

# Zone File System

The *Zone File System* is a method used by *Star Trek: Armada* to store game assets in archive-like files, namely the *objtxt.zfs* and *assets.zfs*. Especially the latter contains most of the [ODF files](#) as well as the [tech tree files](#). The contents are not stored in their plain format, so the whole storage method may be considered kind of a method of obfuscation of actual game assets. The format is, however not very complicated to decipher/unpack.

## ZFS File Format

A ZFS has a header as well as directory-like sections and actual file content sections.

### Header

The header has a length of 28 byte, consisting of the following information:

Start Offset	Data Type	Meaning
0	uint32	ID of the File, usually ZFSF
4	uint32	Version of the Format of the File, for Armada That is 1.
8	uint32	Maximum length of a File Name in Directories ( <i>max. file name length</i> ).
12	uint32	Maximum Number of Files per Directory ( <i>max. files per directory</i> ).
16	uint32	Number of Files Stored in Total in the ZFS.
20	uint32	<i>Cypher Key</i>
24	uint32	Offset of the first Directory Block

### Directory Entry

Each directory entry basically consists of a file name and some meta data of where the file content is stored in the ZFS container and its length. It has the following structure:

Offset	Length	Meaning
0	<i>max. file name length</i>	The File Name of This Directory Entry, to be Used When Extracting the File.
<i>max. file name length</i>	uint32	Offset of the Actual File Contents
<i>max. file name length</i> + 4	uint32	File ID
<i>max. file name length</i> + 8	uint32	File Length (in Bytes)
<i>max. file name length</i> + 12	4	File Hash (Hashing Method is not Clear)
<i>max. file name length</i> + 16	4	Zeros (= Unused)

This gives every directory entry the full length of *max. file name length* + 20 bytes (*max. entry length*).

Each file name is stored as ASCII encoded string, ended with a zero-byte (unless using the maximum number of bytes, then the zero byte might be omitted).

If a file name consists only of zero bytes it means the block is not used. As there are no blocks following an unused block, the occurrence of an only-zero-bytes-file name marks the end of the directory block. This should fit the number of files set in the ZFS header.

## Directories

The meta data of where to find the actual file contents and under what file name they are addressed is found in directory blocks. There must be at least one directory block. (Otherwise no payload files are contained, which makes no sense.) But there can be more than one block. Each block contains the information on where the next directory block is to be found as well as the actual directory entries (see above). Each directory block handles a number of files, the maximum number being set in the header by *max. files per directory*. A directory looks like this:

Offset	Length	Meaning
0	uint32	<i>Next Directory Block Offset</i>
4	<i>max. entry length x max. files per directory</i>	The Actual Directory Entries of This Directory

This implies that without having read the entire directory block-by-block, there is no explicit information of how many directory blocks are in a ZFS file or how many are following after a specific one. Only once reached the last block is it clear, that no other blocks follow.

If the *Next Directory Block Offset* is zero it implies that no more directory blocks follow, marking the block the last one of the directory.

## Encryption/File Contents

The file content itself is not stored in plain in the ZFS files, while the file names and metadata of directories are. The »encryption« of the file contents is actually very simple: Each file is stored in chunks of 4 bytes, each being xor-ed with the *Cypher Key* taken from the header. So in essence the »password« is provided along with the encrypted file contents. This is why this method of storage cannot seriously be considered proper encryption. Neither is the algorithm strong, nor is an already provided password any serious blockage from reading the file. It's at best a method of obfuscating things.

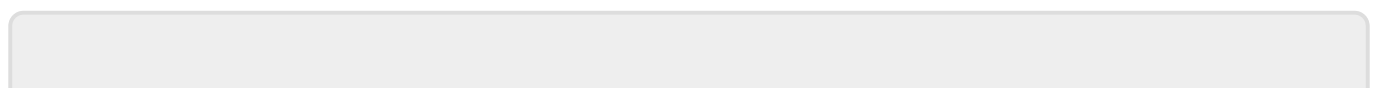
---

[ [Star Trek: Armada](#) ] [ [Game Play](#) ] [ [Technical Support](#) ] [ [Trivia](#) ]

[ [Units](#) ] [ [Hero Ships](#) ] [ [Campaign Ships](#) ] [ [Ship Types](#) ] [ [Stations](#) ] [ [Station Types](#) ] [ [Special Weapons](#) ] [ [Factions](#) ] [ [Modding](#) ]

---

[ [Back to the Games Database](#) ]



From:

<https://mwohlauer.d-n-s.name/wiki/> - **mwohlauer.d-n-s.name** / **www.mobile-infanterie.de**

Permanent link:

[https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:star\\_trek\\_armada\\_1:zone\\_file\\_system&rev=1746402888](https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:star_trek_armada_1:zone_file_system&rev=1746402888)

Last update: **2025-05-04-23-54**

