



# ntexconv Manual

Version 1.3.0

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

# Contents

1	Overview .....	5
2	Procedures for Displaying Image Files for Texture on a NITRO LCD .....	6
3	ntexconv Specifications .....	7
3.1	Image Files .....	7
3.2	NITRO Texture File (NTF) .....	8
3.2.1	NTF Texture Format Types .....	8
3.2.2	NTF File Formats .....	8
3.3	NITRO BG File (NBF) .....	8
3.3.1	NBF BG Format Types .....	8
3.3.2	NBF File Formats .....	9
3.3.3	NBF Output Restrictions .....	9
4	Command Line .....	10
4.1	Command Line Format .....	10
4.2	Command-Line Options for NTF Output .....	10
4.3	Command-Line Options Used When Outputting BG .....	13
4.4	Examples of Command Line Execution for NTF Output .....	15
4.5	Examples of Command-Line Operations for NBF Output .....	16
4.6	Examples of Command-Line Operations when Outputting NTF and NBF at the Same Time .....	17
5	Recommended Conversion Procedures .....	18
5.1	Converting NTF .....	18
5.1.1	Converting to palette4, palette16, or palette256 .....	18
5.1.2	Converting to a5i3 and a3i5 .....	18
5.1.3	Converting to direct .....	18
5.1.4	Converting to tex4x4 .....	18
5.2	Converting NBF .....	19
5.2.1	Converting to palette16, palette256, affine, or palette256x16 .....	19
5.2.2	Converting to direct .....	19
5.2.3	Converting to palette256mp or big256mp .....	19
6	Cautions Concerning NTF and NBF .....	20
6.1	Cautions Concerning Bit Number Differences for Color Values .....	20
6.2	File Name Restrictions .....	20
7	OBJ File Output .....	21
7.1	Character OBJ .....	21
7.2	Bitmap OBJ .....	21
8	Creating Icon Images .....	22
8.1	Icon Images .....	22
8.2	Icon Image Creation Command Examples .....	22

## Tables

---

Table 3-1	List of File Types that ntexconv can Handle.....	7
Table 3-2	Restrictions on Files that ntexconv can Handle .....	7
Table 3-3	List of NTF Texture Format Types.....	8
Table 3-4	NTF File Formats.....	8
Table 3-5	NBF BG Format Types.....	9
Table 3-6	NBF File Formats.....	9
Table 3-7	NBF Output Restrictions .....	9
Table 4-1	Converted Texture Format and Option File Image Format.....	11
Table 4-2	Input File Image Format and Converted Texture Format .....	11
Table 4-3	Input Image File Format and Converted Background Format .....	13
Table 4-4	Texture Format and Converted BG Format.....	13

## Figures

---

Figure 2-1	Overall Procedure .....	6
------------	-------------------------	---

## Revision History

Version	Revision Date	Description
1.3.0	7/01/2005	<ul style="list-style-type: none"><li>• Updated ntexconv source code<ul style="list-style-type: none"><li>• Support for color palettes was changed from RGB only to RGBA.</li><li>• Added <code>-bgnap</code>.</li><li>• When converting from direct to a3i5/a5i3, you can now select <code>-ta</code>, <code>-tc</code>.</li></ul></li><li>• Added description of <code>-bgnap</code> (p15).</li><li>• Added a3i5,a5i3 to description of <code>-tc,-ta</code> (p13).</li><li>• Added new chapter, "Creating icon images" (p25).</li></ul>
1.2.0	8/02/2004	<ul style="list-style-type: none"><li>• Updated ntexconv source code.<ul style="list-style-type: none"><li>• Added <code>-bgof</code>.</li></ul></li><li>• Revised description of <code>-bgp</code>. (p15)</li><li>• Added description of <code>-bgof</code>. (p15)</li></ul>
1.1.1	6/30/2004	<ul style="list-style-type: none"><li>• Added and revised command line execution examples (p. 18)</li><li>• Revised ntexconv</li><li>• Corrected the bug that caused incorrect color subtraction when <code>-ta</code> was not specified during conversion from direct to a3i5 and a5i3</li></ul>
1.1.0	4/20/2004	<ul style="list-style-type: none"><li>• Corrected errors.</li><li>• Changed the header from NITRO to NITRO-SampleTools.</li><li>• Changed graphic file format from <code>.jpeg</code> to <code>.visio</code> (p. 7, Fig. 2-1).</li><li>• Changed the order of the <code>-bgnc</code>, <code>-bgnv</code>, and <code>-bgnh</code> explanations.</li></ul>
1.0.0	3/5/2004	<ul style="list-style-type: none"><li>• Added BG output function.</li><li>• Added a3i5 to Texture Format.</li></ul>
0.5.0	1/30/2004	<ul style="list-style-type: none"><li>• Initial release.</li></ul>

# 1 Overview

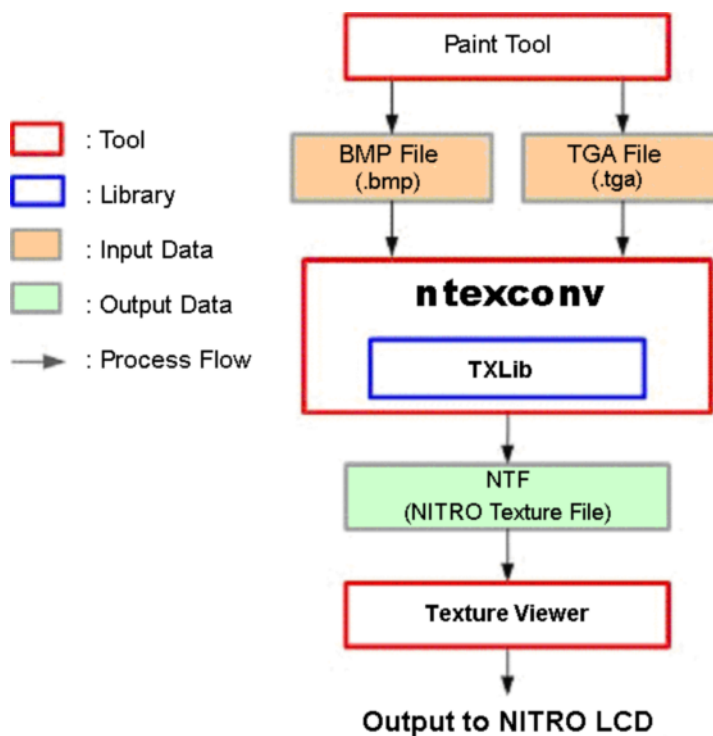
The ntexconv tool creates texture data and BG data files suitable for the NITRO program from BMP (`.bmp` extension) and TGA (`.tga` extension) image files. You can choose to output texture data files and BG data files in either binary or text format. The texture data file output from the ntexconv is referred to as “NTF” (NITRO Texture File) and the BG data file is referred to as “NBF” (NITRO BG File).

## 2 Procedures for Displaying Image Files for Texture on a NITRO LCD

The following is a description of the procedures for converting BMP and TGA image files for textures to NTF and then displaying them on a NITRO LCD. In the diagram, TextureViewer is a sample program for displaying NTF on a NITRO LCD. It was made using the NITRO-SDK. Like ntexconv, it is included in TXLib.

1. Start the paint tool and create an image file for texture.
2. Output the image file for texture in BMP or TGA format from the paint tool.
3. Start ntexconv, convert the image file to NTF, and output.
4. Input into Texture Viewer the NTF that you output, and use make to confirm the image on a NITRO LCD.

Figure 2-1 Overall Procedure



## 3 ntexconv Specifications

### 3.1 Image Files

The following table lists the image files that ntexconv can handle.

**Table 3-1 List of File Types that ntexconv can Handle**

File Type	Image Format	Content
BMP	palette2	2-color color index
	palette16	16-color color index
	palette256	256-color color index
	24bit direct	Color image with 8 bits each of RGB
TGA	gray	Monochrome (8-bit brightness index)
	palette256	256-color color index
	16bit direct	Color image with 5 bits each of RGB and 1 bit of alpha
	24bit direct	Color image with 8 bits each of RGB
	32bit direct	Color image with 8 bits each of RGB and 8 bits of alpha

**Table 3-2 Restrictions on Files that ntexconv can Handle**

File Type	Restriction
BMP	Cannot handle RLE compressed files.

## 3.2 NITRO Texture File (NTF)

With ntexconv, you can output as NTF all texture formats used in NITRO. For a detailed explanation of NTF file contents, see the separate `TXLib_manual.pdf`. Here the contents are simply listed in a table.

### 3.2.1 NTF Texture Format Types

The following table lists NTF texture format types.

**Table 3-3 List of NTF Texture Format Types**

Texture Format	Content
palette4	4-color palette texture
palette16	16-color palette texture
palette256	256-color palette texture
tex4x4	4x4 texel compressed texture
a3i5*	a3i5 translucent texture (32-color palette + 3 bit alpha)
a5i3	a5i3 translucent texture (8-color palette + 5 bit alpha)
direct	Direct color texture

\* a3i5 is a newly added texture format. For this reason, it can be output as NTF but it cannot be displayed with the TEG version of NITRO.

### 3.2.2 NTF File Formats

The following table lists NTF file formats.

**Table 3-4 NTF File Formats**

File Format	File name
Binary format	Texel data: (file name).ntft
	Palette data: (file name).ntfp
	Palette index data: (file name).ntfi
Text format	(file name).c

## 3.3 NITRO BG File (NBF)

With ntexconv, you can output the BG format used in NITRO as NBF.

### 3.3.1 NBF BG Format Types

The following table lists NBF BG format types.



**Table 3-5 NBF BG Format Types**

BG Format	Content
palette16	Text BG 16-color mode
palette256	Text BG 256-color mode
affine	Affine BG
palette256x16	256-color x 16 palette character BG
palette256Bmp	256-color bitmap BG
direct	direct color bitmap BG
big256Bmp	large screen 256-color bitmap BG

### 3.3.2 NBF File Formats

The following table lists NBF file formats.

**Table 3-6 NBF File Formats**

File Format	File Name
Binary Format	Character data: <i>filename.nbfc</i>
	Palette data: <i>filename.nbfp</i>
	Screen data: <i>filename.nbfs</i>
Text Format	<i>filename.c</i>

### 3.3.3 NBF Output Restrictions

The following table lists restrictions when outputting NBF.

**Table 3-7 NBF Output Restrictions**

BG Format	Restriction
palette16	Up to 1024 characters. An error will occur when exceeding 1024 characters.
palette256	Screen data order is the same as affine: data order is left to right and the lines are top to bottom. You must adjust screen data in the program when you make a BG exceeding 256 x 256.
palette256x16	Does not support NBF output that is not NITRO BG size (multiples of 8 for both vertical and horizontal).
affine	Up to 256 characters. An error will occur when exceeding 256 characters. Does not support NBF output that is not NITRO BG size (multiples of 8 for both vertical and horizontal).

## 4 Command Line

### 4.1 Command Line Format

---

The command line for executing ntexconv is as follows.

```
ntexconv input image file name [options]
```

Recognize and process a command line in ntexconv as follows:

- No order is specified for the options or the input image file name. You can specify them in any order you like.
- When conflicting options are specified, the last specified option takes effect.
- If option specifications are wrong or if an error occurs during execution, an error will be displayed, after which a help menu will appear and the program will terminate.

### 4.2 Command-Line Options for NTF Output

---

-v	Displays conversion status in detail. Specify this option together with -s below and use it as a detailed display of script operation contents.
-h	Displays the help screen. If you use this option, the program ignores all other options, and terminates the program after it displays the help screen.
-s <i>filename</i>	This option converts according to command lines that are in a specified script file. If you use this option, the program ignores options other than -tv and -v and executes the script file command lines sequentially from the top of the file. Specifying this option in a script file will cause an error.
-tv	Outputs data (textureData.c and textureData.h) for TextureViewer. If you insert these files and an NTF into TextureViewer, you can easily preview on a NITRO LCD.  The data output destination for TextureViewer data is the directory where the script file is when executing from a script, or the directory where the output NTF is when executing without a script. Specifying this option in the script file is invalid.
-o <i>filename</i>	Specifies the name of the NTF to output. Regardless of the extension, the file format is automatically converted to text format (.c extension). If this option is omitted, the input image file extension will be changed to .c and it will be output in text format. However, if you specify the option, you cannot omit a file name. Always specify a file name. See section 6.2 for details on file name restrictions.
-b	Outputs an NTF in binary format. If you do not specify this option, output will be in text format.
-d <i>filename</i>	Specifies the name of an option file for output. This option file is used to view the quality of the converted image on a PC. After conversion, the image is output to this file in BMP or TGA format. The file extension must be .bmp or .tga. Depending on the texture format of NTF, the option file is output as shown in Table 4-1.

**Table 4-1 Converted Texture Format and Option File Image Format**

NTF Texture Format	Image format of the output option file	
palette4	BMP	16-color color index
palette16		16-color color index
palette256		256-color color index
tex4x4		Color image with 8 bits each of RGB
a3i5		256-color color index
a5i3		16-color color index
direct		Color image with 8 bits each of RGB
palette4	TGA	256-color color index
palette16		256-color color index
palette256		256-color color index
tex4x4		Color image with 8 bits each of RGB and 8 bits of alpha
a3i5		Color image with 8 bits each of RGB and 8 bits of alpha
a5i3		Color image with 8 bits each of RGB and 8 bits of alpha
direct		Color image with 8 bits each of RGB and 8 bits of alpha

**-f *format name*** Selects the conversion texture format (palette4, a5i3, palette16, a3i5, palette256, direct, or tex4x4). If this option is not specified, the image will be converted to the texture format that is closest to the input image file's image format. See below for details.

**Table 4-2 Input File Image Format and Converted Texture Format**

Input file image format		Output texture format
BMP	2-color color index	palette4
	16-color color index	palette16
	256-color color index	palette256
	Color image with 8 bits each of RGB	direct
TGA	Grayscale (8-bit brightness index)	direct
	256-color color index	palette256
	Color image with 5 bits each of RGB and 1 bit of alpha	direct
	Color image with 8 bits each of RGB	direct
	Color image with 8 bits each of RGB and 8 bits of alpha	direct

**-p *filename*** When converting to textures with palettes other than tex4x4 (palette 4, palette16, palette256, a5i3, a3i5), this specifies use of the palette of the named image file. The specified image file must have a palette. Also, use caution as the extra colors are cut when the number of colors in the palette is greater than the number of colors in the texture palette you are converting to.

Example: If an image file with a 256-color palette is specified to a palette and converted to palette16, only the first 16 colors in the palette are used and the rest are cut.

- nal Does not adjust to NTF image size. If this option is not indicated, even if an image file not matching the NTF image size is input, the size is adjusted when outputting the NTF. Specify this option when you want to reduce the size of the NTF file and you do not want to include the extra data.  
See the separate `TXLib_manual.pdf` for details on adjusting NTF size.
- tc=r,g,b Assigns palette color 0 to transparent using the transparent color. r, g, and b represent the elements of the transparent color and must be specified in the range of 0(0x00) to 255(0xff).  
This option is only effective when the input image file format is direct and output texture format is palette4, palette16, palette256, a3i5, or a5i3.
- ta Assigns palette color 0 to transparent using alpha. The program creates transparency using alpha values greater than or equal to 0x80 as opaque, and values less than 0x80 as transparent.  
This option is only effective when the input image file is direct and the output texture format is palette4, palette16, palette256, a3i5, or a5i3.

### Additional Options for tex4x4

When you specify `tex4x4` with `-f`, you can also choose options that determine how you convert to `tex4x4`. Following is a description of these options.

- a0 The conversion algorithm converts "Fast. Without linear interpolation."
- a1 The conversion algorithm converts "Fast. With linear interpolation."
- a2 The conversion algorithm converts "Round robin. Without linear interpolation."
- a3 The conversion algorithm converts "Round robin. With linear interpolation."
- ta Specifies a transparent color mode in which conversion is performed using alpha-based transparency. The program creates transparency using alpha values greater than or equal to 0x80 as opaque, and values less than 0x80 as transparent. If you do not specify this option or the following `-tc=r,g,b` option, conversion will be in 4-color mode.
- tc=r,g,b Specifies a transparent color mode in which conversion is performed using transparent color-based transparency. Transparent color is expressed in r,g,b. Set in the range 0(0x00) – 255(0xff).
- cp=r,g,b Specifies the compression level for number of `tex4x4` palettes. A `tex4x4` palette is one that is used in a 4x4 texel block. The palette has two colors when linear interpolation is used and four colors when linear interpolation is not used. The common color range is expressed in r,g,b. Set in the range of 0(0x00) – 31(0x1f) because the program internally multiplies each element value by 8. If "0" is specified for all of these values, only `tex4x4` palettes that have identical colors will be shared. This will reduce the number of `tex4x4` palettes.
- ncp Do not compress the number of `tex4x4` palettes.

Even if these options are not specified (initial state), the following options are automatically selected.

- a2 The conversion algorithm converts "Round robin. Without linear interpolation."
- cp=0,0,0 Only `tex4x4` palettes with the same colors are shared and the number of `tex4x4` palettes is compressed.

Also, since `-ta` and `-tc` are not specified in the initial state, conversion is performed in 4-color mode.

## 4.3 Command-Line Options Used When Outputting BG

### Options

- `-no` Do not output NTF. Specify this option when you want to output only NBF.
- `-bg` Outputs NBF.
- `-bgb` Outputs NBF in binary format. If this option is not specified, output will be in text format.
- `-bgo filename` Specifies the NBF file name to output. Regardless of the extension, the file format will be automatically converted to text format ( `.c` extension). If this option is omitted, the input image file extension will be changed to `.c` and it will be output in text format. However, if you specify this option, you cannot omit the file name. Always specify a file name. For details on file name restrictions, see Section 6.2 – File Name Restrictions.
- `-bgf format name` Selects the BG format to convert (palette 16, palette 256, affine, palette 256x16, palette 256Bmp, direct, or big256Bmp). If this option is not specified, the BG format is converted to the image format closest to the input image file format as described below. Also, if the texture format is indicated with `-f` when outputting NTF, you can only output NBF when that format is palette 16, palette 256, or direct format. The table below shows the correspondence.

**Table 4-3 Input Image File Format and Converted Background Format**

Input Image File Format		Output BG Format
BMP	2-color color index	error
	16-color color index	palette16
	256-color color index	palette256
	Color image with 8 bits each of RGB	direct
TGA	Black and white (8 bit brightness index)	direct
	256-color color index	palette256
	Color image with 5 bits each of RGB, 1 bit of alpha	direct
	Color image with 8 bits each of RGB	direct
	Color image with 8 bits each of RGB, 8 bit alpha	direct

**Table 4-4 Texture Format and Converted BG Format**

Texture Format	Output BG Format
palette4	error
palette16	palette16
palette256	palette256
tex4x4	error
a5i3	error
a3i5	error
direct	direct

- bgp *palette number*** Specifies the number of the color palette that is applied to the screen data. If this option is not specified, palette number defaults to "0."  
This option is only available when converting to palette16 or palette256x16.
- bgnap** Outputs only the number of palettes that are actually being used.  
If this option is not specified, the number of color palettes will be forcibly adjusted and the output palettes will be those for 16 colors (when outputting palette16) or 256 colors (when outputting palette256, palette256x16 or affine). The additional palette colors will be black.  
This option is only valid when the BG format to output is palette16, palette256 or palette256x16, or when converting to affine.
- bgof *offset value*** Specifies the color index offset value that is applied to the character data. If this option is not used, the offset value defaults to "0".  
This option is only available when you are converting the output BG to palette16, palette256, palette256x16, or affine.  
If the offset causes the color index value to exceed the maximum, the color index value defaults to the maximum. Note that no error or warning will be generated.  
For example, if the output BG format is palette16 and you specify -bg of 10, if the original color index value is 7, it will become 15. This is because the maximum color index value for palette16 is 15.
- bgnc** Do not compress the number of characters. This option is only available when converting to palette16, palette256, or palette256x16.
- bgnv** Do not apply V flip. If this option is not specified, V flip will apply and the number of characters will be compressed.  
This option is only available when converting to palette16, palette256, or palette256x16. Also, this option is not available when -bgnc (see below) is specified.
- bgnh** Do not apply H flip. If this is not specified, H flip will apply and the number of characters will be compressed.  
This option is only available when converting to palette16, palette256, or palette256x16. Also, this option is not available when -bgnc (see below) is specified.
- ta** Assigns palette color 0 with alpha-based transparency. The program creates transparency using alpha values greater than or equal to 0x80 as opaque, and values less than 0x80 as transparent. Palette color 0 is assigned because the BG treats palette color 0 as transparent.  
This option is only available when the input image file is direct and you convert output BG format to a BG in the palette.
- tc=r,g,b** Assigns palette color 0 with transparent color-based transparency. r, g, and b represent the elements of the transparent color and must be specified in the range of 0(0x00) – 255(0xff).  
This option is only available when the input image file is direct and you convert to an output BG format in the palette.

## 4.4 Examples of Command Line Execution for NTF Output

Example 1: `ntexconv direct.tga`

Reads a file named `direct.tga`, converts it to NTF, and then outputs it. Because no NTF output destination is specified, it will output to the same directory as `direct.tga` using the file name `direct.c`.

Example 2: `ntexconv direct.tga -o tga_to_direct.c`

This example specifies an NTF output target for example 1. It will output a file named `tga_to_direct.c`.

Example 3: `ntexconv direct.tga -o c:\ tmp\directToPalette256.c -f palette256`

This example specifies a conversion texture for example 2. The input image file will be converted to `palette256`, and output as a file named `directToPalette256.c`.

Example 4: `ntexconv direct.tga -p forPalette256.bmp -f palette256`

This is an example of specifying the palette and outputting an NTF file. In this case, it makes the color index using the `forPalette256.bmp` palette having the 256-color palette, and outputs a file named `direct.c`.

Example 5: `ntexconv direct.tga -tc=0xff,0xff,0xff -f palette 256`

This example specifies the transparent color and outputs NTF. The specified color goes into palette color 0.

Example 6: `ntexconv direct.tga -d after.bmp`

This example outputs an option file.

Example 7: `ntexconv direct.tga -f tex4x4 -a3`

This specifies which type of conversion to use when converting to `tex4x4`. This example specifies conversion algorithm 3 – “Round robin. With linear interpolation.”

Example 8: `ntexconv direct.tga -f tex4x4 -ta`

This specifies the transparent color mode when converting to `tex4x4`. In this case, transparency is created using alpha. If there is no alpha in the input image file that is the conversion source, no transparency will be performed.

Example 9: `ntexconv direct.tga -f tex4x4 -tc=255,0,0`

This specifies the color transparency mode when converting to `tex4x4`. In this example, transparency is created using the transparent color. In the example, the transparent color is set to red.

Example 10: `ntexconv -s testScript.txt -v -tv`

This is an example of executing using a script file. If the script file has been stored in text format, the extension is irrelevant. Because `-v` is specified in the example, conversion details will be displayed on the PC screen.

Because `-tv` is specified, data for TextureViewer (`textureData.c` and `textureData.h`) is output. It is output to the script file's directory.

You can see the script file (`testScript.txt`) contents below.

```
// Convert direct.tga into each texture format.
direct.tga      -o directToPalette4.c  -f palette4
direct.tga      -o directToA5i3.c      -f a5i3
direct.tga      -o directToPalette16.c -f palette16
```

Be careful of the following things when making a script file:

- In the script file, write one image conversion command line per line.
- Command lines in a script file omit the execute program name (`ntexconv`).

The following are permitted in a script file.

- Tabs can be used.
- “//” at the beginning of a line indicates a comment.
- You can input a carriage return at the beginning of a line only.

## 4.5 Examples of Command-Line Operations for NBF Output

---

Example 1: `ntexconv direct.tga -no -bg`

This example outputs only NBF without outputting NTF.

Example 2: `ntexconv direct.tga -f palette256 -no -bg -bgf affine`

This is an example specifying the BG format to convert. To convert to affine, you must first convert to palette256.

Example 3: `ntexconv direct.tga -no -bg -bgo test_bg.c -bgb`

This is an example of specifying the output destination and outputting in binary format.

Example 4: `ntexconv direct.tga -f palette16 -no -bg -bgp 1`

This example specifies palette color number as “1” and then converts to palette16.

Example 5: `ntexconv direct.tga -f palette 256 -p forPalette256.bmp -no -bg`

This example specifies the palette and converts to palette256. In this case, it creates a color index using the `forPalette256.bmp` palette having the 256-color palette.

Example 6: `ntexconv direct.tga -f palette256 -tc=0xff,0xff,0xff -no -bg`

This example specifies the transparent color and outputs BG. The specified color goes into palette color 0.

Example 7: `ntexconv direct.tga -f palette256 -bgnv -bgnh -no -bg`

This example converts to palette256 without flipping. Character count increases because flip does not occur.

Example 8: `ntexconv direct.tga -f palette256 -bgnc -no -bg`

This example converts to palette256 without compressing the number of characters at all.



## 4.6 Examples of Command-Line Operations when Outputting NTF and NBF at the Same Time

---

**Example 1:** `ntexconv direct.tga -bg -bgo direct_bg.c`

Reads the file named `direct.tga`, converts it to NTF and NBF, and then outputs it.

Because no NTF output destination is specified, it will output in the same directory as `direct.tga` using the file name `direct.c`. NBF output target is output with specified file name `direct_bg.c`.

**Example 2:** `ntexconv direct.tga -bg -bgo direct_bg.c -bgb`

This example outputs NTF in text format and NBF in binary format.

**Example 3:** `ntexconv direct.tga -f palette256 -bg`

This example outputs NTF and NBF at the same time and in the same format. However, if you set to `-f` a format (such as `a5i3`) that is not available in NBF, an error will occur.

**Example 4:** `ntexconv direct.tga -f palette256 -bg -bgo directToAffine_bg.c -bgf affine`

This example outputs NTF and NBF at the same time and in different formats. However, if you set to `-f` a format (such as `a5i3`) that is not in NBF, an error will occur.

## 5 Recommended Conversion Procedures

### 5.1 Converting NTF

---

#### 5.1.1 Converting to palette4, palette16, or palette256

---

palette4, palette16, and palette256 textures have color indices. We recommend that you output from the paint tool in color indexed BMP format, and use ntexconv for NTF output.

#### 5.1.2 Converting to a5i3 and a3i5

---

a5i3 and a3i5 textures have color indices and alphas. These are NITRO proprietary texture formats. Therefore, they are difficult formats to convert to NTF while confirming with a normal paint tool. We recommend that you use the paint tool to output 32-bit direct format TGA, and then use ntexconv to convert to a5i3 or a3i5. You can get good results by using the paint tool to reduce to 8 colors for (a5i3) or 32 colors (a3i5) before converting.

#### 5.1.3 Converting to direct

---

A direct texture has a color image and alpha image. We recommend that you output in 32-bit direct TGA from the paint tool, and convert that with ntexconv. At this time, the alpha value will be converted from 8-bit to 1-bit (the 7 lower bits will be cut). Therefore you can get good results if you use the paint tool to binarize the alpha channel beforehand. Although there is also a 16-bit direct TGA format, most paint tools cut the alpha component when they output in this format. Therefore, we recommend that you do not use this if you need an alpha value.

#### 5.1.4 Converting to tex4x4

---

tex4x4 is a NITRO proprietary format. It is a color index texture in which a 4x4 texel is treated as one block, having a palette that is reduced from inside the block to either two or four colors. You cannot confirm using a paint tool. Therefore, we recommend that when confirming the image you first output in direct TGA format, use ntexconv to convert to tex4x4, and then output a BMP or TGA file. For alpha-based transparency in transparent color mode, determination is based on whether the alpha value is greater than 0x80 (opaque) or less than 0x80 (transparent). As with conversion to direct, we recommend that you use the paint tool to binarize the alpha channel beforehand.

See the attached "Description of NITRO 4x4 texel Compressed Texture.pdf" for a detailed description of tex4x4 specification.

## 5.2 Converting NBF

---

### 5.2.1 Converting to palette16, palette256, affine, or palette256x16

---

See 5.1.1 Converting to palette4, palette16, or palette256.

Since these background formats are character backgrounds, exceeding the character count will cause an error. Therefore, you must make the image file while being paying attention to the 8x8 block.

### 5.2.2 Converting to direct

---

See 5.1.3 Converting to direct.

### 5.2.3 Converting to palette256mp or big256mp

---

See 5.1.1 Converting to palette4, palette16, or palette256.

## 6 Cautions Concerning NTF and NBF

### 6.1 Cautions Concerning Bit Number Differences for Color Values

---

NITRO uses as color values 5 bits each of RGB, and 16 bits in total. Therefore, the colors in the color palettes held by palette4 – palette256 will also be 16-bit. Because BMP and TGA for a PC are normally 24-bit (8 bits each of RGB), when output NTF or NTB are displayed on NITRO, that image will be inferior to the original BMP or TGA. Therefore, confirm images on a NITRO machine even if you are using a texture that has the same texture format as the BMP images.

### 6.2 File Name Restrictions

---

The NTF and NBF output file names must be names that can be used when you run make on the compiler. For example, when Japanese is used in the file name, an error will occur when you run make on the compiler. To prevent this kind of error, ntexconv checks whether the file name can be used when make is run on the compiler. If an incompatible file name is used, an error message is displayed and the program quits. The NTF and NBF are not output. The following is a list of characters and symbols available for use in NTF and NBF file names.

Symbols available for use:        \_

Characters available for use:    letters (a-z, A-Z), numerals (cannot be used at beginning of file name)

Examples of file name:

direct_1.c	(Correct.)
direct-1.c	(Incorrect. The hyphen ("–") cannot be used.)

## 7 OBJ File Output

ntexconv does not support OBJ file output as files like NTF and NBF. However, you can use an NBF as an OBJ file. This section explains how to use an NBF as an OBJ file. See the separate “NITRO Programming Manual” for OBJ data structure and OBJ details.

### 7.1 Character OBJ

---

Character OBJ character and palette data are the same as BG (palette16 or palette256) character data and palette data. The difference from text BG is that “character OBJ does not have screen data.” Therefore, use it as a character OBJ file according to the following procedures:

1. Output NBF as palette16 when the BG format is in 16-color mode and as palette256 when in 256-color mode. When outputting, make sure that the input image size is the OBJ size (8x8 to 64x64). Also, specify that you do not want to perform character count compression.
2. Open the NBF that was output and delete screen data unnecessary for OBJ.
3. When inserting that OBJ file into the NITRO program, set the OBJ mapping mode to one-dimensional mapping.

The following is an example of a command for outputting OBJ character:

Example: Making a 256-color mode character OBJ from a 256-color color index image file

```
ntexconv palette256.tga -no -bg -bgnc -bgo charObj_256.c
```

### 7.2 Bitmap OBJ

---

Bitmap OBJ data structure is the same as that of direct color bitmap BG (direct). Therefore, follow the procedure below to use a direct BG as a bitmap OBJ file:

1. Output the NBF with the BG format in “direct.”
2. When inserting that OBJ file into the NITRO program, set the OBJ mapping mode to one-dimensional mapping.

The following is an example of command for outputting a bitmap OBJ:

Example: Making a bitmap OBJ from a direct color image file

```
ntexconv direct.tga -no -bg -bgo bitmapObj.c
```

## 8 Creating Icon Images

### 8.1 Icon Images

---

ntexconv can also be used to make icon images for banners. An icon image is a character image with a 4x4-character (32-dot x 32-dot) **palette16 format**.

### 8.2 Icon Image Creation Command Examples

---

**Example 1:** `ntexconv -no -bg -bgb -bgnc`

This is the simplest example. It does not convert the image format, so an error will occur if the original image file is not converted from a palette16 format.

**Example 2:** `ntexconv -no -bg -bgb -bgnc -f palette16`

This example creates an icon image from a palette16 format image file, reducing the number of colors to 16.

**Example 3:** `ntexconv -no -bg -bgb -bgnc -f palette16 -tc=255,0,0`

This example creates an icon image by converting the image file to 16 colors. The color to eliminate is selected from the image file by specifying `-tc=R,G,B`. The color specified with `-tc` will be entered in Palette 0.

**Example 4:** `ntexconv -no -bg -bgb -bgnc -f palette16 -ta`

This example creates an icon image by converting the image file to 16 colors. The color to eliminate is selected from the image file and pixels in the TGA file with an alpha value of less than 0x80 will be treated as eliminated and Palette 0 will be allocated to them. Black will be entered in Palette 0.



Windows is a registered trademark or trademark of Microsoft Corporation (USA) in the U.S. and other countries.

All other company and product names are the trademark or registered trademark of the respective companies.

© 2003-2005 Nintendo

No part of the contents of this document may be reproduced, copied, transferred, distributed, or given without the permission from Nintendo.