

Revolution NintendoWare

NW4R fontcvtr

Revision Date: 2008/08/22

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

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and are protected by national and international copyright laws. 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.

Table of Contents

1	Introduction	6
1.1	About NW4R fontcvtr	6
1.2	Configuration of the Manual.....	6
1.3	What fontcvtr Can Do.....	6
1.4	Font Licenses.....	7
2	Update History	8
3	File Structure	11
4	Operations	13
4.1	Graphical User Interface	13
4.2	Conversions	14
4.2.1	Overview.....	14
4.2.2	Creating a Font.....	15
4.3	Input Specifications	15
4.3.1	Image.....	15
4.3.2	brfnt/brfna Fonts	18
4.3.3	Windows Fonts.....	18
4.4	Output Specifications	20
4.4.1	Image.....	20
4.4.2	brfnt/brfna	23
4.5	Filter Specifications	25
5	Precautions	26
5.1	Precautions when Converting Windows Fonts	26
5.2	Character Encoding Using Unicode.....	26
6	About Xerces-C++	27

Figures

Figure 1-1	Relationships of Elements Used to Create Fonts	7
Figure 3-1	fontcvtr File Structure	11
Figure 4-1	Main Dialog Box for fontcvtr	13
Figure 4-2	Basic Flow for Creating a Font Resource	15
Figure 4-3	Image Tab in the Input Specification Section	15
Figure 4-4	Example of an RGB5A3 Font.....	17
Figure 4-5	Effect of Linear Interpolation	17
Figure 4-6	Result of Handling the Effect of Linear Interpolation.....	18
Figure 4-7	brfnt/brfna Tab in the Input Specification Section	18
Figure 4-8	Windows Font Tab in the Input Specification Section.....	19

Figure 4-9 Image Tab in the Output Specification Section 21
Figure 4-10 Specifying Cell Size and Glyph Output Position 22
Figure 4-11 brfnt/brfna Tab in the Output Specification Section 23
Figure 4-12 Filter Specification 25

Tables

Table 5-1 Places Where Character Codes Are Converted..... 26

Revision History

Revision Date	Description
2008/08/22	Support for Version 1.5.0.
2008/04/03	Changed the Xerces-C++ license statement.
2007/10/23	Support for Version 1.4.5.
2007/07/17	<ul style="list-style-type: none">• Corrected typos.• Support for Version 1.4.4.
2007/01/26	Support for Version 1.4.3.
2006/11/30	Corrected typos.
2006/10/30	Support for Version 1.4.2.
2006/10/19	Support for Version 1.4.1.
2006/09/20	Support for Version 1.4.1.
2006/09/07	<ul style="list-style-type: none">• Support for Version 1.4.0.• Added "Update History."• Made an independent manual, "fontcvtr I/O File Specifications" out of this document.
2006/08/17	<ul style="list-style-type: none">• Support for Version 1.3.3.• Changed the glyph image extraction conditions in section 6.1.2.1 Glyph Images.
2006/07/04	Replaced "Revolution" with "Wii." Added notes.
2006/06/30	Support for Version 1.3.2
2006/05/17	Support for Version 1.3.0.
2006/04/04	Support for Version 1.1.0.
2006/03/06	NOA release.
2005/12/26	Initial version.

1 Introduction

1.1 About NW4R fontcvtr

NW4R fontcvtr (hereafter referred to as “fontcvtr”) is a Microsoft Windows application for creating font resources (.brfnt files) and archive fonts (.brfna files). These files are used as fonts by NintendoWare for Revolution when drawing text.

1.2 Configuration of the Manual

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

- fontcvtr_Manual.pdf – NW4R fontcvtr (manual for the GUI-version of fontcvtr)
This is the manual you are now reading. This manual describes the GUI (Windows) version of fontcvtr, including features specific to the GUI version and features shared with the CUI (command line) version.
- fontcvtrc_Manual.pdf – NW4R fontcvtrc (manual for the command line version of fontcvtr)
This manual is for nw4r_fontcvtrc.exe, the command line version of fontcvtr. It describes the features specific to the command line version. The reader is assumed to have also read fontcvtr_Manual.pdf.
- fontcvtr_Format.pdf – fontcvtr format (fontcvtr I/O file specifications)
This manual describes the various file formats that are input to and output from fontcvtr. Specifically covered are image files (.bmp and .tga), text filter files (.x11t), text order files (.x1or), and glyph group files (.xggp).

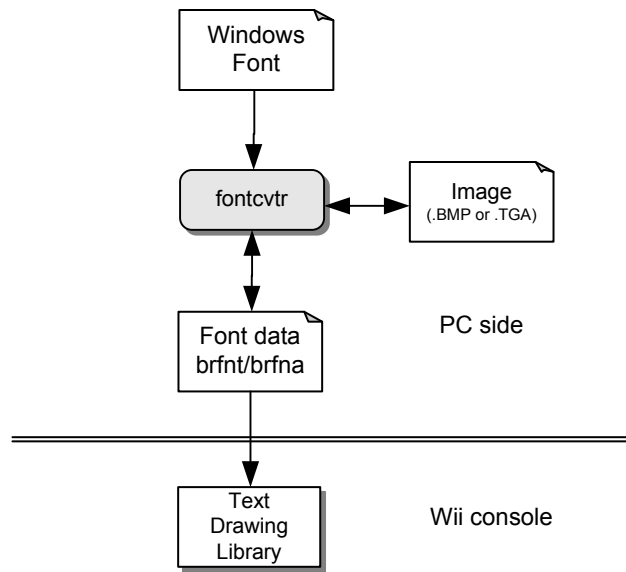
1.3 What fontcvtr Can Do

The fontcvtr tool creates font resources from Windows fonts. Although this tool can be used to create brfnt/brfna 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 brfnt/brfna files, Windows fonts can first be written to an image format, adjusted, and then converted to brfnt/brfna. 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 Relationships of Elements Used to Create Fonts



1.4 Font Licenses

Although fontcvtr can convert any font installed on a PC for use as a display font on the Wii console, no font licenses are included with fontcvtr or NintendoWare for Revolution.

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

2 Update History

2008/08/22 Release (Version 1.5.0)	
New	<ul style="list-style-type: none"> Added a feature to specify the average width for input Win, and allowed the Height:Width size ratio to be changed.
2007/10/23 Release (Version 1.4.5)	
New	<ul style="list-style-type: none"> In the input Win character width specification, added a requirement to use the character width specified by the font for space characters and to use the glyph width for all other cases.
2007/07/17 Release Patch 1 (no update)	
2007/07/17 Release (Version 1.4.4)	
Changed	<ul style="list-style-type: none"> Changed so a warning is output for fonts with linefeed widths exceeding 127. Changed so an error is output as required when glyphs do not exist.
2007/03/16 Release (no update)	
2007/01/26 Release (Version 1.4.3)	
Changed	<ul style="list-style-type: none"> The default status for software anti-aliasing in the input Win has been changed to ON. (CUI version) Command line help has been updated.
2006/10/30 Release Patch 1 (no update)	
2006/10/30 Release (Version 1.4.2)	
New	<ul style="list-style-type: none"> Made it possible to specify the color for each of the locations in the output image when using Image output.
Corrected	<ul style="list-style-type: none"> Corrected a problem where the glyph width would be calculated too large if a color close to white is located at the edge of the glyph when converting to I4, IA4, RGB565, or RGB5A3 format when using Image input. Corrected a problem where the color format was being ignored when converting from the input Image to output Image.
2006/09/20 Release (Version 1.4.1)	
Corrected	<ul style="list-style-type: none"> Corrected a problem in the Image input where the alpha channel is referenced for "color format" even when alpha is unnecessary. Corrected a problem in the brfmt/brfna output where character is not correctly shared when the same character is included in multiple groups.
2006/09/07 Release (Version 1.4.0)	
New	<ul style="list-style-type: none"> Added feature to handle the effect of linear interpolation when using Image input. Added a feature for soft anti-aliasing when using Win input. Added archive font output when using brfmt output. Made it so that sheet size can be specified when using brfmt output.

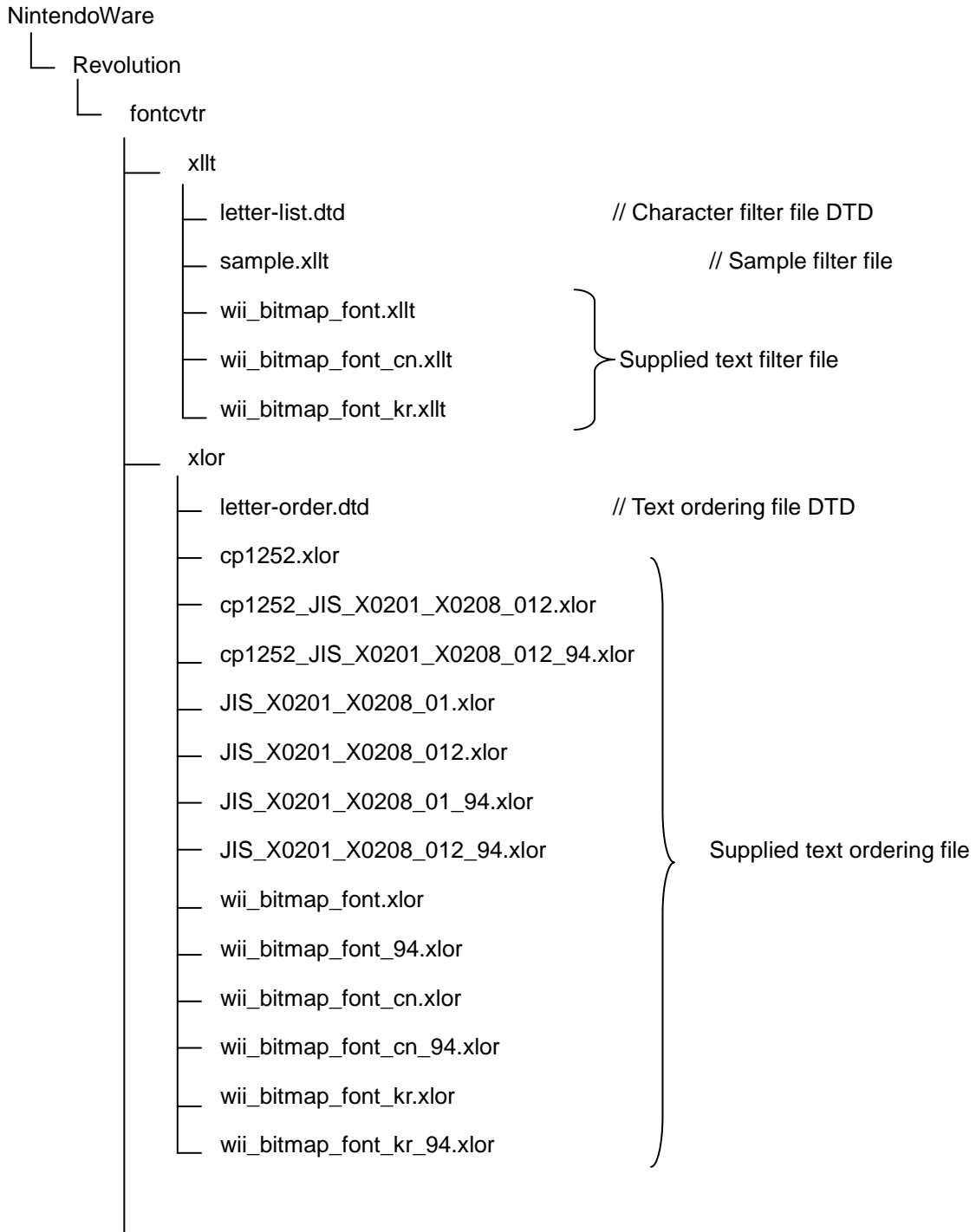
Changed	<ul style="list-style-type: none"> • Made it so that output in I4 or I8 texture format is possible when converting from Win input to brfmt output. • Improved the font size specification used with Win input so that it is closer to the number of output pixels. • Changed the specifications of letter order files and updated to Version 1.1. • Made supplied text order files compatible with Version 1.1. • Made it so that a descent of zero can be specified for image files. • (GUI Version) Increased speed of startup.
Corrected	<ul style="list-style-type: none"> • Changed functionality so that the maximum character width, rather than the maximum glyph width previously used, is used as the font width when using Win input and brfmt output. • Corrected a problem where an error would not result when a nonexistent font was specified. • Corrected a problem where an ascent of zero could not be specified for an image file. • Corrected a problem where the left space width and right space width were reversed when using a character width of zero. • Standardized the left space width and right space width when using a glyph width of zero in cases where they used to differ depending on the input. • Other detailed changes.
2006/07/04 Release patch2 (Version 1.3.3)	
Corrected	<ul style="list-style-type: none"> • Changed the conditions for detecting a glyph image in an image that includes an alpha channel when using Image input.
2006/07/04 Release patch1 (No update)	
2006/07/04 Release (Version 1.3.2)	
Changed	<ul style="list-style-type: none"> • Adjusted quality when using 16-level output with Win input. • (GUI Version) Made it so the extension filter on the file selection dialog box changes depending on the image format when using Image output. • (GUI Version) Made it so the extension of the output file name is adjusted depending on the image format when using Image output. • (GUI Version) Made it so that confirmation is asked for when overwriting a file. • (GUI Version) Established independent messages when outputting files.
Corrected	<ul style="list-style-type: none"> • Corrected a problem where white space characters were not being output normally depending on the font when using Win input.
2006/05/17 Release patch3 (No update)	
2006/05/17 Release patch2 (No update)	
2006/05/17 Release patch1 (Version 1.3.1)	
Corrected	<ul style="list-style-type: none"> • Corrected a problem where the ascent and descent were shifted in the input image. • Corrected a problem where the baseline position was shifted by one pixel when an image was used as input and a brfmt file as output. • Corrected a problem where fontcvtrc.exe would sometimes terminate with an error when using a brfmt file as output.
2006/05/17 Release (Version 1.3.0)	
New	<ul style="list-style-type: none"> • Added the concepts of descent, ascent, and font width to the image format and brfmt files.
Corrected	<ul style="list-style-type: none"> • (Command Line Version) Corrected a problem where the operations performed by using the options <code>-oe utf8</code> and <code>-oe utf16</code> were reversed.

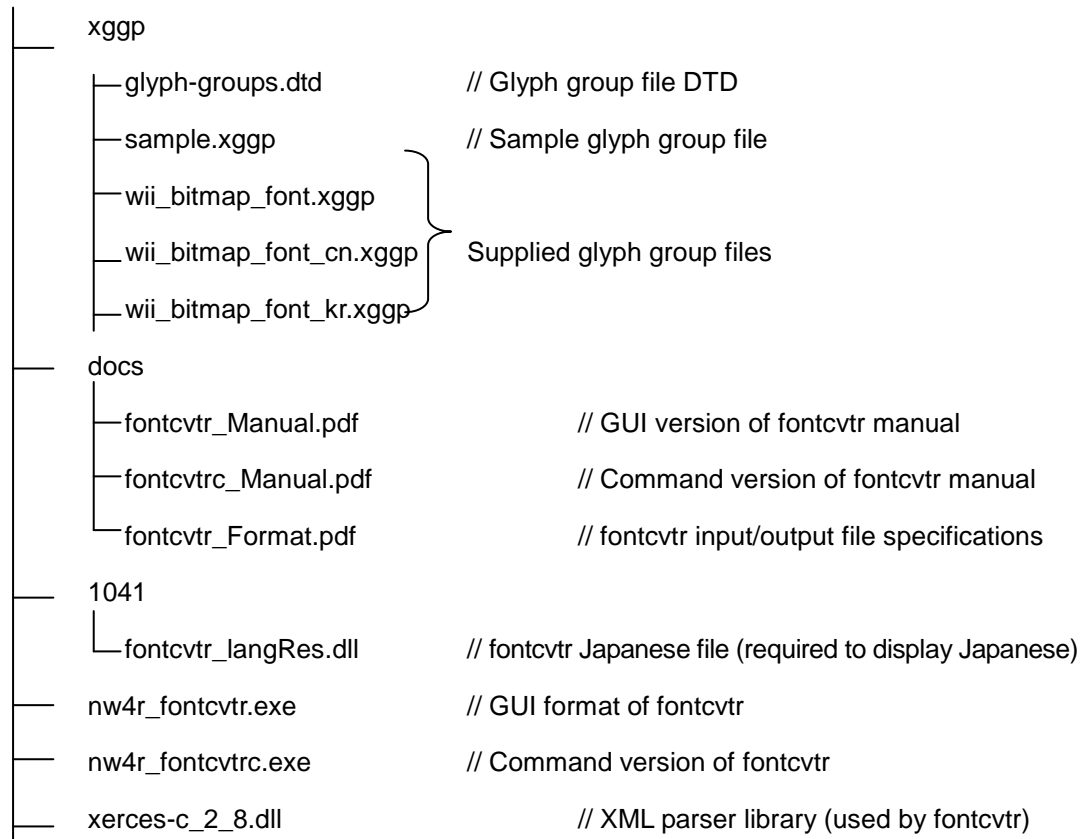
2006/04/10 Release patch2 (No update)	
2006/04/10 Release patch1 (Version 1.2.0)	
Changed	<ul style="list-style-type: none"> Made it so the alpha channel of image files is input/output for glyph images only.
Corrected	<ul style="list-style-type: none"> Corrected a problem where the baseline position was being read as zero for image files without a grid drawn when using Image input. Corrected a problem where the glyph width could not be read normally when using I4 or I8 format brfnt files as input. Corrected a problem where an exception would occur when using brfnt input and brfnt output. Corrected a problem when using brfnt output where the cell width in brfnt files being output was one higher than the actual value.
2006/04/10 Release (Version 1.1.0)	
New	<ul style="list-style-type: none"> Made it so TGA format files can be output.
Changed	<ul style="list-style-type: none"> Fixed the dpi of output BMP files (that used to be retrieved from the desktop) to 72 dpi.
2006/03/06 Release (No update)	
2006/01/23 Release (Version 1.0.0)	
Initial version.	

3 File Structure

The fontcvtr files are located in the `NintendoWare/Revolution/Fontcvtr` directory. Figure 3-1 shows the structure of the files and folders related to fontcvtr.

Figure 3-1 fontcvtr File Structure





For information on the text filter file, text order file, or glyph group file shown in the file configuration diagram, see the manual *fontcvtr Formats* (`fontcvtr_Format.pdf`).

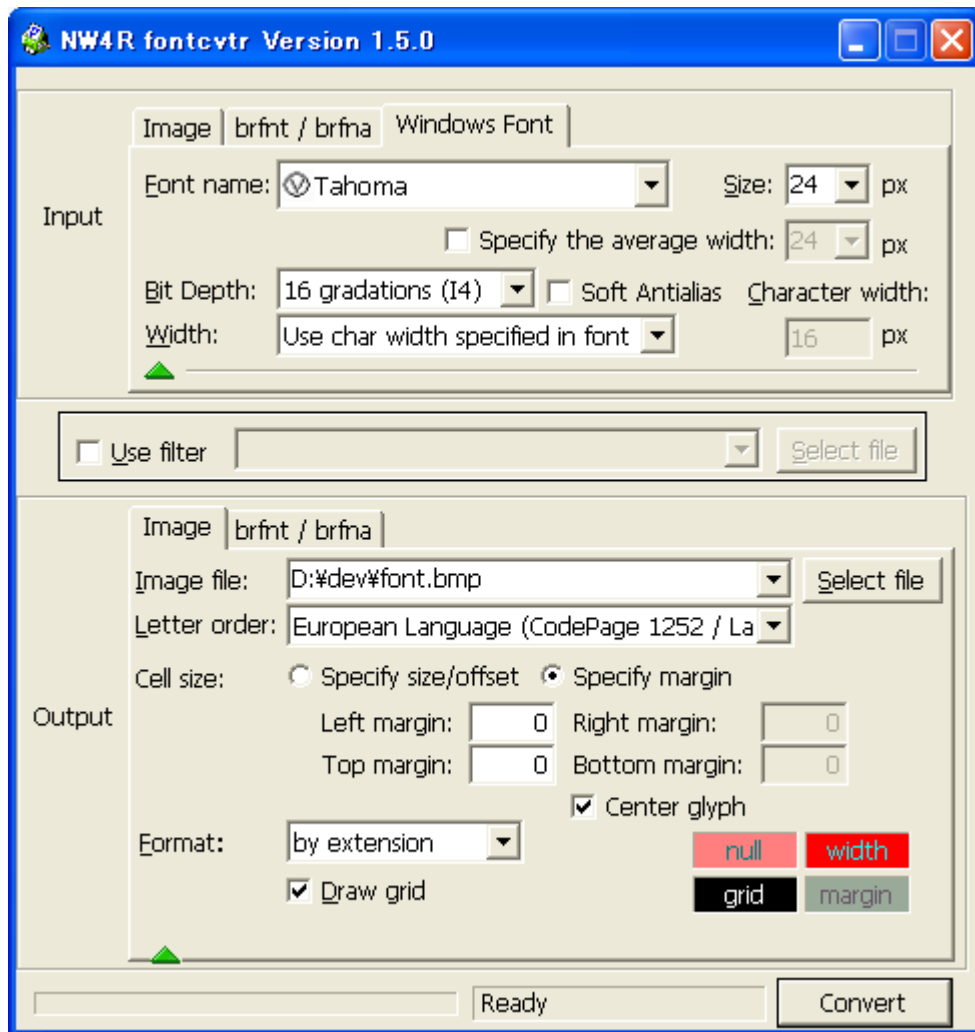
Although not shown in Figure 3-1, a setup file named `nw4r_fontcvtr.ini` is created in the directory in which `nw4r_fontcvtr.exe` resides when the GUI version of `fontcvtr` is started. The display state of the GUI version of `fontcvtr` is automatically recorded in this setup file. The next time `fontcvtr` is started, this file is used to restore the state in effect at the time `fontcvtr` was most recently exited.

4 Operations

4.1 Graphical User Interface

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

Figure 4-1 Main Dialog Box for fontcvtr



The main (**NW4R fontcvtr**) 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 fontcvtr operations support the dragging and dropping of files into fields where files can be specified.

The fontcvtr 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.

4.2 Conversions

4.2.1 Overview

Follow this procedure to convert fonts using fontcvtr:

Set the input resource by selecting the tab for the resource to be converted from the Input specifications section. For details, see section 0

1. **Input:** Set the output resource by selecting the tab for the resource to be output from the Output specifications section. For details, see section 4.3 Input Specifications.
2. **Output:** Specify a filter, if needed. If no filter is needed, clear the **Use filter** check box. For details, see section 4.4 Output Specifications.
3. **Filter:** Click **Convert**.

If fontcvtr encounters no problems, it completes the conversion and displays no messages. If fontcvtr 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 fontcvtr cannot correct the problem, it displays an error message and the conversion fails.

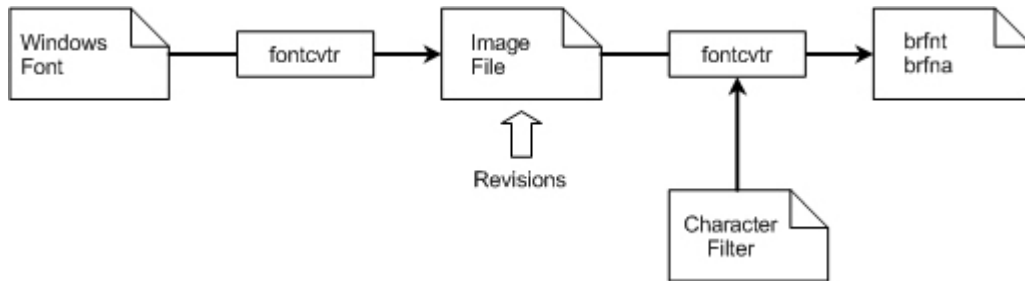
In short, if no messages or only warning messages are displayed, the conversion has succeeded and an output file has been created.

4.2.2 Creating a Font

The basic function of fontcvtr is to create fonts according to the flow shown in Figure 4-2.

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 brfmt 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 4-2 Basic Flow for Creating a Font Resource

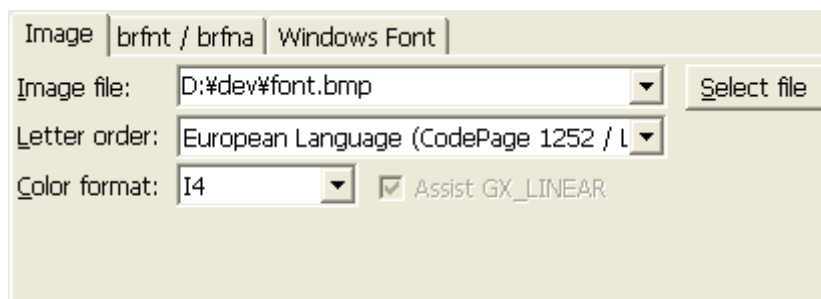


4.3 Input Specifications

4.3.1 Image

Figure 4-3 shows the **Image** tab of the Input Specifications section. A BMP or TGA image conforming to specifications given in *fontcvtr Formats* (`fontcvtr_Format.pdf`) can be specified as the input.

Figure 4-3 Image Tab in the Input Specification Section



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 *fontcvtr Formats* (`fontcvtr_Format.pdf`).

- **Color format**

Specifies how the colors for each pixel in the image are to be interpreted. Possible values and their meaning are:

- **I4 or I8**

If the image uses a direct color format, the average value for the three RGB colors is determined for each pixel. If I4 is specified, this average value is then converted into one of 16 gradations; if I8 is specified, it is converted into one of 256 gradations. If the image uses an index color format, the index value is used without change as the intensity. The alpha component is ignored.
- **IA4 or IA8**

The intensity component is calculated just as when I4 or I8 is specified. If IA4 is specified, the alpha value for each pixel is converted into one of 16 gradations by dropping the lower 4 bits regardless of whether a direct color format or an index color format is used. If IA8 is specified, the alpha value is used as is.
- **RGB565**

Each RGB component is converted into a 5-bit value for red, a 6-bit value for green, and a 5-bit value for blue. Each color component is converted by dropping the appropriate number of lower bits. The alpha component is ignored.
- **RGB5A3**

If alpha is 255, each RGB component is converted into a 5-bit value. If alpha is less than 255, each RGBA component is converted by dropping the lower bits into a 4-bit value for red, a 4-bit value for green, a 4-bit value for blue, and a 3-bit value for alpha. Each component is converted by dropping the appropriate number of lower bits.
- **RGBA8**

Input RGBA data is used as is.

- **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 (IA, IA8, RGB5A3, or RGBA8) that includes an alpha channel.

It is possible to use a Wii hardware-based linear interpolation function for textures when displaying glyphs as textures bound to polygons when drawing characters under NintendoWare for Revolution. 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 4-4 is enlarged 10 times and displayed on a black background, it will appear as shown on the left in Figure 4-5 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 4-5 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 4-5. 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 4-4 as an example, the result of processing is as shown in Figure 4-6.

Figure 4-4 Example of an RGB5A3 Font

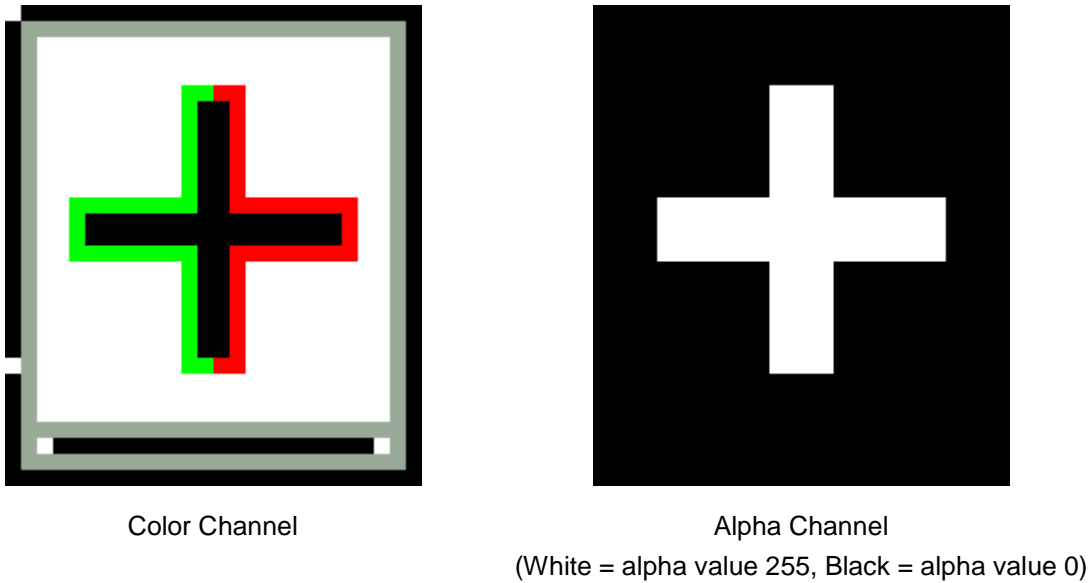


Figure 4-5 Effect of Linear Interpolation

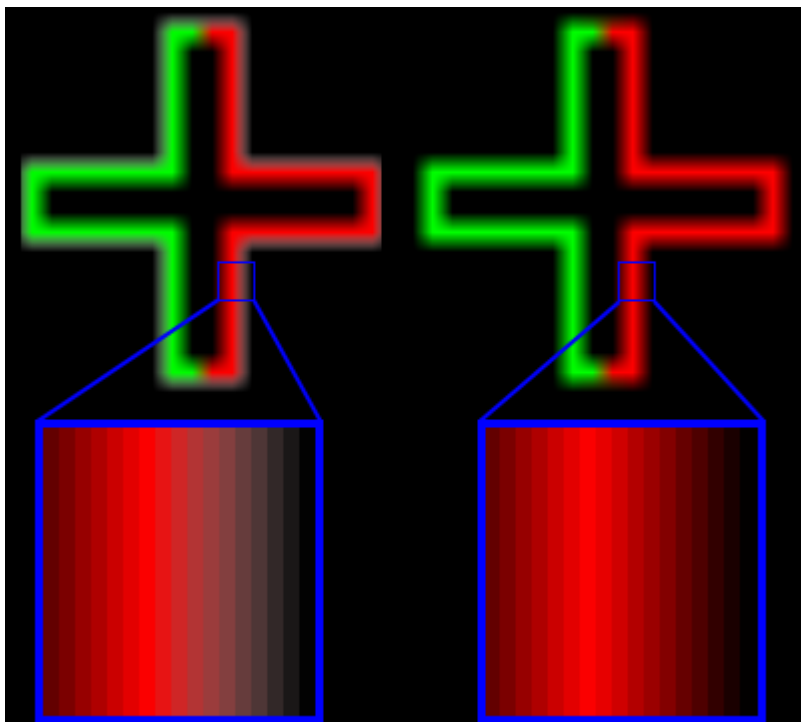
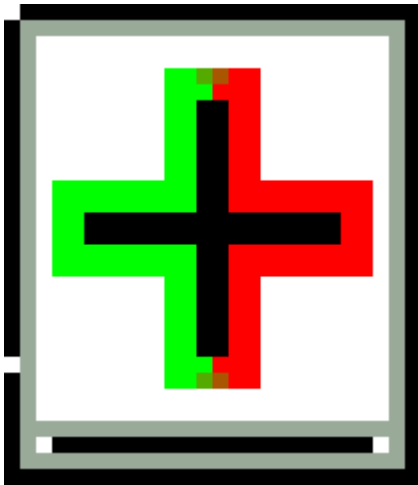
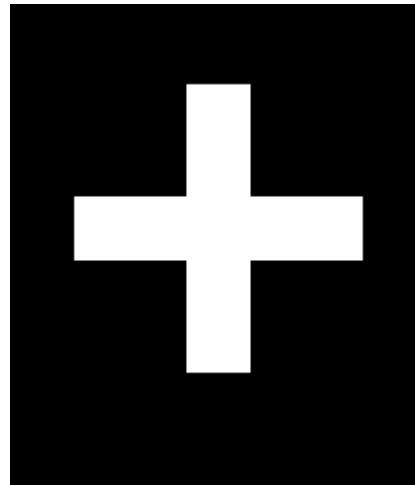


Figure 4-6 Result of Handling the Effect of Linear Interpolation



Color Channel



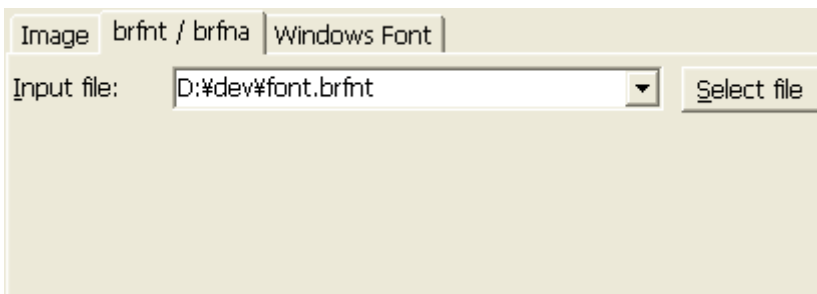
Alpha Channel

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

4.3.2 brfnt/brfna Fonts

Figure 4-7 shows the **brfnt/brfna** tab of the **Input** section. A brfnt or brfna file created using fontcvtr can be specified as input.

Figure 4-7 brfnt/brfna Tab in the Input Specification Section



Specify the following information on this tab:

- **Input file**
Specifies the brfnt or brfna file to be converted.

4.3.3 Windows Fonts

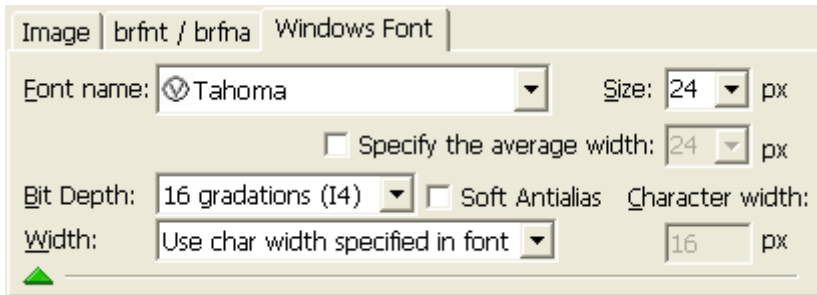
Figure 4-8 shows the **Windows Font** tab of the **Input** section. Use this tab to specify as input a font installed on Microsoft Windows.

When a Windows font is specified as input, all the characters of that font that lie in the Unicode range 0x0000 through 0xFFFFD 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 **brfnt** 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.

By default, only the **Font name** and **Size** items are displayed on the **Windows Font** tab. Clicking the green triangle icon toggles the display to show other items.

Figure 4-8 Windows Font Tab in the Input Specification Section



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 brfnt 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:

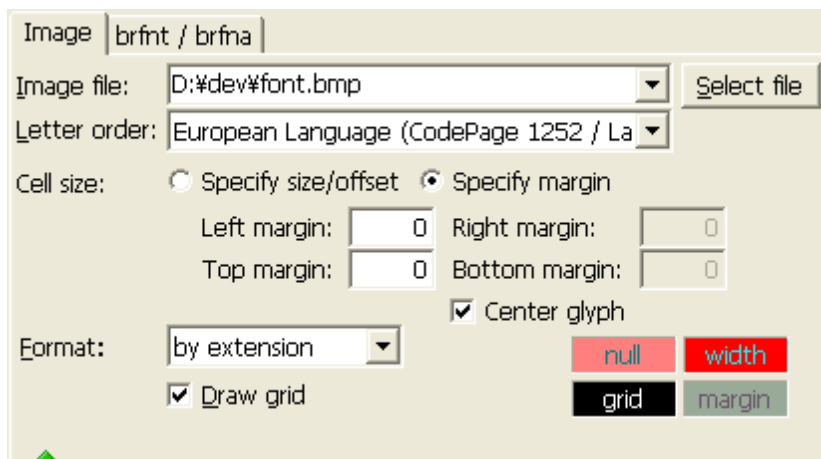
- Use the glyph width as the character width
The character width is made equal to the glyph width. Consequently, no spaces are inserted before or after the glyph. This results in a proportional font.
For space characters, a character width of 0 is output as there is no associated glyph; its use would 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.

4.4 Output Specifications

By default, only those items required for conversion are displayed on the tabs in the Output Specifications section. Clicking the green triangle icon toggles the display of additional items.

4.4.1 Image

Figure 4-9 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 *fontcvtr Formats* (*fontcvtr_Format.pdf*). Image files output here can subsequently be used as image input in the Input Specifications section.

Figure 4-9 Image Tab in the Output Specification Section

Specify the following information on this tab:

- **Image file**

Specifies the path for the image file to be output.

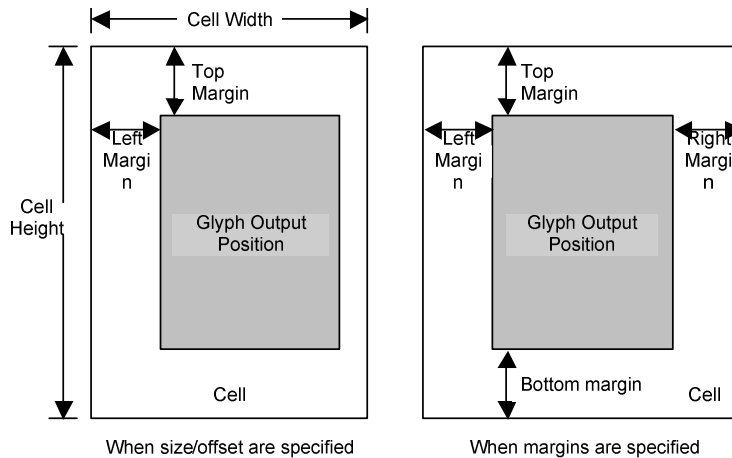
When a setting other than **by Extension** is selected for **Format**, an extension is added at conversion time if the image file extension differs from that selected in **Format**. However, if the image file extension is one of those supported by fontcvtr (.bmp or .tga), the extension will replace (rather 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 *fontcvtr Formats* (fontcvtr_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 4-10. If **Specify margin** is selected, cell size and glyph output position are specified. These parameters are shown on the right in Figure 4-10.

Figure 4-10 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.

4.4.2 brfnt/brfna

Figure 4-11 shows the **brfnt/brfna** tab used for output specifications. Input font data is converted to brfnt or brfna format for use by NintendoWare for Revolution.

Figure 4-11 brfnt/brfna Tab in the Output Specification Section

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 `.brfnt` or `.brfna`.

- **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, the font resource is created without using a glyph group file. If a glyph group file is used, a compressed archive font (brfna) 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**

Font glyphs are stored in the brfnt 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 brfna 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 Revolution. 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 Revolution. 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 brfmt format, the corresponding setting values inside the brfmt input data are used as default values.

4.5 Filter Specifications

Figure 4-12 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 brfmt 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 *fontcvtr Formats* (*fontcvtr_Format.pdf*).

Figure 4-12 Filter Specification



- **Use filter**
When selected, the specified filter is applied during conversion.

5 Precautions

5.1 Precautions when Converting Windows Fonts

Both Windows 2000 and Windows XP have 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 2000 and 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 fontcvtr. As a result, Japanese characters will not be output when using a font (like Tahoma) that does not include them.

5.2 Character Encoding Using Unicode

The fontcvtr 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, fontcvtr also handles it as U+FF5E.

Table 5-1 shows the character code conversions that fontcvtr generates for the various inputs and outputs.

Table 5-1 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] brfmt	Converts from the encoding stored in the brfmt 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] brfmt	Converts from Unicode to the specified encoding.
Text filter file	Converts to Unicode when loaded.
Letter order file	Converts to Unicode when loaded.

6 About Xerces-C++

fontcvtr uses Xerces-C++ developed by the Apache Software Foundation (<http://www.apache.org/>). Copies of the Xerces-C++ NOTICE file and license file can be found in the following directories.

- NintendoWare/Revolution/Documents/Xerces-C++/NOTICE
- NintendoWare/Revolution/Documents/Xerces-C++/LICENSE

Microsoft and Windows are the registered trademarks of Microsoft Corporation in the USA and other countries.
Other company and product names are the trademarks or registered trademarks of their respective companies.

© 2005–2008 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.