Class

Wraps a source stream and offsets all read/write/seek calls by a given value.

Methods

Method	Description
<code>Close()</code>	Closes the underlying stream.
<code>Flush()</code>	Flushes the source stream.
<code>Read(buffer, offset, count)</code>	Reads a sequence of bytes from the source stream and advances the position within the stream by the number of bytes read.
<code>ReadByte()</code>	Reads a byte from the stream and advances the position within the source stream by one byte, or returns -1 if at the end of the stream.
<code>Seek(offset, origin)</code>	Sets the position within the current stream, which is equal to the position within the source stream minus the offset.
<code>SetLength(value)</code>	Sets the effective length of the stream, which is equal to the length of the source stream minus the

Method	Description
	offset.
<code>Write(buffer, offset, count)</code>	Writes a sequence of bytes to the source stream and advances the current position within this stream by the number of bytes written.
<code>WriteByte(value)</code>	Writes a byte to the current position in the source stream and advances the position within the stream by one byte.

Properties

Property	Description
<code>CanRead</code>	Gets a value indicating whether the source stream supports reading.
<code>CanSeek</code>	Gets a value indicating whether the source stream supports seeking.
<code>CanWrite</code>	Gets a value indicating whether the source stream supports writing.
<code>Length</code>	Gets the effective length of the stream, which is equal to the length of the source stream minus the offset.
<code>Offset</code>	Gets the number of bytes to offset all calls before redirecting to the underlying stream.

Property	Description
Position	Gets or sets the effective position of the stream, which is equal to the position of the source stream minus the offset.
Source	Gets the underlying stream that this OffsetStream calls into.

Remarks

This class is used to trick archive an packing or unpacking process into reading or writing at an offset into a file, primarily for self-extracting packages. `WixToolset.Dtf.Compression.dll` version `4.0.2+644ed0118bae1aa885bb6cc906dc6c0b904de4d9`

Close() Method

Closes the underlying stream.

Declaration

```
public void Close()
```

Flush() Method

Flushes the source stream.

Declaration

```
public void Flush()
```

Read(buffer, offset, count) Method

Reads a sequence of bytes from the source stream and advances the position within the stream by the number of bytes read.

Declaration

```
public int Read(  
    System.Byte[] buffer,  
    int offset,  
    int count  
)
```

Parameters

Parameter	Type	Description
buffer	System.Byte[]	An array of bytes. When this method returns, the buffer contains the specified byte array with the values between offset and (offset + count - 1) replaced by the bytes read from the current source.

Parameter	Type	Description
offset	int	The zero-based byte offset in buffer at which to begin storing the data read from the current stream.
count	int	The maximum number of bytes to be read from the current stream.

Return value

`int` The total number of bytes read into the buffer. This can be less than the number of bytes requested if that many bytes are not currently available, or zero (0) if the end of the stream has been reached.

ReadByte() Method

Reads a byte from the stream and advances the position within the source stream by one byte, or returns -1 if at the end of the stream.

Declaration

```
public int ReadByte()
```

Return value

`int` The unsigned byte cast to an Int32, or -1 if at the end of the stream.

Seek(offset, origin) Method

Sets the position within the current stream, which is equal to the position within the source stream minus the offset.

Declaration

```
public System.Int64 Seek(  
    System.Int64 offset,  
    System.IO.SeekOrigin origin  
)
```

Parameters

Parameter	Type	Description
offset	System.Int64	A byte offset relative to the origin parameter.
origin	System.IO.SeekOrigin	A value of type SeekOrigin indicating the reference point used to obtain the new position.

Return value

`System.Int64` The new position within the current stream.

SetLength(value) Method

Sets the effective length of the stream, which is equal to the length of the source stream minus the offset.

Declaration

```
public void SetLength(  
    System.Int64 value  
)
```

Parameters

Parameter	Type	Description
value	System.Int64	The desired length of the current stream in bytes.

Write(buffer, offset, count) Method

Writes a sequence of bytes to the source stream and advances the current position within this stream by the number of bytes written.

Declaration

```
public void Write(  

```

```
System.Byte[] buffer,  
int offset,  
int count  
)
```

Parameters

Parameter	Type	Description
buffer	System.Byte[]	An array of bytes. This method copies count bytes from buffer to the current stream.
offset	int	The zero-based byte offset in buffer at which to begin copying bytes to the current stream.
count	int	The number of bytes to be written to the current stream.

WriteByte(value) Method

Writes a byte to the current position in the source stream and advances the position within the stream by one byte.

Declaration

```
public void WriteByte(  
    System.Byte value  
)
```


Parameters

Parameter	Type	Description
value	System.Byte	The byte to write to the stream.

CanRead Property

Gets a value indicating whether the source stream supports reading.

Declaration

```
public bool CanRead { get; set; }
```

CanSeek Property

Gets a value indicating whether the source stream supports seeking.

Declaration

```
public bool CanSeek { get; set; }
```

CanWrite Property

Gets a value indicating whether the source stream supports writing.

Declaration

```
public bool CanWrite { get; set; }
```

Length Property

Gets the effective length of the stream, which is equal to the length of the source stream minus the offset.

Declaration

```
public System.Int64 Length { get; set; }
```

Offset Property

Gets the number of bytes to offset all calls before redirecting to the underlying stream.

Declaration

```
public System.Int64 Offset { get; set; }
```

Gets or sets the effective position of the stream, which is equal to the position of the source stream minus the offset.

Declaration

```
public System.Int64 Position { get; set; }
```

Source Property

Gets the underlying stream that this OffsetStream calls into.

Declaration

```
public System.IO.Stream Source { get; set; }
```