Class

ArchiveByteStream

An archive stream that reads from and writes to buffers.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst | macOS 11.0+ | tvOS 14.0+ | visionOS | watchOS 7.0+

```
class ArchiveByteStream
```

Topics

Creating an Archive Byte Stream

`init(object: _AAOptionalObjectWrapper<_AAByteStreamTraits>.AAType?, owned: Bool)`

Returns a new archive byte stream from the specified traits and entry message processing callback.

Using Archive Byte Streams

`func close(updatingContext: ArchiveEncryptionContext) throws`

Closes the stream, releases associated resources, and writes the sealed container attributes to the specified encryption context.

Compressing Data

`static func compressionStream(using: ArchiveCompression, writingTo: ArchiveByteStream, blockSize: Int, flags: ArchiveFlags, threadCount: Int) -> ArchiveByteStream?`

Creates a compression sequential output stream.

```
static func withCompressionStream<E>(using: ArchiveCompression, writingTo: ArchiveByteStream, blockSize: Int, flags: ArchiveFlags, threadCount: Int, (ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a compression sequential output stream.

```
static func compressionStream(appendingTo: ArchiveByteStream, flags: ArchiveFlags, threadCount: Int) -> ArchiveByteStream?
```

Reopens a compression sequential output stream.

```
static func withCompressionStream<E>(appendingTo: ArchiveByteStream, flags: ArchiveFlags, threadCount: Int, (ArchiveByteStream) throws -> E) throws -> E
```

Reopens a compression sequential output stream and calls the given closure.

Decompressing Data

```
static func decompressionStream(readingFrom: ArchiveByteStream, flags: ArchiveFlags, threadCount: Int) -> ArchiveByteStream?
```

Creates a decompression sequential input stream.

```
static func withDecompressionStream<E>(readingFrom: ArchiveByteStream, flags: ArchiveFlags, threadCount: Int, (ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a decompression sequential input stream.

```
static func randomAccessDecompressionStream(readingFrom: ArchiveByteStream, allocationLimit: Int, flags: ArchiveFlags, threadCount: Int) -> ArchiveByteStream?
```

Creates a decompression random-access input stream.

```
static func withRandomAccessDecompressionStream<E>(readingFrom: ArchiveByteStream, allocationLimit: Int, flags: ArchiveFlags, threadCount: Int, (ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a decompression random access input stream.

Encrypting Data

```
static func encryptionStream(appendingTo: ArchiveByteStream, encryption
Context: ArchiveEncryptionContext, flags: ArchiveFlags, threadCount:
Int) -> ArchiveByteStream?
```

Reopens an existing encryption sequential output stream.

```
static func encryptionStream(writingTo: ArchiveByteStream, encryption
Context: ArchiveEncryptionContext, flags: ArchiveFlags, threadCount:
Int) -> ArchiveByteStream?
```

Creates a encryption sequential input stream.

Decrypting Data

```
static func decryptionStream(readingFrom: ArchiveByteStream, encryption
Context: ArchiveEncryptionContext, flags: ArchiveFlags, threadCount:
Int) -> ArchiveByteStream?
```

Creates a decryption sequential input stream.

```
static func randomAccessDecryptionStream(readingFrom: ArchiveByteStream
, encryptionContext: ArchiveEncryptionContext, allocationLimit: Int,
flags: ArchiveFlags, threadCount: Int) -> ArchiveByteStream?
```

Creates a decryption random access input stream.

Processing Data

```
static func process(readingFrom: ArchiveByteStream, writingTo: Archive
ByteStream) throws -> Int64
```

Processes data between two byte streams.

File Streaming

```
static func fileStream(fd: FileDescriptor, automaticClose: Bool) ->
ArchiveByteStream?
```

Creates a stream from an open file descriptor.

```
static func withFileStream<E>(fd: FileDescriptor, automaticClose: Bool,
(ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a file stream created from the specified file descriptor.

```
static func fileStream(path: FilePath, mode: FileDescriptor.AccessMode, options: FileDescriptor.OpenOptions, permissions: FilePermissions) -> ArchiveByteStream?
```

Opens a new file descriptor using the given path and parameters, and creates a stream from the file descriptor.

```
static func withFileStream<E>(path: FilePath, mode: FileDescriptor.AccessMode, options: FileDescriptor.OpenOptions, permissions: FilePermissions, (ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a file stream.

```
static func temporaryFileStream() -> ArchiveByteStream?
```

Creates a new temporary file stream.

```
static func withTemporaryFileStream<E>((ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with a temporary file stream.

Streaming with Custom Streams

```
static func customStream<C>(instance: C) -> ArchiveByteStream?
```

Returns a new archive byte stream instance mapped to an object that conforms to the archive byte stream protocol.

```
static func withStream<C, E>(wrapping: C, (ArchiveByteStream) throws -> E) throws -> E
```

Calls the given closure with an archive byte stream instance mapped to an object that conforms to the archive byte stream protocol.

```
static func sharedBufferPipe(capacity: Int) -> (output: ArchiveByteStream, input: ArchiveByteStream)?
```

Creates a pair of streams and links them by a shared buffer.

Relationships

Conforms To

ArchiveByteStreamProtocol

See Also

Apple Archive streams

`protocol` ArchiveStreamProtocol

A set of methods that defines the interface for using an archive stream that reads from and writes to data blobs.

`class` ArchiveStream

An archive stream that reads from and writes to data blobs

`protocol` ArchiveByteStreamProtocol

A set of methods that defines the interface for using an archive stream that reads from and writes to buffers.