

# Nintendo NITRO-System

## Texture File Supplemental Data

Version 1.0.0

**The contents in this document are highly  
confidential and should be handled accordingly.**

**Confidential**

These coded instructions, statements, and computer programs contain proprietary information of Nintendo of America Inc. and/or Nintendo Company Ltd. and are protected by Federal copyright law. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

## Contents

---

1	Introduction .....	5
2	Structure of the Supplemental Data.....	6
3	Data Blocks .....	7
3.1	Types of Data Blocks.....	7
3.2	Regarding the Size of Textures .....	8
3.3	Adobe Photoshop-Specific Data .....	8
3.3.1	The Compression Method for nns_cidx.....	9
4	Restrictions on TGA Files.....	11
4.1	The ID Field .....	11
4.2	Image Data .....	11
5	Restrictions on PIC Files .....	12
5.1	The Comments Region.....	12
5.2	Image Data .....	12

## Tables

---

Table 1-1 Overview of Texture Files .....	5
Table 3-1 The Types of Data Blocks.....	7
Table 3-2 Adobe Photoshop-specific Data.....	9
Table 4-1 The TGA File's ID Field.....	11
Table 5-1 The PIC File's Comments Region.....	12

## Figures

---

Figure 2-1 Supplemental Data Blocks .....	6
---	---

## Revision History

Version	Revision Date	Description
1.0.0	8/22/2005	Initial release.

# 1 Introduction

NINTENDO NITRO-System uses files that have supplemental NITRO-specific data added to Targa (TGA) files or Softimage PIC files. These files are used as texture files to be applied to 3D models.

This supplemental data contains texture format information, texel data, and texture palette data.

The 3D CG tool's Intermediate File Output plug-in reads this supplemental data if it is included in the texture file and outputs it to an intermediate file.

An overview of the content of texture files is shown in Table 1-1 below.

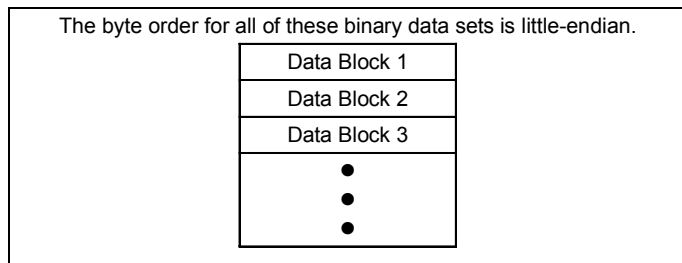
**Table 1-1 Overview of Texture Files**

<b>TGA file</b>	<b>PIC file</b>
TGA Header	PIC Header
TGA Image Data	PIC Image Data
Supplemental Data	Supplemental Data

## 2 Structure of the Supplemental Data

In the supplemental data, a number of data blocks with various kinds of binary data are included, as shown in Figure 2-1 below.

**Figure 2-1 Supplemental Data Blocks**



Data blocks begin with a data block header with the structure shown below. This data block header is 12 bytes in size and stores the data block size and an identifier that indicates the type of the data block.

**Table 2-1 The Data Block Header**

Type	Description	Size
char[8]	Identifier that indicates the type of the data block. This always stores 8 ASCII characters.	8 bytes
u32	The size of the data block. This size includes the 12-byte size of the data block header. This size does not need to be aligned to an integral multiple of a specified number of bytes.	4 bytes

In this manual, data types are indicated with the following notation:

- u8: 8-bit unsigned integer
- u16: 16-bit unsigned integer
- u32: 32-bit unsigned integer

## 3 Data Blocks

### 3.1 Types of Data Blocks

Table 3-1 shows the different types of data blocks.

Some data blocks are always present and others are present only under certain conditions, as shown in Table 3-1.

The `nns_frmt` block always appears first, and the `nns_enbd` block always appears last. However, the rest of the data blocks can appear in any order.

String type data does not need to be null-terminated.

**Table 3-1 The Types of Data Blocks**

Identifier	Data Type	Description
<code>nns_frmt</code>	String	Texture format name. <b>palette4</b> : 4-color palette texture <b>palette16</b> : 16-color palette texture <b>palette256</b> : 256-color palette texture <b>tex4x4</b> : 4x4 texel compressed texture <b>a3i5</b> : A3I5 translucent texture <b>a5i3</b> : A5I3 translucent texture <b>direct</b> : Direct color texture
<code>nns_c0xp</code> <b>Note</b> : Exists when using color0 as the transparent color with 4/16/256-color palette texture	No data	This data block is only a header. When this data block type exists, the palette's color0 is used as the transparent color. When this data block type does not exist, the palette's color0 color information is used as a color.
<code>nns_txel</code>	u16 array or u32 array	Texel data. u32 array for 4x4 texel compressed texture; u16 array for all else.
<code>nns_pidx</code> <b>Note</b> : Exists for 4x4 texel compressed texture	u16 array	Texture palette index data.
<code>nns_pnam</code> <b>Note</b> : Exists except when using direct color texture	String	The name of the texture palette. The intermediate file output plug-in uses this name to determine whether the palette is shared or not. When the name is null characters (the size, excluding the header, is 0), the intermediate file output plug-in automatically determines the texture palette name from the name of the texture file. <b>Note</b> : Full-width characters, half-width kana characters, and double quotation marks (") cannot be used.

Identifier	Data Type	Description
nns_pcol <b>Note:</b> Exists except when using direct color texture	u16 array	Texture palette data. The number of colors in the palette is equal to the size (excluding the header) divided by 2. The number of palette colors is fixed to 4 for 4-color palette textures. Otherwise, the number of palette colors is a multiple of 8.
nns_gnam	String	The name of the tool that output the texture file. Example: "Adobe Photoshop"
nns_gver	String	The version of the tool that output the texture file. Example: "6.0 (Win)"
nns_pshp <b>Note:</b> Exists when file is output from Adobe Photoshop	A set of data blocks	Adobe Photoshop-specific data. For details, see the explanation in Section 3.3 Adobe Photoshop-Specific Data.
nns_imst <b>Note:</b> Exists when file is output from iMageStudio for NITRO	A set of data blocks	Data specific to iMageStudio for NITRO.
nns_endb	No data	This data block is only a header. This data block indicates the end of the supplemental data. When analyzing the supplemental data, skip any data that might exist after this data block.

## 3.2 Regarding the Size of Textures

The upper limit for the height and width of a texture is 1024 texels.

If the texture height or width is not an acceptable NITRO size (8, 16, 32, 64, 128, 256, 512 or 1024), then the texel colors on the right and bottom are used to make the data compatible for NITRO use, and this data is stored in the texel data and texture palette data in the supplemental data. In this situation, the selected NITRO compatible size is the size that is larger than the current width and height and is the closest to the original size of the texture.

For example, if the width of the texture is 20, then 12 texels of color on the right side get added to create a width of 32, and this data is stored in the texel data and texture palette data in the supplemental data. Note, however, that data with a width of 20 is stored in the image data portion of the TGA or PIC file.

## 3.3 Adobe Photoshop-Specific Data

Adobe Photoshop-specific data includes both data unique to Adobe Photoshop that cannot be stored in TGA and PIC files and the options for creating 4x4 texel compressed textures.

The intermediate file output plug-in skips over this data block.

The data in this data block comprises a set of data blocks. The format for the data block header is the same as that explained in Section 2 Structure of the Supplemental Data

The `nns_pver` block always appears first, but the other data blocks can appear in any order.



**Table 3-2 Adobe Photoshop-Specific Data**

Identifier	Data Type	Description
nns_pver	String	The version of the Adobe Photoshop-specific data. Currently, this is set to 1.0.
nns_gray <b>Note:</b> Exists when Photoshop's image mode is set to grayscale.	No data	This data block is only a header. If this data block exists, the image mode becomes grayscale when a texture file is read into Photoshop.
nns_ctbl <b>Note:</b> Exists when Photoshop's image mode is set to index color.	u8 array	Photoshop color table data. One-color 24-bit color data (8 bits for R, G and B) for each color is stored in the order of RGB. The number of colors in the color table is equal to the size, excluding the header, divided by three. If this data blocks exists, the image mode becomes index color when a texture file is read into Photoshop.
nns_cidx <b>Note:</b> Exists when Photoshop's image mode is set to index color.	u8 array	Color index data for each texel in Photoshop. Run-length encoded data is stored here. (Details are explained in the next section.)
nns_xpid <b>Note:</b> Exists when Photoshop's image mode is set to index color and there are transparent colors.	u8 (1 byte)	Transparent color index in Photoshop's color table (0 to 255). If no transparent color has been specified, this data block will not exist.
nns_t44o <b>Note:</b> Exists when set to 4x4 texel compressed texture	u8[5] (5 bytes)	Options for creating 4x4 texel compressed textures. <b>1st byte (linear interpolation flag)</b> 0x00: No linear interpolation 0x01: Use linear interpolation <b>2nd to 4th bytes (palette compression coefficients)</b> Range for sharing colors in R, G, and B channels for palette compression. Takes 1 byte each in the range of 0x00 to 0x1f for R, G, and B (in that order) 0x00 is highest image quality without palette compression, and 0x1f is lowest image quality with largest rate of palette compression. <b>5th byte (filter flag)</b> 0x00: 4x4 texel compressed texture filter not applied 0x01: 4x4 texel compressed texture filter is being applied

### 3.3.1 The Compression Method for nns\_cidx

The original data uses one byte for each texel, arranged in order starting from the top to bottom, and left to right of the image.

The compressed data is compressed line by line using run-length encoding. Each compressed line consists of a control byte and a group comprising one or more sets of texel data. The highest bit of the

control byte indicates whether there are contiguous sets of the same texel data, while the lower 7 bits indicate a value equivalent to the group length – 1. If the highest bit of the control byte is 1, then there is only one set of texel data, and that texel data continues for the length of the group. If the highest bit of the control byte is 0, the number of texel data sets is the same as the group length, and the texel data is used as is.

## 4 Restrictions on TGA Files

### 4.1 The ID Field

A TGA file can store an ID field of 0 to 255 bytes (of any data) after the 18-byte TGA header.

For a TGA file to which supplemental data has been added, the start of this ID field contains the following 20 bytes of data:

**Table 4-1 The TGA File's ID Field**

Type	Description	Size
char[16]	The identifier and the version of the supplemental data. At the present time, this is set to NNS_Tga Version 1.0 (the remaining 1 byte takes 0x00). The intermediate file output plug-in uses the identifier to determine whether the supplemental data exists.	16 bytes
u32	The offset from the start of the supplemental data file (little-endian).	4 bytes

The size of the ID field is specified by the 1st byte in the TGA header.

### 4.2 Image Data

The TGA file image data can have a number of different color modes, such as RGB, index color and grayscale. However, the NINTENDO NITRO-System only uses TGA file image data in RGB format. This is because some 3D CG tools cannot use TGA files if a mode other than RGB is employed.

Even if the graphics tools you use offer index color and grayscale modes, be sure to output your files in RGB mode.

The bits-per-pixel value can be set to either RGB 16 bits, RGB 24 bits or RGBA 32 bits.

For the compression method, specify either no compression or run-length encoding.

## 5 Restrictions on PIC Files

### 5.1 The Comments Region

---

The PIC header in the PIC file has an 80-byte region for comments, and this can be used to store data.

For a PIC file to which supplemental data has been added, add 20 bytes of data as indicated below at the start of this comments region:

**Table 5-1 The PIC File's Comments Region**

Type	Description	Size
char[16]	The identifier and the version of the supplemental data. At the present time, the version is set to NNS_Pic Version 1.0 (the remaining 1 byte takes 0x00). The intermediate file output plug-in uses the identifier to determine whether the supplemental data exists.	16 bytes
u32	The offset from the start of the supplemental data file (little-endian).	4 bytes

The comments region begins from the 9th byte in the PIC header.

### 5.2 Image Data

---

In the image data of PIC files used by NINTENDO NITRO-System, the first channel is the RGB channel.

If an alpha channel exists, it is stored after the RGB channel.

A channel has 8 valid bits.

The only compression method is mixed run-length encoding.

Adobe and Photoshop are registered trademarks or trademarks of Adobe Systems Incorporated.

OPTiX, web technology and iMageStudio are registered trademarks or trademarks of web technology Corp.

Softimage is a registered trademark or trademark of the U.S. company Avid Technology Inc.

All other company names, product names, etc., are registered trademarks or trademarks of the companies to which they belong.

© 2004-2005 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed or loaned in whole or in part without the prior approval of Nintendo.