Structure

compression_stream

A structure representing a compression stream.

iOS | iPadOS | Mac Catalyst | macOS | tvOS | visionOS | watchOS

```
struct compression_stream
```

Overview

The basic workflow for using the stream interface is as follows:

1. Initialize the state of your `compression_stream` structure by calling `compression_stream_init(_:_:_:)` with the `operation` parameter set to specify whether you are encoding or decoding, and the chosen algorithm specified by the `algorithm` parameter. This allocates storage for the state that allows you to resume encoding or decoding across calls.
2. Set the `dst_buffer`, `dst_size`, `src_buffer`, and `src_size` fields of the `compression_stream` object to point to the next blocks that your code processes.
3. Call `compression_stream_process(_:_:)`. If no further input will be added to the stream via subsequent calls, `flags` should be `COMPRESSION_STREAM_FINALIZE` (otherwise 0). If `compression_stream_process(_:_:)` returns `COMPRESSION_STATUS_END`, there is no further output from the stream.
4. Repeat steps 2 and 3 as necessary to process the entire stream.
5. Call `compression_stream_destroy(_:)` to free the state object in the stream structure.

Topics

Initializers

`init(dst_ptr: UnsafeMutablePointer<UInt8>, dst_size: Int, src_ptr: UnsafePointer<UInt8>, src_size: Int, state: UnsafeMutableRawPointer?)`

Returns a new compression stream structure.

Compression Stream Properties

`var dst_ptr: UnsafeMutablePointer<UInt8>`

A pointer to the first byte of the destination buffer.

`var dst_size: Int`

The size, in bytes, of the destination buffer.

`var src_ptr: UnsafePointer<UInt8>`

A pointer to the first byte of the source buffer.

`var src_size: Int`

The size, in bytes, of the source buffer.

`var state: UnsafeMutableRawPointer?`

The stream state object of the compression stream.

Relationships

Conforms To

`BitwiseCopyable`

See Also

Multiple-step compression

```
func compression_stream_init(UnsafeMutablePointer<compression_stream>, compression_stream_operation, compression_algorithm) -> compression_status
```

Initializes a compression stream for either compression or decompression.

```
func compression_stream_process(UnsafeMutablePointer<compression_stream>, Int32) -> compression_status
```

Performs compression or decompression using an initialized compression stream structure.

```
func compression_stream_destroy(UnsafeMutablePointer<compression_stream>) -> compression_status
```

Frees any memory allocated by stream initialization function.

```
struct compression_status
```

A set of values used to represent the status of stream compression.

```
struct compression_stream_flags
```

A set of values used to represent stream compression flags.

```
struct compression_stream_operation
```

A set of values used to represent a stream compression operation.

```
struct compression_algorithm
```

A structure for values that represent compression algorithms.