

# libgxt Overview

© 2011 Sony Computer Entertainment Inc.  
All Rights Reserved.  
SCE Confidential

## Table of Contents

---

<b>About This Document .....</b>	<b>3</b>
Typographic Conventions.....	3
Related Documentation.....	3
<b>1 Introduction .....</b>	<b>4</b>
Characteristics.....	4
Files .....	4
<b>2 GXT File Format .....</b>	<b>5</b>
File Layout.....	5
GXT File Header .....	6
GXT Texture Info .....	6
<b>3 Initializing Textures from GXT Data .....</b>	<b>7</b>
Initializing a Texture.....	7
Multiple Textures in GXT Data.....	7
Palettes .....	7

---

## About This Document

---

This document describes the GXT file format and provides information about the runtime library that can be used to extract information from this format.

### Typographic Conventions

The typographic conventions used in this guide are explained in this section.

#### Text

File names, source code and command-line text are formatted in a fixed-width font. For example:

```
target\include\gxt.h
```

#### Hyperlinks

Hyperlinks (underlined and in blue) are available to help you to navigate around the document. To return to where you clicked a hyperlink, select **View > Toolbars > More Tools** from the Adobe Reader main menu, and then enable the **Previous View** and **Next View** buttons.

#### Notes

Additional advice or related information is presented as a “Note” surrounded by a box. For example:

<b>Note:</b> Example note.
----------------------------

### Related Documentation

Any updates or amendments to this guide can be found in the release notes that accompany the release.

For detailed descriptions of the offline texture tools, refer to the *Texture Pipeline User's Guide*. For detailed descriptions of the GPU data layout for each texture format and type, refer to the *GPU User's Guide*.

# 1 Introduction

This chapter introduces the GXT library.

## Characteristics

The GXT library facilitates the runtime manipulation of GXT files produced using the offline texture tools.

## Files

The files required to use the GXT library are listed in Table 1.

**Table 1 Prerequisite Files**

File Name	Description
target/include_common/gxt.h	Header file.

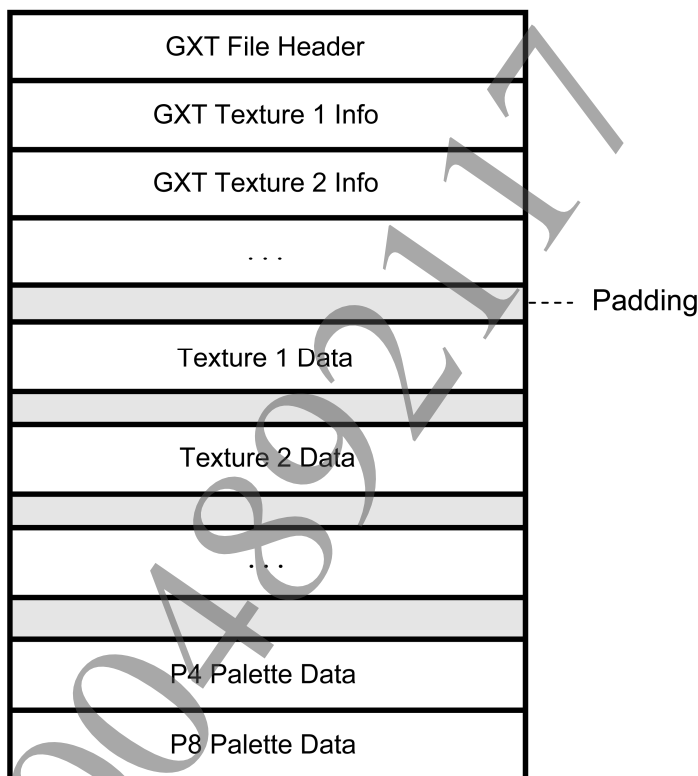
## 2 GXT File Format

This chapter describes the GXT file format.

### File Layout

A GXT file consists of a header section, followed by a data section.

**Figure 1 Overall Layout of GXT**



The header section consists of two parts:

- The GXT file header contains information such as GXT version, number of textures, and so on.
- Immediately after the GXT file header, there are one or more GXT texture info blocks – one for each texture contained within the file. These blocks contain information about the texture (format, dimensions, and so on) and an offset to the texture data.

The texture data section follows the header section.

- Texture data is arranged in a GPU-ready format. Mipmaps and cubemap faces are included in the texture data for a single texture. Padding is inserted before the start of the texture data to ensure that the texture data always starts on a 16-byte boundary.
- Palette data follows the texture data. P4 (16 entry) palettes are stored first followed by P8 (256 entry) palettes. Padding is inserted before the start of the palette data to ensure that it always starts on a 64-byte boundary.

GXT files that contain palettes must be aligned to 64 bytes in memory; otherwise, the palette data will not be aligned correctly. GXT files that do not contain palettes only require 16-byte alignment because this is sufficient for all texture formats that do not use palettes.

Refer to the *GPU User's Guide* for descriptions of the GPU data layout for each texture format and type.

## GXT File Header

A GXT file header contains the information shown in Table 2.

**Table 2 GXT File Header Data**

Member	Description
<i>tag</i>	'GXT\0'.
<i>version</i>	GXT file format version number.
<i>numTextures</i>	Number of textures contained within the file.
<i>dataOffset</i>	The offset to the start of the texture data, relative to the start of the file (bytes).
<i>dataSize</i>	Size of all texture data (bytes).
<i>numPalettes16</i>	The number of 16 entry palettes.
<i>numPalettes256</i>	The number of 256 entry palettes.

## GXT Texture Info

A GXT texture info contains the information shown in Table 3.

**Table 3 GXT Texture Info Data**

Member	Description
<i>dataOffset</i>	Offset to the texture data for this texture, relative to the start of the file (bytes).
<i>dataSize</i>	Size of the texture data (bytes).
<i>paletteIndex</i>	The index of the palette for this texture.
<i>flags</i>	Texture flags.
<i>type</i>	Texture type ( <i>SceGxmTextureType</i> ).
<i>format</i>	Texture format ( <i>SceGxmTextureFormat</i> ).
<i>width</i>	Texture width.
<i>height</i>	Texture height.
<i>mipCount</i>	Number of mipmaps.

## 3 Initializing Textures from GXT Data

This chapter provides code samples that demonstrate how to initialize a `SceGxmTexture` from GXT data in memory.

### Initializing a Texture

In this example, `gxt` is assumed to point to GXT data in non-mapped memory. First, the texture data is copied to GPU mapped memory, and then this memory is used to initialize the texture.

**Note:** It is also possible to load the texture data directly into mapped memory.

```
// Is this a valid GXT?
assert(sceGxtCheckData(gxt) == SCE_OK);

// Get the size of the texture data.
const uint32_t dataSize = sceGxtGetDataSize(gxt);

// Allocate memory.
int32_t textureDataUid;
void *textureData = buffer; // Some mapped memory that the user supplies

// Get pointer to start of the texture data.
const void *dataSrc = sceGxtGetDataAddress(gxt);

// Copy texture data to GPU mapped memory.
memcpy(textureData, dataSrc, dataSize);

// Set up the texture control words for texture 0.
SceGxmTexture texture;
sceGxtInitTexture(&texture, gxt, textureData, 0);
```

For additional information about GPU memory mapping considerations, refer to the *libgxm Overview*.

### Multiple Textures in GXT Data

For GXT files that contain multiple textures, individual textures can be initialized by passing the appropriate index to `sceGxtInitTexture`, as shown in the following example:

```
// Get the number of textures contained in the GXT.
uint32_t numTextures = sceGxtGetTextureCount(gxt);

// Initialize all textures.
for(uint32_t i=0; i<numTextures; i++) {
    sceGxtInitTexture(&textures[i], gxt, textureData, i);
}
```

### Palettes

For palettized textures, the default palette for the texture is set during the call to `sceGxtInitTexture()`. Different palettes within the GXT file can be set by passing the appropriate index to `sceGxtSetPalette()`. Depending on the format of the texture, the index refers either to the list of P4 palettes or the list of P8 palettes.

```
// Use palette 1 for this texture
sceGxtSetPalette(&texture, gxt, textureData, 1);
```