NintendoWare for CTR

NW4C FontConverter

2010/12/02

Version 1.0.2

PROVISIONAL TRANSLATION

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

Confidential

Table of Contents

Tables

Figures

1 Introduction

1.1 About NW4C FontConverter

NW4C FontConverter (hereafter referred to as “FontConverter”) is a Microsoft Windows application for creating font resources (`.bcfnt` files) and archive fonts (`.bcfna` files). These files are used as fonts by NintendoWare for CTR when drawing text.

1.2 Manual Structure

Including this manual, the three manuals associated with FontConverter are:

- `FontConverter_Manual.pdf` – NW4C FontConverter (manual for the GUI version of FontConverter)

  This is the manual you are now reading. This manual describes the GUI (Windows) version of FontConverter, including features specific to the GUI version and features shared with the CUI (command line) version.

- `FontConverterConsole_Manual.pdf` – NW4C FontConverterConsole (manual for the command-line version of FontConverter)

  This manual is for `NW4C_FontConverterConsole.exe`, the command-line version of FontConverter. It describes the features specific to the command-line version. The reader is assumed to have also read `FontConverter_Manual.pdf`.

- `FontConverter_Format.pdf` – FontConverter format (FontConverter I/O file specifications)

  This manual describes the various file formats that are input to and output from FontConverter. Specifically covered are image files (`.bmp` and `.tga`), text filter files (`.xllt`), text order files (`.xlor`), and glyph group files (`.xggp`).

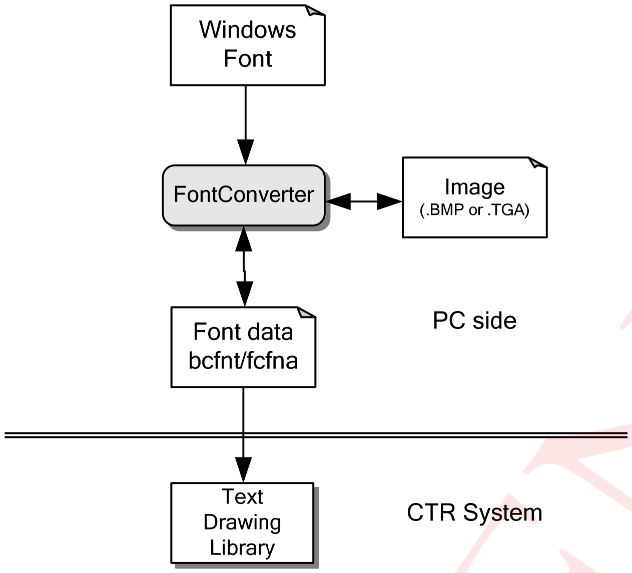
1.3 What FontConverter Can Do

The FontConverter tool creates font resources from Windows fonts. Although this tool can be used to create bcfnt/bcfna files directly from Windows fonts, it can also write Windows fonts as image files in BMP or TGA format. Since the image output can be used as input to create bcfnt/bcfna files, Windows fonts can first be written to an image format, adjusted, and then converted to bcfnt/bcfna. Original fonts can also be created by starting with original image files.

Figure 1-1 illustrates the different elements used to create fonts for the Wii console and how they relate to each other.

In each step of the conversion process, you can create the smallest possible set of fonts by extracting only those characters needed from the original font set.

Figure 1-1 Relationship Between FontConverter and Related Resources



1.4 Font Licenses

Although FontConverter can convert any font installed on a PC for use as a display font on a CTR system, no font licenses are included with FontConverter or NintendoWare for CTR.

You have the responsibility of obtaining a font license to sell software that uses any of these fonts.

2 File Structure

The FontConverter files are located in the `NintendoWare/CTR/tools/FontConverter` directory. Figure 2-1 shows the structure of the files and folders related to FontConverter.

Figure 2-1 FontConverter File Structure

```
NintendoWare/CTR
│  documents
│  └ FontConverter
│      ├ FontConverter_Manual.pdf               // GUI version of FontConverter Manual
│      ├ FontConverterConsole_Manual.pdf        // CUI version of FontConverter Manual
│      └ FontConverter_Format.pdf               // FontConverter I/O File Specifications
└ tools
   │  FontConverter
   │  ├ xllt
   │  │  ├ letter_list.dtd                       // Character Filter File DTD
   │  │  └ sample.xllt                           // Sample File Filter
   │  │  xlor
   │  │  ├ letter_order.dtd                      // Text Ordering File DTD
   │  │  ├ cp1252.xlor                           ┐
   │  │  ├ cp1252_JIS_X0201_X0208_012.xlor       │
   │  │  ├ cp1252_JIS_X0201_X0208_012_94.xlor    │
   │  │  ├ JIS_X0201_X0208_01.xlor               ├ Supplied Text Ordering Files
   │  │  ├ JIS_X0201_X0208_012.xlor              │
   │  │  ├ JIS_X0201_X0208_01_94.xlor            │
   │  │  └ JIS_X0201_X0208_012_94.xlor           ┘
   │  └ xggp
   │     ├ glyph-groups.dtd                      // Glyph Group File DTD
   │     └ sample.xggp                           // Sample Glyph Group File
   │  NW4C_FontConverter.exe                     // FontConverter GUI
   └ NW4C_FontConverterConsole.exe               // FontConverter CUI
```

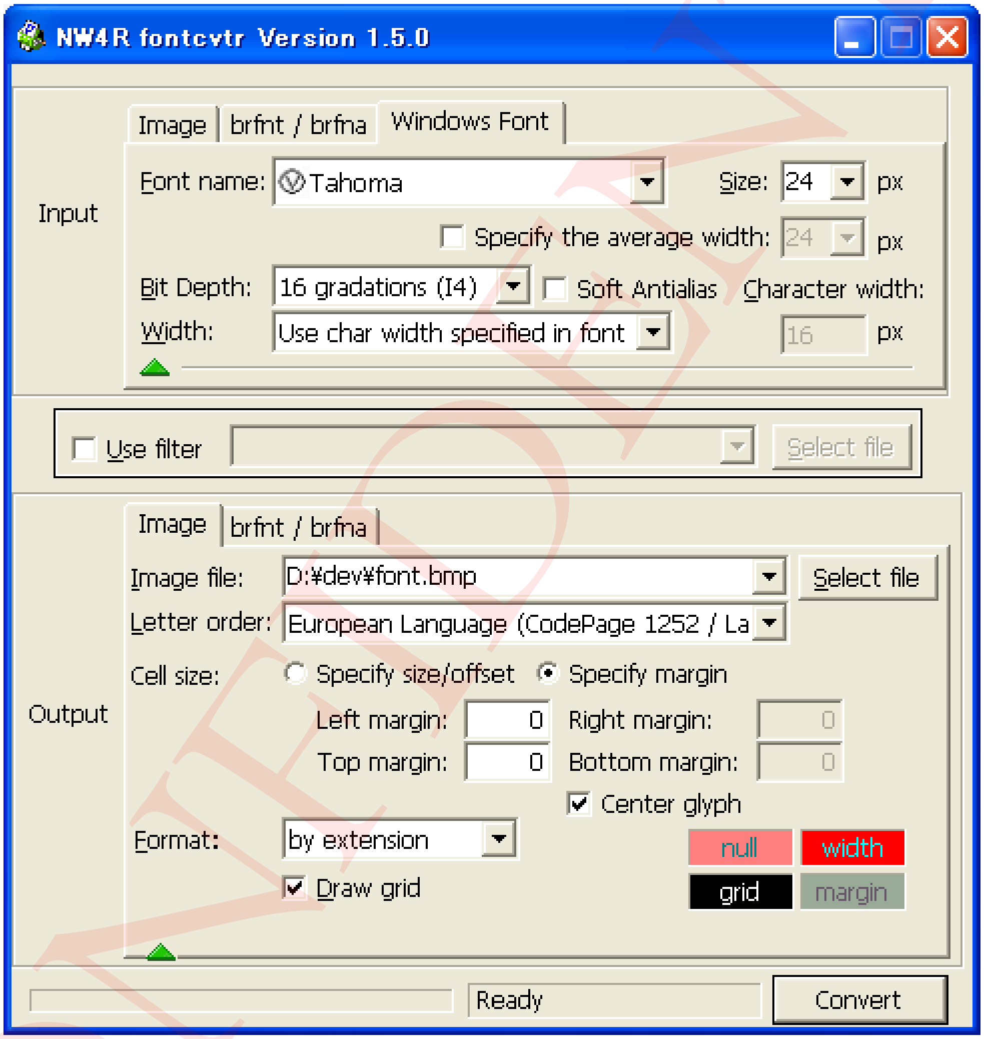
For information on the text filter file, text ordering file, or glyph group file shown in Figure 3-1, see *FontConverter Formats* (`FontConverter_Format.pdf`).

3 Operations

3.1 Graphical User Interface

Figure 3-1 shows the main dialog box in the Graphical User Interface (GUI) version of FontConverter (hereafter referred to simply as “FontConverter”). The FontConverter application is based on dialog boxes. Nearly all FontConverter operations are handled within the dialog box shown in Figure 4-1.

Figure 3-1 Main Dialog Box for FontConverter



NW4R fontcvtr Version 1.5.0

Input

Image | brfnt / brfna | Windows Font

Font name: Tahoma ▾ Size: 24 ▾ px

□ Specify the average width: 24 ▾ px

Bit Depth: 16 gradations (I4) ▾ □ Soft Antialias Character width:

Width: Use char width specified in font ▾ 16 px

□ Use filter ▾ Select file

Output

Image | brfnt / brfna

Image file: D:¥dev¥font.bmp ▾ Select file

Letter order: European Language (CodePage 1252 / La ▾

Cell size: ○ Specify size/offset ● Specify margin

Left margin: 0 Right margin: 0

Top margin: 0 Bottom margin: 0

☑ Center glyph

Format: by extension ▾

☑ Draw grid

null width grid margin

Ready Convert

The main dialog box is divided into four sections:

- Input specifications in the top portion
- Filter specifications in the middle
- Output specifications in the bottom portion
- Status display at the very bottom

The **Input** section is used to set resources and options to be used for the conversion source. The **Output** section is used to set post-conversion resources. Once these settings have been specified, click **Convert** to carry out the conversion.

The **Status** section at the bottom of the dialog box shows the current progress of conversion with a progress bar and a status indicator. The other sections of the dialog box are described below.

All FontConverter operations support the dragging and dropping of files into fields where files can be specified.

The FontConverter tool includes a Warning/Error window that is displayed automatically when a warning or an error is generated. The contents of the display are cleared each time a conversion is performed.

The main dialog box closes automatically when the conversion completes with no warnings or errors.

3.2 Conversions

3.2.1 Overview

Follow this procedure to convert fonts using FontConverter.

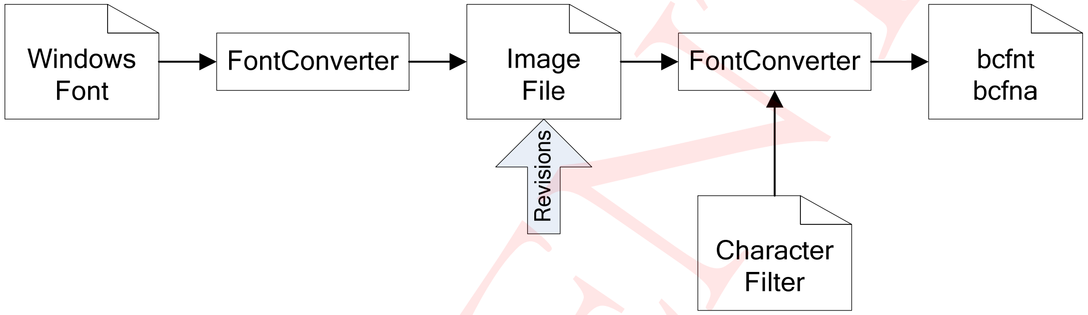
1. **Input**: Set the input resource by selecting the tab for the resource to be converted from the Input specifications section. For details, see section 33.3 Input Specifications.
2. **Output**: Set the output resource by selecting the tab for the resource to be output from the Output specifications section. For details, see section 3.4 Output Specifications.
3. **Filter**: Specify a filter, if needed. If no filter is needed, clear the **Use filter** check box. For details, see section 3.5 Filter Specifications.
4. Click **Convert**.
5. If FontConverter encounters no problems, it completes the conversion and displays no messages. If FontConverter encounters a problem with the input resource, output resource, or the settings made for either, it displays a warning in the **Warning/Error** window. It also attempts to correct the problem and complete the conversion. If FontConverter cannot correct the problem, it displays an error message and the conversion fails.
   In short, if no messages or only warning messages are displayed, the conversion has succeeded and an output file has been created.

3.2.2 Creating a Font

The basic function of FontConverter is to create fonts according to the flow shown in Figure 3-2.

To create a font resource, the font used as the base is converted from a Windows font to an image and then written to an image file. This image file can be revised as necessary. The image file is then converted to a bcfnt or bcfna file. During this second conversion, the data is passed through a filter that ensures that only those characters required by the application are stored in the output file.

Figure 3-2 Basic Flow for Creating a Font Resource



3.3 Input Specifications

3.3.1 Image

Figure 3-3 shows the **Image** tab of the Input Specifications section. A BMP or TGA image conforming to specifications given in *FontConverter Formats* (`FontConverter_Format.pdf`) can be specified as the input.

Figure 3-3 Image Tab in the Input Specification Section

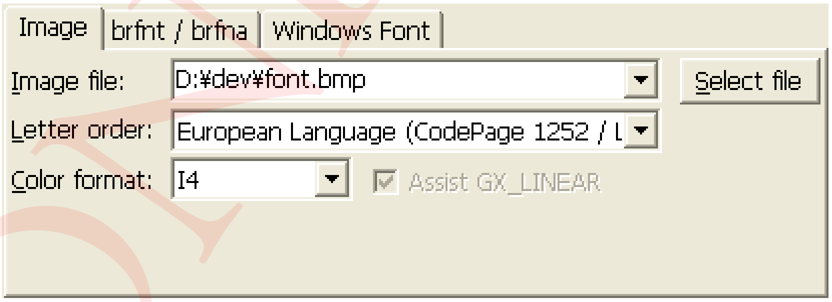


Image file: D:¥dev¥font.bmp

Letter order: European Language (CodePage 1252 / L

Color format: I4

☑ Assist GX_LINEAR

Specify the following information on this tab:

- **Image file**

Specifies the image file to be converted.

- **Letter order**

Selects the letter order in which characters appear in the image file. For details, see *FontConverter Formats* (`FontConverter_Format.pdf`).

- **Color format**

Specifies how the colors for each pixel in the image are to be interpreted.

The color format values and their meanings are as follows.

- A4 or A8
  If the image uses a direct color format, the average value for the three RGB colors is determined for each pixel. If A4 is specified, this average value is then converted into one of 16 gradations; if A8 is specified, it is converted into one of 256 gradations. If the image uses an indexed color format, the index value is used unchanged as the alpha component.

- LA4 or LA8
  For images that use a direct color format, the luminance component takes the average value of the three RGB components at each pixel; this value is then converted into 16 gradient levels when LA4 is specified and 256 levels when LA8 is specified. For images that use an indexed color format, the index value is used unchanged as the luminance component.
  The alpha component takes the alpha value at each pixel for both direct color formats and indexed color formats. This value is used unchanged when LA8 is specified or with its lower four bits truncated when LA4 is specified (to make 16 gradient levels).

- RGB565
  The lower bits are truncated to create 5-bit R, 6-bit G, and 5-bit B components. The alpha component is ignored.

- RGB5A1
  The lower bits are truncated to create 5-bit R, 5-bit G, 5-bit B, and 1-bit A components.

- RGBA4
  The lower bits are truncated so that each RGBA component is 4-bit.

- RGB8
  The input RGB components are used unchanged. The alpha component is ignored.

- RGBA8
  The input RGBA components are used unchanged.

- **Assist GX_LINEAR**

The process of handling the effect of linear interpolation is carried out for glyphs when this option is selected. This process is only effective in the case of a color format (LA4, LA8, RGB5A1, RGBA4, or RGBA8) that includes an alpha channel.

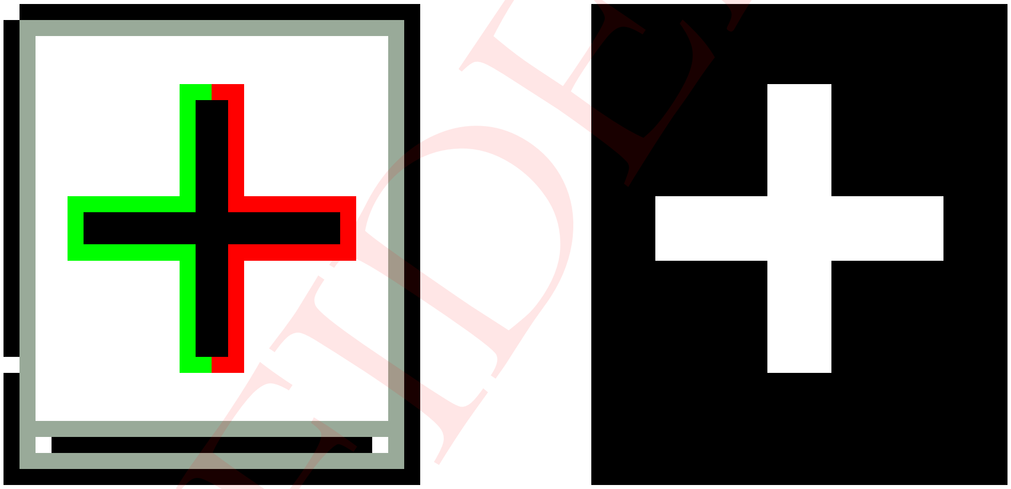
It is possible to use a CTR hardware-based linear interpolation function for textures when displaying glyphs as textures bound to polygons when drawing characters under NintendoWare for CTR.

However, when the linear interpolation function for textures is used, pixels with zero for their alpha value may be referenced, resulting in an unexpected display.

For example, if the RGB font shown in Figure 3-4 is enlarged 10 times and displayed on a black background, it will appear as shown on the left in Figure 3-5 with a white haze surrounding the glyph. This is due to the fact that linear interpolation calculations cause colors such as a muddy translucent red (RGBA=255,127,127,127) to arise in the area where completely opaque red (RGBA=255,0,0,255) and completely transparent white (RGBA=255,255,255,0) are adjacent to each other. Usually, you can expect colors such as the translucent red (RGBA=255, 0, 0, 127) shown on the right in Figure 3-5 to arise.

During the process of handling the effect of linear interpolation, glyphs are corrected so that they are displayed as shown on the right in Figure 3-5. Specifically, the color of pixels that are completely transparent (alpha = 0) is replaced with the average color of the eight surrounding pixels that are not completely transparent (alpha > 0). The alpha value itself is not replaced. Taking Figure 3-4 as an example, the result of processing is as shown in Figure 3-6.

Figure 3-4 Example of an RGBA4Font



Color Channel

Alpha Channel
(White = alpha value 255, Black = alpha value 0)

Figure 3-5 Effect of Linear Interpolation

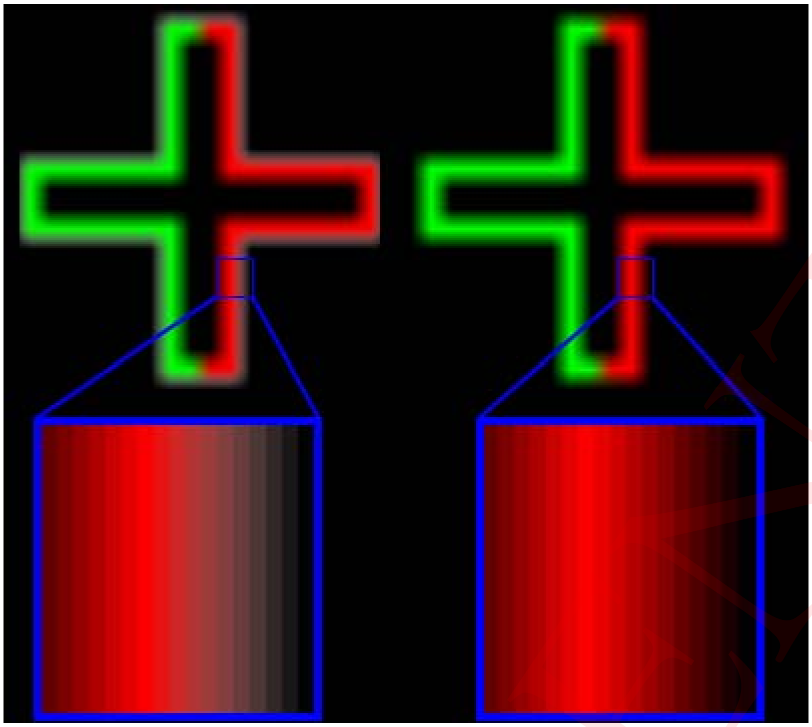
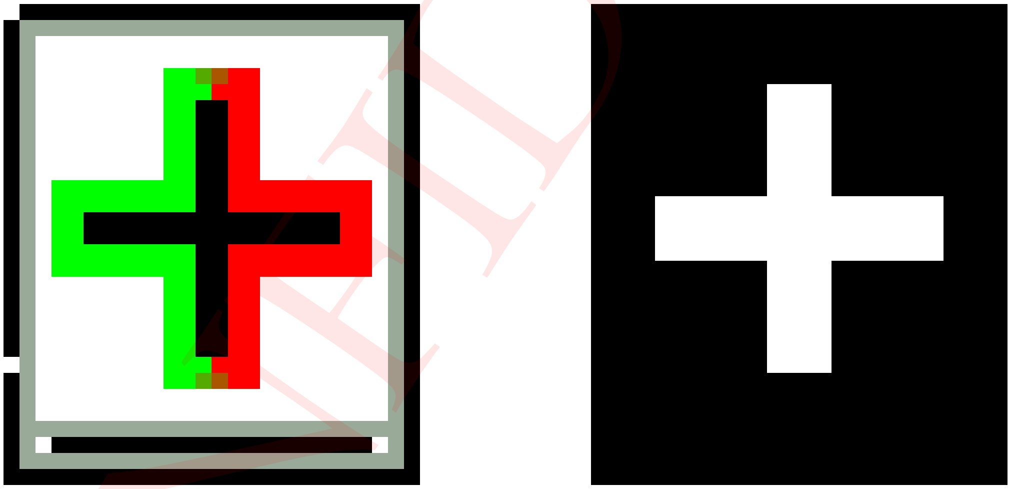


Figure 3-6 Result of Handling the Effect of Linear Interpolation



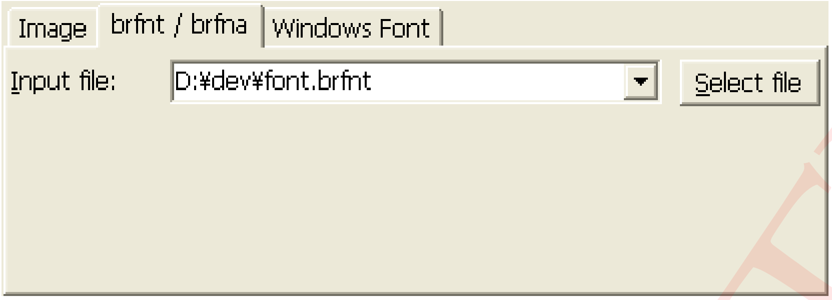
Color Channel

Alpha Channel
(White = alpha value 255, Black = alpha value 0)

3.3.2 bcfnt/bcfna Fonts

Figure 3-7 shows the **bcfnt/bcfna** tab of the **Input** section. A bcfnt or bcfna file created using FontConverter can be specified as input.

Figure 3-7 bcfnt/bcfna Tab in the Input Specification Section



Input file: D:¥dev¥font.brfnt

Specify the following information on this tab:

- **Input file**

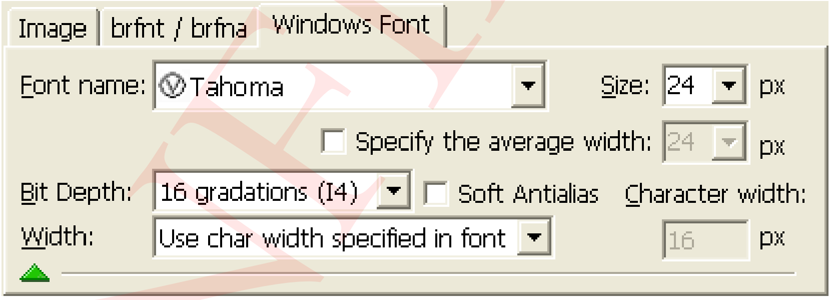
Specifies the bcfnt or bcfna file to be converted.

3.3.3 Windows Fonts

Figure 3-8 shows the **Windows Font** tab of the **Input** section. Use this tab to specify as input a font installed on Microsoft Windows.

When a Windows font is specified as input, all the characters of that font that lie in the Unicode range 0x0000 through 0xFFFD are passed to the output. If **Image** is specified as the output, the output characters are limited to those in the text order file. If **bcfnt** is specified as the output, many of the characters in that font are output unless a text filter has been specified. Caution is required because some fonts can output tens of thousands of characters.

Figure 3-8 Windows Font Tab in the Input Specification Section



Font name: Tahoma

Size: 24 px

□ Specify the average width: 24 px

Bit Depth: 16 gradations (I4)

□ Soft Antialias

Character width: 16 px

Width: Use char width specified in font

Specify the following information on this tab:

- **Font name**

Specifies the font to be converted. Fonts installed on Windows are displayed in a list. Fonts shown with the same background color that is used for tool tips are raster (bitmap) fonts, whereas those shown with the usual background are vector (outline) fonts.

- **Size**

Specifies the size in pixels of the font to be converted. Treat this value as a nominal one since the size of the font that is actually output depends on the font.

- **Specify Average Width**

Enables the specification of average width of the conversion-source font such that the height:width ratio can be changed by specifying the average width.

- **Average Width**

Specifies the average width of the conversion-source font in pixels. Because the font size that is actually output depends on the font itself, please consider this value as a reference.

- **Bit Depth** (number of gradations)

Specifies the number of gradations to be output. The information given in parentheses represents the texture format used when outputting in bcfnt format. Only two gradations can be selected for raster fonts.

- **Soft Antialias** (soft anti-aliasing)

Although glyphs output when using multi-level output (output other than two-level output) are subjected to anti-aliasing, this anti-aliasing is made softer. The quality of glyphs being output may improve depending on the font.

- **Width** (character width)

Specifies how the character width of the font to be output should be calculated using one of the following three methods:

- Use the glyph width as the character width

  The character width is made equal to the glyph width. Consequently, no spaces are inserted before or after the glyph. This results in a proportional font.
  For space characters, a character width of 0 is output as there is no associated glyph; its use would therefore make it appear as if there was no space character. When using space characters, you must either output with the **Use Glyph Width (Protect Space Characters)** setting or adjust the character width of the space character after creating an image file as output.

- Use Glyph Width (Protect Space Characters)

  Although basically the same as **Set Character Width to Glyph Width**, space characters have the same character width as when **Use the character width specified by the font** is specified.

- Use the character width specified by the font

  The character width of each font is used as is. Consequently, proportional fonts result in proportional fonts and fixed-width fonts result in fixed-width fonts.

- Use the same width for all characters

  All characters are made to have the same width by using just the right spacing before and after glyphs. The width shared by all characters is specified in the **Fixed Width** field. Output is fixed width.

- **Fixed width**

Specify a width shared by all characters in pixels. A value can be entered into this field only if **Use the same width for all characters** has been selected for the character width.

3.4 Output Specifications

3.4.1 Image

Figure 3-9 shows the **Image** tab. The font data that is used as input to the conversion process is output as an image file as described in *FontConverter Formats* (`FontConverter_Format.pdf`). Image files output here can subsequently be used as image input in the Input Specifications section.

Figure 3-9 Image Tab in the Output Specification Section

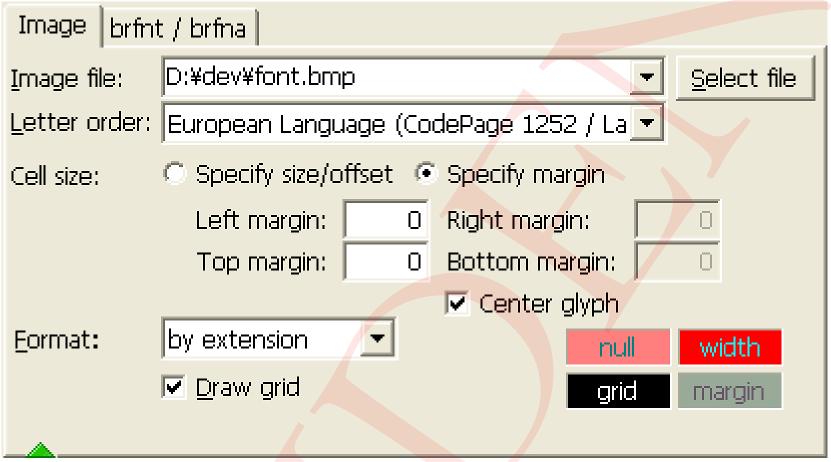


Image file: D:¥dev¥font.bmp

Letter order: European Language (CodePage 1252 / La

Cell size: ○ Specify size/offset ● Specify margin

Left margin: 0 Right margin: 0

Top margin: 0 Bottom margin: 0

☑ Center glyph

Format: by extension

☑ Draw grid

Specify the following information on this tab:

- **Image file**

Specifies the path for the image file to be output.

When a setting other than **by Extension** is selected for **Format**, an extension is added at conversion time if the image file extension differs from that selected in **Format**. However, if the image file extension is one of those supported by FontConverter (.bmp or .tga), the extension will replace (rather than be added to) the specified extension.

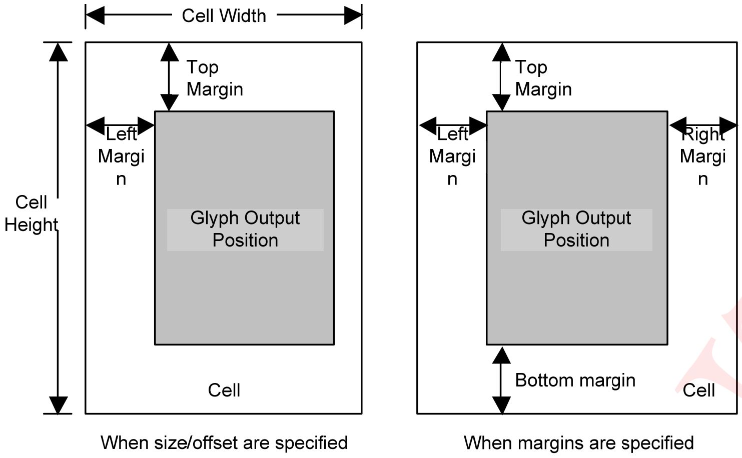
- **Letter order**

Selects the order in which characters are to be output to the image file. For details, see *FontConverter Formats* (`FontConverter_Format.pdf`).

- **Cell size**

If **Specify size/offset** is selected, cell size and glyph output position are specified. These parameters are shown on the left in Figure 3-10. If **Specify margin** is selected, cell size and glyph output position are specified. These parameters are shown on the right in Figure 3-10.

Figure 3-10 Specifying Cell Size and Glyph Output Position



Selecting **Center Glyph** centers the glyph output position in the cell. Specifically, this is handled as follows for each respective setting.

- When Size/Offset is selected:

Top margin and **Left margin** cannot be input. The glyph is automatically centered based on the cell width and height.

- When Margin is selected:

Right margin and **Bottom margin** cannot be input. The glyph is centered by setting the right margin equal to the left margin and the bottom margin equal to the top margin.

- **Format**

Specifies the format of the image file to be output. If **by extension** is selected, the output format is determined based on the image file extension specified in Image File.

- **Draw grid**

A grid is displayed if the **Draw Grid** check box is selected.

- **null / width / grid / margin**

Clicking any rectangle opens a color selection dialog box, allowing you to specify the color of each area in the image file to be output.

- null
  Specifies the color used to fill cells for which <null/> is specified in the text ordering file.
- width
  Specifies the color of the width line.
- grid
  Specifies the color of the grid to be drawn when the **Draw grid** check box has been selected.

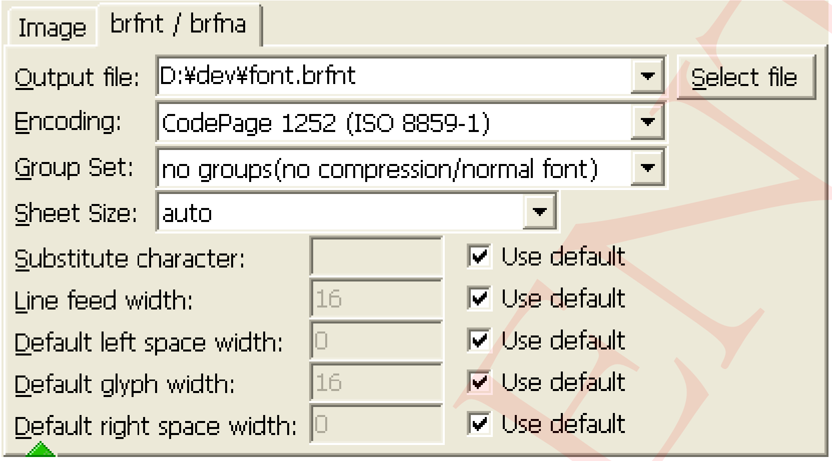
- margin

  You can specify the color of the one-pixel region between the cell and the grid.

3.4.2 bcfnt/bcfna

Figure 3-11 shows the **bcfnt/bcfna** tab used for output specifications. Input font data is converted to bcfnt or bcfna format for use by NintendoWare for CTR.

Figure 3-11 bcfnt/bcfna Tab in the Output Specification Section



Output file: D:¥dev¥font.brfnt

Encoding: CodePage 1252 (ISO 8859-1)

Group Set: no groups(no compression/normal font)

Sheet Size: auto

Substitute character: ☑ Use default

Line feed width: 16 ☑ Use default

Default left space width: 0 ☑ Use default

Default glyph width: 16 ☑ Use default

Default right space width: 0 ☑ Use default

Specify the following information on this tab:

- **Output file**

Specifies the file name to be output.

A file extension is added if the extension is not an extension supported by the selection made under **Group Set**. Note, however, that the extension is replaced (rather than added) when the extension of the file is `.bcfnt` or `.bcfna`.

- **Encoding**

Specifies the character encoding format supported by the font to be output.

- **Group Set**

Selects the glyph group file to be used. If **no groups(no compression/normal font)** is selected, a normal font resource (bcfnt) is created without using a glyph group file. If a glyph group file is used, a compressed archive font (bcfna) is created with an embedded group set defined by the glyph group file.

If a glyph that is undefined in any glyph group file exists in input, that glyph is assigned to a special glyph group with the name “*”.

- **Sheet Size**

Note: You cannot currently specify the sheet size.

Font glyphs are stored in the bcfnt in texture format and are called “sheets.” This setting specifies the size of the sheet. Specified as an area where height and width are both given by a power of two, the sheet size allowing the smallest font resource file size is used.

For example, if **64K pixels (256x256 equivalent)** is selected, height times width is 65536 (=64K) pixels. Out of the five sheet sizes possible for this area (that is, 64x1024, 128x512, 256x256, 512x128, and 1024x64), the sheet size that allows the smallest font resource file size is used.

If **auto** is selected, the sheet size that results in the smallest overall font resource size is used. The file size of the bcfna format data output will not necessarily be minimal when using archive fonts due to the fact that this determination is made before data is compressed.

- **Substitute character**

Specifies the substitute character. This character is displayed when an attempt is made to render a character not found in the font resource under NintendoWare for CTR. The substitute character can be specified either through direct character input or by specifying a character code.

If only one character is entered, direct character entry is assumed. If more than one character is entered, a character code specification is assumed. Octal, decimal, and hexadecimal values as defined in the C language specification can be used to specify a character code.

When the **Use default** check box is selected, the character that has the smallest character code in the font resource is used.

- **Leading**

Specifies the height of one line. This value is used as the line height by NintendoWare for CTR. Valid values range from 0 to 255.

When the **Use default** check box is selected, the height of the font is used as the value.

- **Default left space width**

Specifies the default left space width. This value is used for characters that do not have a fixed left space width of their own. Valid values range from -128 to 127.

When the **Use default** check box is selected, a 0 is used as the value.

- **Default glyph width**

Specifies the default glyph width. This value is used for characters that do not have a fixed glyph width of their own.

When the **Use default** check box is selected, the maximum glyph width of the font is used as the value.

- **Default right space width**

Specifies the default right space width. This value is used for characters that do not have a fixed right space width of their own.

When the **Use default** check box is selected, a 0 is used as the value.

- **Use default**

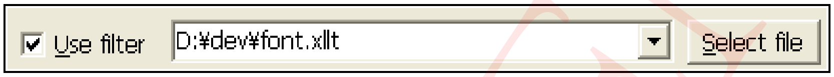
When the **Use default** check box is selected to the right of each input field, a default value is used for that parameter. For details on the values actually used, see the description given for each item. However, if the input is in the bcfnt format, the corresponding setting values inside the bcfnt input data are used as default values.

3.5 Filter Specifications

Figure 3-12 shows the filter specification field in the dialog box. If a text filter file is specified, characters not defined in the text filter file will not be output. This makes it possible to create a compact bcfnt file containing only those characters required. If a text filter file is not used, all input characters are output.

For information on text filter files, see *FontConverter Formats* (`FontConverter_Format.pdf`).

Figure 3-12 Filter Specification



☑ Use filter

D:¥dev¥font.xllt

- **Use filter**

When selected, the specified filter is applied during conversion.

4 Precautions

4.1 Precautions when Converting Windows Fonts

Windows XP has automatic font linking. This feature is used to automatically display characters using another font when a character not included in the specified font must be displayed. For example, the Tahoma font is installed as standard in Windows XP but does not include Japanese characters. Even so, automatic font linking allows you to display Japanese characters in Notepad or other applications even when a font such as Tahoma is set as the default font. Although this feature is useful for normal PC applications, it can result in accidentally incorporating an unlicensed font into a commercial game software product.

Automatic font linking is disabled and only characters included in the specified font are output when using FontConverter. As a result, Japanese characters will not be output when using a font (like Tahoma) that does not include them.

4.2 Unicode and FontConverter

The FontConverter tool uses Unicode for the internal processing of character codes. It cannot handle characters not included in the Unicode code set. Conversion must be performed based on Windows conversion rules because Windows features are used to convert the various encodings. For example, although the JIS character code 0x8160 for "~" is defined in Unicode as U+301C, it is defined as U+FF5E under Windows. Consequently, FontConverter also handles it as U+FF5E.

Table 4-1 shows the character code conversions that FontConverter generates for the various inputs and outputs.

Table 4-1 Places Where Character Codes Are Converted

Location	Details of Conversion
[Input] BMP	References the text order file and assigns from cell position to Unicode.
[Input] bcfnt	Converts from the encoding stored in the bcfnt file to Unicode.
[Input] Windows	Outputs characters in Unicode from U+0000 to U+FFFD.
[Output] BMP	References the text order file and assigns from Unicode to the cell position.
[Output] bcfnt	Converts from Unicode to the specified encoding.
Text filter file	Converts to Unicode when loaded.
Letter order file	Converts to Unicode when loaded.

5 Revision History

Version	Revision Date	Category	Description
1.0.2	2010/12/02	Changed	- Overall Updated document properties.
1.0.1	2010/09/09	Changed	- 3.3 Input Specifications Deleted text about the green triangle icon.
1.0.0	210/07/02	Changed	- Overall Changed format.
	2010/01/15	Changed	- 3.3 Input Specifications Changed description about texture format - Overall Revised content of figures
	2009/10/30	-	- Initial version