# Texture Packager Library

## Runtime Specifications

Version 1.1

PROVISIONAL TRANSLATION

**The content of this document is highly confidential and should be handled accordingly.**

# Table of Contents



# Code



# Figures

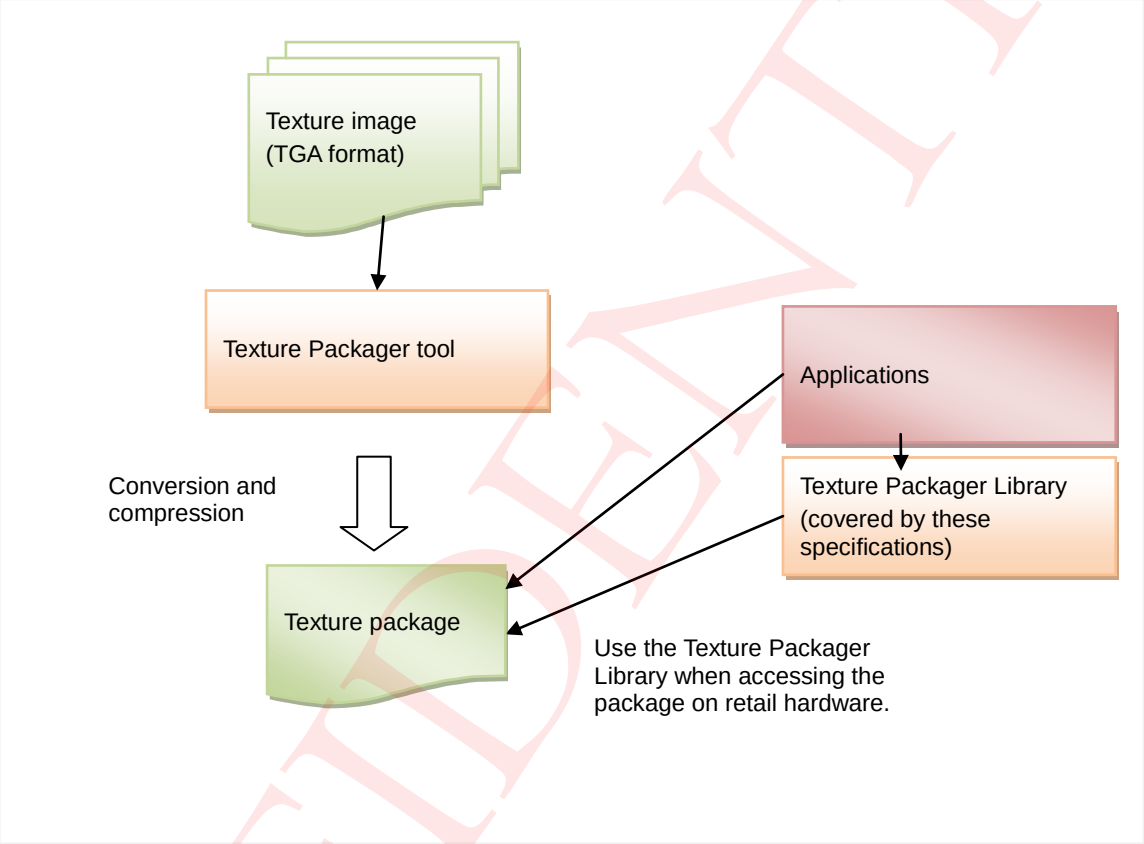
# Revision History

| Version | Revision Date | Description |
|---|---|---|
| 1.1 | 2010/07/27 | • Corrected the function names in Chapter 2 Using the Tool.<br>• |
| 1.0 | 2010/05/20 | • Official release.<br>• Grouped objects in Figure 1-1 Texture Packager Library Structure.<br>• Revised documentation to reflect name changes for structures and other elements. |
| 0.9.3 | 2010/05/19 | • Used Word template to rework format. |
| 0.9.2 | 2010/04/27 | • In Chapter 3 Function Reference, fixed function names to use camel case. |
| 0.9.1 | 2010/04/23 | • In 3.1.4 Function to Get a Texture, revised procedure for getting a texture so that it also gets a mipmap. Previously, there was no way to get a mipmap. |
| 0.9.0 | 2010/03/25 | • Rough draft. |

# 1 Overview of Features

The Texture Packager Library is used to read textures (image data) from texture package files created by the Texture Packager tool (`ctr_TexturePackager.exe`).

**Figure 1-1 Texture Packager Library Structure**



# 2 Using the Tool

1. An application loads a texture package file into RAM. The pointer to the start of the texture package file must be 128-byte aligned.
2. Call the `GetTextureIndex` function, passing as arguments a pointer to the start of the texture package file and the name of the texture file as specified in the texture settings file at the time of conversion. The function returns the index of the texture file.
3. Call the `GetTexture` function, passing the index obtained in step 2 as an argument, to get the texture (a pointer to the image data). If the pointer to the start of the texture package file from Step 1 is 128-byte aligned, the pointer to the image data obtained from calling the `GetTexture` function will also be 128-byte aligned.

**Code 2-1**

```
#include "tpl_ReadTexturePackage.h"

void* hoge(void* pTexPackage)
{
  unsigned int texIndex = 0;
  // Gets texture index value
  texIndex = nn::tpl::CTR::GetTextureIndex(pTexPackage, "data/images001.tga");

  // Returns the obtained texture data pointer
  return nn::tpl::CTR::GetTexture(pTexPackage, texIndex);
}
```

# 3 Function Reference

## 3.1 Function to Get a Texture Index

The `GetTextureIndex` function gets the index for a texture. The index indicates its position in the texture package file.

### 3.1.1 Syntax

```
s16 GetTextureIndex(
const void* pTexPackage,
const char* texPath
);
```

### 3.1.2 Arguments

- ***pTexPackage*** [in] Pointer to the start of texture package.
  ***texPath*** [in] Path to the texture file.

### 3.1.3 Return Values

The texture index value. Returns -1 if nothing is found in the specified path.

### 3.1.4 Function to Get a Texture

The `GetTexture` function gets a pointer to the texture data. The value of the ***mipLevel*** argument includes the main texture.

### 3.1.5 Syntax

```
void* GetTexture(
          s32* mipLevel,
          u32* mipmapSize,
          const void* pTexPackage,
          const s16 texIndex
);
```

### 3.1.6 Arguments

| | | |
|---|---|---|
| ***mipLevel*** | [out] | Number of mipmap levels (must be 1 or higher). |
| ***mipmapSize*** | [out] | Sizes of each mipmap. (Specify an array.) |
| ***pTexPackage*** | [in] | Pointer to the start of the texture package. |
| ***texIndex*** | [in] | Texture file index value. |

### 3.1.7 Return Values

Returns a pointer to the texture data. Returns ‘0’ if nothing matches the index value.

## 3.2 Function to Get Texture Information

Gets information for a single texture.

### 3.2.1 Syntax

```
bool GetTextureInfo(
          CtrTextureInfo* pTexInfo
          const void* pTexPackage,
          const s16 texIndex,
);
```

### 3.2.2 Arguments

| | | |
|---|---|---|
| ***pTexInfo*** | [out] | Pointer to structure for the texture file information. |
| ***pTexPackage*** | [in] | Pointer to start of texture package. |
| ***texIndex*** | [in] | Texture file index value. |

### 3.2.3 Return Values

Returns `true` if there is an entry at the specified index value. The ***pTexInfo*** argument points to the texture information upon return.

Returns `false` if there is no matching entry for the index value or if an error occurred.

## 3.3 Function to Check the Texture Package Header (helper function)

Checks whether the texture package file can be used. (Checks the magic code and file format version number.)

### 3.3.1 Syntax

```
bool CheckTexturePackageHeader(
          const void* pTexPackage,
);
```

### 3.3.2 Arguments

| | | |
|---|---|---|
| ***pTexPackage*** | [in] | Pointer to start of the texture package. |

### 3.3.3 Return Values

Returns `true` if the version number and other data are valid.

Returns `false` if the data is invalid.