

[Accelerate](#) / Compressing file system directories

## Article

# Compressing file system directories

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

## Overview

In this article, you'll learn how to use `AppleArchive` to compress the contents of an entire directory to a single archive file.

The code below compresses the contents of a directory name `src` using the [Algorithm.lzfs](#) algorithm, and stores the result in a file named `directory.aar`.

## Create the file stream to write the compressed file

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

```
let archiveDestination = NSTemporaryDirectory() + "directory.aar"
let archiveFilePath = FilePath(archiveDestination)

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

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

## Create the compression stream

Create the compression stream, and specify the compression algorithm as `lzfs`. Specify the file-writing stream as the stream that receives the compressed data:

```
guard let compressStream = ArchiveByteStream.compressionStream(
    using: .lzfs,
    writeTo: writeFileStream) else {
    return
}
defer {
    try? compressStream.close()
}
```

## Create the encoding stream

Create the encoding stream. The encoding stream encodes its data as a byte stream, and sends the encoded data to the compression stream:

```
guard let encodeStream = ArchiveStream.encodeStream(writeTo: compressStream) else
    return
}
defer {
    try? encodeStream.close()
}
```

## Define the header keys

Create a field key set that defines the fields in the archive header:

```
guard let keySet = ArchiveHeader.FieldKeySet("TYP,PAT,LNK,DEV,DAT,UID,GID,MOD,FLG,M1
    return
}
```

For more information about three-letter keys, see `init(_ :)`.

## Compress the directory contents

Use `writeDirectoryContents(archiveFrom:path:keySet:selectUsing:flags:threadCount:)` to write the directory contents to the encode stream. In turn, the encode stream writes to the compression stream and then, the compression stream writes to the file stream. Finally, the file stream writes the archive file to the file system:

```
let sourcePath = NSTemporaryDirectory() + "src/"
let source = FilePath(sourcePath)

do {
    try encodeStream.writeDirectoryContents(
        archiveFrom: source,
        keySet: keySet)
} catch {
    fatalError("Write directory contents failed.")
}
```

On return, `directory.aar` exists in `NSTemporaryDirectory()` and contains the compressed contents of `src/`.

---

## See Also

### Directories, Files, and Data Archives



Compressing single files

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



Decompressing single files

Recreate a single file from a compressed file.



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.