

# Project File

## High Level Overview

Each COLDSTEEL Sprite Studio project is saved as `.ctsp` file (COLDSTEEL Sprite Studio Project). This single file contains all the needed information to load and restore a previously created project. The name preceding the file extension is the name of the project.

## Description

`.ctsp` files are located within the `data/projects/` folder within the archive of the installation of the program. The file contains all needed information about the project itself, palettes, animations, layers, and artboards.

Numerical values are stored in the big endian byte order.

## Layout

This is the `.ctsp` file format:

Entry Name	Entry Type	Entry Size (bytes)
Project Name	<code>String</code>	Varying
Channels Per Pixel	<code>byte</code>	1
Number of Source Artboards	<code>int</code>	4
Number of Visual Layers	<code>int</code>	4
Number of NonVisual Layers	<code>int</code>	4
Number of Animations	<code>int</code>	4
Palette Chunks	Palette Chunk <code>[]</code>	Varying
Visual Layer(s)	Visual Layer Prototype <code>[]</code>	Varying

NonVisual Layer(s)	NonVisual Layer Prototype <code>[]</code>	Varying
Artboard Chunk(s)	Artboard Chunk <code>[]</code>	Varying
Animation Chunk(s)	Animation Chunk <code>[]</code>	Varying

Palette Chunks is a list of palette chunks. Each palette chunk contains data needed to create a color palette from which color indices apply. There will always be five palette chunks because there are five palettes in the program.

Entry Name	Entry Type	Entry Size (bytes)
Palette Width	<code>int</code>	4
Palette Height	<code>int</code>	4
Palette Channels Per Pixel	<code>byte</code>	1
Palette Data	<code>byte[]</code>	Varying

The second composite type within the format is the Visual Layer(s) entry, which is a list of strings which hold the names of visual layers.

The Visual Layer(s) format:

Entry Name	Entry Type	Entry Size (bytes)
Visual Layer Names	<code>String[]</code>	Varying

The third composite type within the format is the NonVisual Layer(s) entry, which is a list of strings and bytes representing the names of the nonvisual layers and how many bytes each layer is

The NonVisual Layer(s) format:

Entry Name	Entry Type	Entry Size (bytes)
Nonvisual Layer Names	<code>String</code>	Varying
Size	<code>byte</code>	1

The next composite type within the format is the artboard chunk entry. An artboard chunk contains necessary information about artboards. Each artboard has its own chunk and there will be Number Of Source Artboards artboard chunks.

## Artboard Chunk(s)

Entry Name	Entry Type	Entry Size (bytes)
Artboard Name	String	Varying
Artboard Width	int	4
Artboard Height	int	4
Active Layer Index	int	4
Is Active Layer Visual	boolean	1
Visual Layer Data Chunk(s)	Visual Layer Data Chunk []	Varying
Nonvisual Layer Data Chunk(s)	Nonvisual Layer Data Chunk []	Varying

Visual Layer Data Chunks are sub chunks found within artboard chunks that store data about visual layers. Some data about visual layers can delegate to the artboard, which is width, height, and channels per pixel.

## Visual Layer Data Chunk(s)

Entry Name	Entry Type	Entry Size (bytes)
Name	String	Varying
Locked	boolean	1
Hiding	boolean	1
Is Compressed	boolean	1
Pixel Data	byte[]	Varying

Nonvisual Layer Data Chunks are sub chunks found within artboard chunks that store data about nonvisual layers. Some data about visual layers can delegate to the artboard, that being width and height.

Entry Name	Entry Type	Entry Size (bytes)
Name	String	Varying
Locked	boolean	1
Hiding	boolean	1
Bytes Per Pixel	byte	1
Is Compressed	boolean	1

Pixel Data	<code>byte[]</code>	Varying
------------	---------------------	---------

The last chunk is the animation chunk list. Each animation for the project is encapsulated in an animation chunk and numberAnimations chunks are found after artboards.

Entry Name	Entry Type	Entry Size (bytes)
Animation Name	<code>String</code>	Varying
Number Frames	<code>int</code>	4
Default Swap Time	<code>float</code>	4
Default Updates	<code>int</code>	4
Default Swap Type	<code>String</code>	Varying
Animation Frame Chunk(s)	Animation Frame Chunk <code>[]</code>	Varying

The Animation Frame Chunk List is a list of Animation Frame Chunks, each of which contain data about a single animation frame. There will be a Number Frames number of animation frame chunks in the Animation Frame Chunk(s) entry.

Entry Name	Entry Type	Entry Size (bytes)
Artboard Name	<code>String</code>	Varying
Frame Time	<code>float</code>	4
Frame Updates	<code>int</code>	4
Swap Type	<code>String</code>	Varying

## How to read Arrays

The first byte (at least) of arrays is a list size prefix. Read the highest two bits of the first byte of an array entry into another number. This number tells how many *additional* bytes to read ahead to compose a total size in characters.

So, if the highest two bits, into their own value (done by shifting the first byte right 6) equals 1, there is one more byte to read to complete the size prefix. Perform a bitwise AND to remove the high two bits from the original byte and read the additional byte, then combine the two into a big endian number. This is the number of remaining elements in the list.