

## □ Documentation

[Accelerate](#) / Decompressing single files

Article

# Decompressing single files

Recreate a single file from a compressed file.



## Overview

In this article, you'll learn how to use AppleArchive to decompress a previously compressed file, and write the decompressed data to a file.

The code below decompresses the file generated using the steps explained in [Compressing single files](#).

## Create the file stream to read the source archive

The [ArchiveByteStream](#) class provides static factory methods that create streams for different functions. In this case, use [fileStream\(path:mode:options:permissions:\)](#) to create a byte stream that reads the source file:

```
let archiveFilePath = FilePath(NSTemporaryDirectory()) + "myFile.pdf.lzfse"

guard let readFileStream = ArchiveByteStream.fileStream(
    path: archiveFilePath,
    mode: .readOnly,
    options: [ ],
    permissions: FilePermissions(rawValue: 0o644)) else {
    return
}

defer {
    try? readFileStream.close()
}
```

## Create the file stream to write the decompressed file

You also use `fileStream(path:mode:options:permissions:)` to create the file stream that writes the decompressed file to the file system. In this case, use the `writeOnly` mode:

```
let destinationFilePath = FilePath(NSTemporaryDirectory()) + "myFile_decompressed.pdf"

guard let writeFileStream = ArchiveByteStream.fileStream(
    path: destinationFilePath,
    mode: .writeOnly,
    options: [ .create ],
    permissions: FilePermissions(rawValue: 0o644)) else {
    return
}

defer {
    try? writeFileStream.close()
}
```

# Create the decompression stream

Create the decompression stream. Specify the file-reading stream as the input stream that provides the compressed data:

```
guard let decompressStream = ArchiveByteStream.decompressionStream(readingFrom: read  
    print("unable to create compress stream")  
    return  
}  
defer {  
    try? decompressStream.close()  
}
```

# Decompress the source archive

Finally, call `process(readingFrom:writingTo:)` to write the output of the decompression stream to the file-writing stream:

```
print("Handle `ArchiveByteStream.process` failed.")
```

```
}
```

On return, `myFile_decompressed.pdf` exists in `NSTemporaryDirectory()` and contains the decompressed contents of `myFile.pdf.lzfse`.

---

## See Also

### Directories, Files, and Data Archives

📄 [Compressing single files](#)

Compress a single file and store the result on the file system.

📄 [Compressing file system directories](#)

Compress the contents of an entire directory and store the result on the file system.

📄 [Decompressing and extracting an archived directory](#)

Recreate an entire file system directory from an archive file.

📄 [Compressing and saving a string to the file system](#)

Compress the contents of a Unicode string and store the result on the file system.

📄 [Decompressing and Parsing an Archived String](#)

Recreate a string from an archive file.