

TWL-System

fontcvtr Manual

Operating Font Converter

2009/02/18

**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/or its licensed developers 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	7
1.1	About fontcvtr	7
1.2	What fontcvtr Can Do	7
1.3	Font Licenses	7
2	File Structure	8
3	Explanation of Operations	9
3.1	GUI	9
3.2	Converting Fonts	10
3.2.1	Basic Operations	10
3.2.2	Flow for Creating a Font Resource	10
3.3	Input Specifications	11
3.3.1	BMP	11
3.3.2	NITRO Font	11
3.3.3	Windows Font	12
3.3.4	LC Font	13
3.4	Output Specifications	14
3.4.1	BMP	14
3.4.2	NITRO Font	16
3.5	Filter Specification	17
4	Character Filter Files	19
4.1	Overview	19
4.2	Structure	19
4.3	How to Use	20
4.4	Document Type Definition	20
5	Letter Order Files	21
5.1	Overview	21
5.2	Structure	21
5.3	How to Use	22
5.4	DTD	23
5.5	The Included Letter Order files	23
6	BMP Image Format	24
6.1	The Structure of BMP Images	24
6.1.1	Block	24
6.1.1.1	Grid Area	25
6.1.1.2	Margin Area	25

6.1.1.3	Cell	25
6.1.1.4	Width Line Region	25
6.1.2	Cells and Width Lines	25
6.1.2.1	Glyph Image	26
6.1.2.2	Width Line	26
6.1.2.3	Glyphs With No Output	27
6.1.3	Location Information	27
6.1.3.1	Baseline	28
	The Ascender Line and Descender Line	28
6.1.3.2	28
6.1.4	Vertical-Horizontal Identification Marker	30
6.2	BMP Output Fonts	31
6.3	BMP Input Fonts	31
7	Vertical Fonts and Vertical-Style Fonts	32
7.1	Vertical Fonts	32
7.1.1	What are Vertical Fonts?	32
7.1.2	How to Create a Vertical Font	32
7.1.3	Output BMP Images of Vertical Fonts	32
7.2	Vertical-Style Fonts	33
7.2.1	What are Vertical-Style Fonts?	33
7.2.2	How to Create Vertical-Style Fonts	33
7.3	Combinations of Vertical Writing and Vertical-Style	33
8	Cautions	34
8.1	Converting Windows Fonts	34
8.2	Unicode Inside fontcvtr	34
8.3	Warning Message About Duplicate Cells in a Letter Order File	35
8.4	Unicode in the Letter Order File	35
9	Warning and Error Messages	36
9.1	Warning Messages	36
9.2	Error Messages	36
10	About Xerces-C++	42

Code

Code 4-1 sample.xlft	19
Code 5-1 cp1252.xlft	21

Tables

Table 4-1 Character Filter Definition Elements	20
Table 5-1 Letter Order Definition Elements.....	22
Table 5-2 The Included Letter Order Files	23
Table 6-1 Interpreting Location Information Points	29
Table 8-1 Places Where Character Codes are Converted	34
Table 8-2 The 2:1 Relationship of Characters Converted from Shift_JIS to Unicode.....	35
Table 9-1 Explanation of Warning Messages	36
Table 9-2 Explanation of Error Categories.....	36
Table 9-3 Descriptions of Error Messages.....	37

Figures

Figure 1-1 Relationships Among Font-Related Resources	7
Figure 2-1 File Structure	8
Figure 3-1 Main Dialog Box of fontcvtr	9
Figure 3-2 The Basic Flow for Creating a Font Resource	10
Figure 3-3 Input: BMP.....	11
Figure 3-4 Input: NITRO Font	11
Figure 3-5 Input: Windows Font.....	12
Figure 3-6 Input: LC Font.....	13
Figure 3-7 Output: BMP	15
Figure 3-8 Specifying Cell Size and Glyph Output Position.....	16
Figure 3-9 Output: NITRO Font	16
Figure 3-10 Filter Specification	18
Figure 6-1 Example of BMP Image (Number of Blocks: 16x9).....	24
Figure 6-2 Schematic Pattern Diagram of Block	24
Figure 6-3 Schematic Pattern Diagram of Cell and Width Line	26
Figure 6-4 Left and Right Placement in Cell	26
Figure 6-5 All Location Information Pixels	27
Figure 6-6 Position of Baseline	28
Figure 6-7 Ascender Line and Descender Line Position Model Information	28
Figure 6-8 Font Height	30
Figure 6-9 The Vertical- Horizontal Identification Marker	30
Figure 7-1 BMP Output of a Vertical Font.....	32

Revision History

Revision Date	Description
2009/02/18	Revised description of error messages.
2008/08/05	Added 6.1.3 Location Information.
2008/05/30	Made revisions in line with the NITRO-System name change (from NITRO-System to TWL-System).
2008/04/08	<ul style="list-style-type: none">• Changed the format of the Revision History.• Changed the license notification for Xerces-C++.• Made revisions throughout the entire document.
2007/11/26	Updated for Version 1.1.3.
2007/04/25	<ul style="list-style-type: none">• Corrected the misaligned number for figure.• Added description about specifying the font size when using Windows fonts for text input.
2007/03/14	Support for Version 1.1.0.
2005/09/08	Corrected misprint in Letter Order file name.
2005/07/11	Corrected a misnamed function.
2005/05/25	Initial version.

1 Introduction

1.1 About fontcvtr

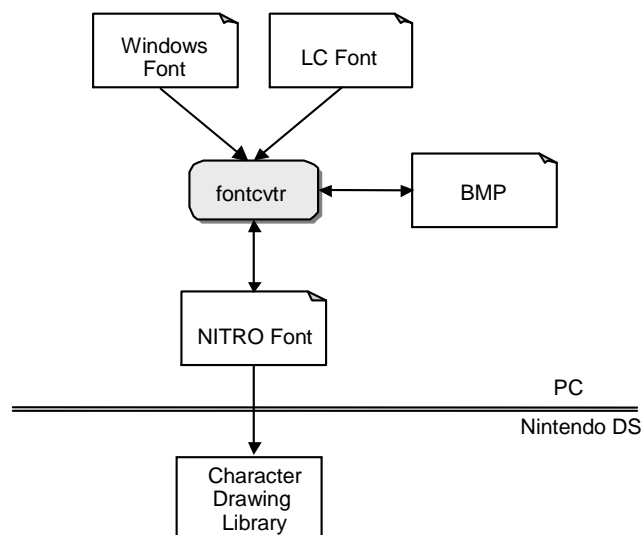
fontcvtr is a Microsoft Windows tool for creating NITRO fonts in a format that can be used by a TWL or Nintendo DS system. NITRO fonts can be used to easily draw character strings on the TWL or DS system using the TWL-System G2D Character Drawing Library.

1.2 What fontcvtr Can Do

fontcvtr converts Windows and Sharp LC fonts into NITRO fonts. This tool can also be used to write non-NITRO fonts out as BMP files. Because BMP files can be loaded and converted into NITRO fonts, the non-NITRO fonts can first be converted into and fine-tuned as BMP files before converting the BMP files into NITRO fonts. With fontcvtr, you can use glyph data from BMP files to create unique NITRO fonts. These relationships are shown in Figure 1-1.

In each step of the conversion, you can create only the fonts that you need by extracting the needed text characters from the original font set.

Figure 1-1 Relationships Among Font-Related Resources



1.3 Font Licenses

fontcvtr can convert any font installed on a PC to a NITRO font for use on the TWL or Nintendo DS system. However, a user license is required to use these non-NITRO fonts in game software that will be sold. You need to obtain licenses for each game software title.

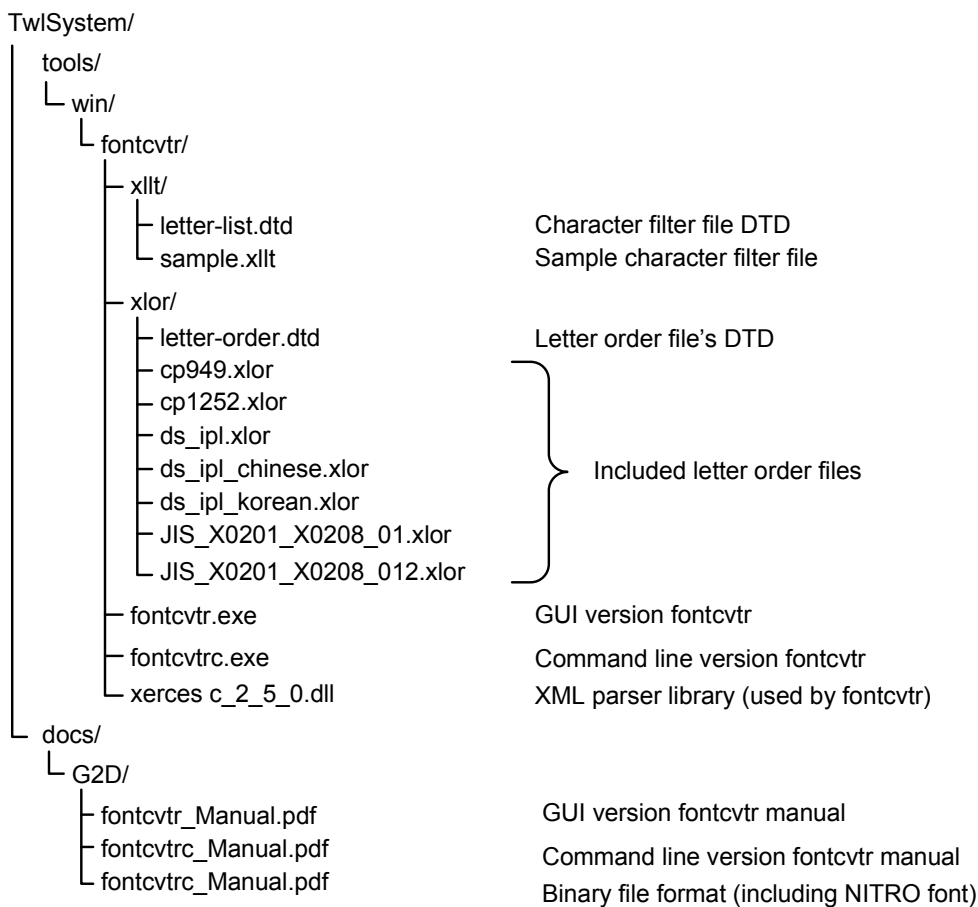
fontcvtr and TWL-System do not come with licenses for any fonts, including LC fonts.

2 File Structure

fontcvtr files are located in `TwlSystem/tools/win/fontcvtr`. The user manual is located in `TwlSystem/docs/G2D`.

The file structure of all the fontcvtr related folders and files is shown in Figure 2-1.

Figure 2-1 File Structure



For more details, see Chapter 4 Character Filter Files and Chapter 5 Letter Order Files.

This fontcvtr manual covers topics shared by the fontcvtr (GUI version) and fontcvtrc (command line version), as well as topics unique to fontcvtr. To read topics unique to the command line version, read the *fontcvtrc Operation Manual*.

When you start fontcvtr, a settings file called `fontcvtr.ini` is created in the same directory as `fontcvtr.exe`. (Note that `fontcvtr.ini` is not shown in Figure 2-1). The display state of fontcvtr is automatically stored in this settings file so the display is in the same state the next time fontcvtr runs.

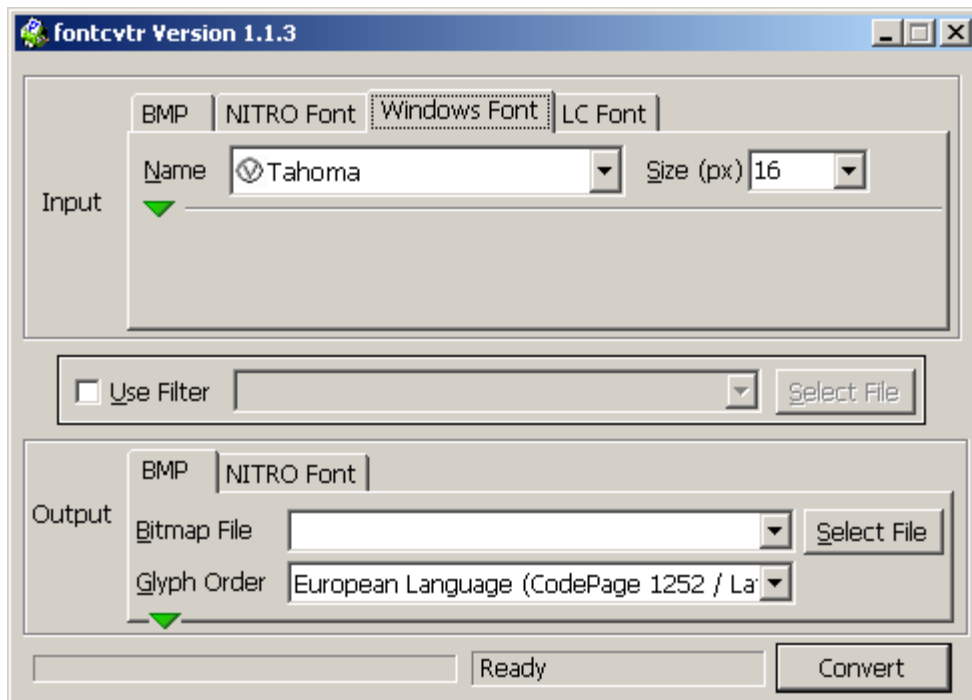
3 Explanation of Operations

3.1 GUI

Figure 3-1 shows a screen shot from fontcvtr.

The fontcvtr GUI is a dialog box; all operations can be performed from this dialog box.

Figure 3-1 Main Dialog Box of fontcvtr



The main dialog box can be divided into four components: Input Specifications at the top, Filter Specification in the middle, Output Specifications at the bottom, and Status Display at the very bottom of the dialog box. The Input Specifications is used to set the resource and options for the font-conversion source. The Output Specifications is used to specify the font-conversion target. Once the settings have been specified, click **Convert** and fontcvtr executes the font conversion process.

The Status Display shows the status of the process in a progress bar.

All fontcvtr operations support file drag and drop into fields where files can be specified.

3.2 Converting Fonts

3.2.1 Basic Operations

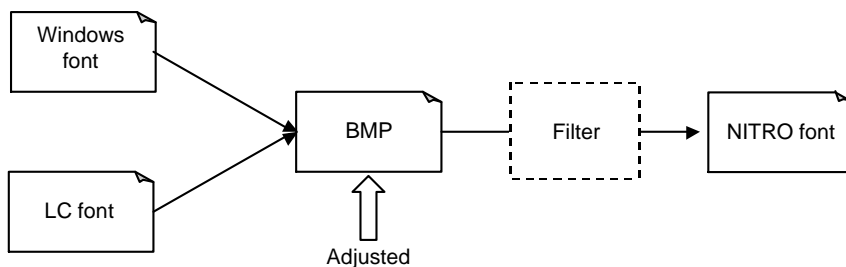
Follow these procedures when using fontcvtr to convert fonts.

1. Open the tab for the conversion-source resource and set the input resource. For further detail, see section 3.3 Input Specifications.
2. Open the tab for the output-target resource and set the output resource. For further detail, see section 3.4 Output Specifications.
3. Specify a filter as necessary. If you are not going to use a filter, clear the **Use Filter** check box. For further detail, see section 3.5 Filter Specification.
4. Click **Convert**.
5. The conversion process executes. If there is a problem in the input resource, the output resource, or either of their settings, a warning message window is displayed. If fontcvtr cannot compensate for the problem, an error displays in the message window and the conversion fails. If no error is displayed or if the displayed message is only a warning, then the conversion process succeeds. For further details, see Chapter 9 Warning and Error Messages.

3.2.2 Flow for Creating a Font Resource

The Windows or LC font is first converted to a BMP file and creates the base font. Adjustments are made if necessary and then converted from a BMP file to a NITRO font. A filter can then be used so that only text characters needed by the application are stored in the NITRO font.

Figure 3-2 The Basic Flow for Creating a Font Resource

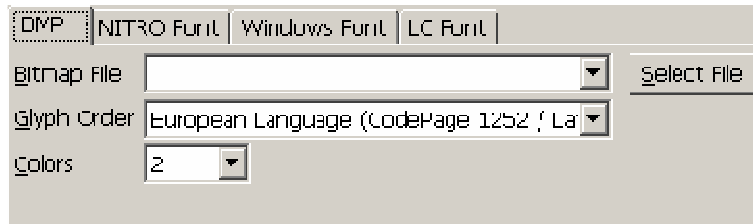


3.3 Input Specifications

3.3.1 BMP

Figure 3-3 shows the **BMP** tab of the Input Specifications dialog box. Use this tab to specify a BMP file as the input source, as explained further in Chapter 6 BMP Image Format.

Figure 3-3 Input: BMP



- **Bitmap File**

Specifies the conversion source BMP file.

- **Glyph Order**

Selects the letter order of the BMP file. For further details, see Chapter 5 Letter Order Files.

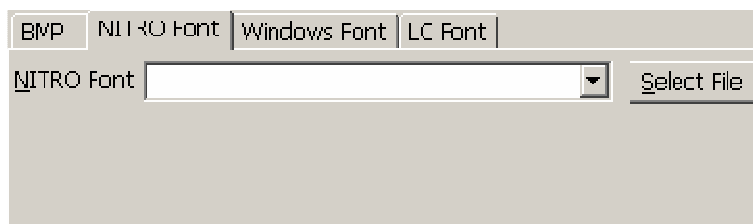
- **Colors**

Specifies the valid number of colors in the BMP file. It also specifies the number of colors in the font that are output. If you use more colors than are specified, the additional colors are treated as transparent colors. For example, if you specify 4 for the drop-down **Colors** box, the BMP color palette output is color numbers 0-3, and all color numbers above 3 are treated as transparent colors.

3.3.2 NITRO Font

Figure 3-4 shows the **NITRO Font** tab of Input Specifications. Use this tab to specify a NITRO font that has been created by fontcvtr as the input source.

Figure 3-4 Input: NITRO Font



- **NITRO Font**

Specifies a NITRO font as the conversion source.

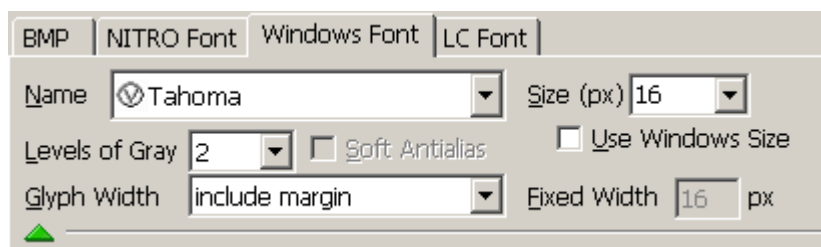
3.3.3 Windows Font

Figure 3-5 shows the **Windows Font** tab of the Input Specifications dialog box. Use this tab to specify one of the installed Windows fonts as the input source.

Note that when a Windows font is entered as the input source, the text characters of that font in the entire Unicode (UCS2) range are passed to the output. If BMP is specified for the output, then the output text characters are restricted by the Letter Order file. If a NITRO font is specified for the output but a Letter Order file is *not* specified, then *all* text characters for that font will be output. Depending on the NITRO font specified, the output could result in tens of thousands of text characters.

In the initial state, the **Windows Font** tab only shows a font **Name** and **Size** drop-down list box. By clicking the green triangle, you can show or hide other fields.

Figure 3-5 Input: Windows Font



- **Name**

Specify the Windows font name to use as the conversion source. Click the down arrow to view a list of all fonts installed on the PC. Fonts whose background is the tooltip color are raster fonts (bitmap fonts), while those shown with a white background are vector fonts (outline fonts).

Font names preceded by @ are displayed vertically.

- **Size**

Specify the conversion source font size in pixels. Depending on the font, the value might not adhere strictly to the size. Especially with raster fonts, the difference might be striking if a small font size is specified.

- **Use Windows Size**

Selecting this option causes the font size specified in the **Size** field to be interpreted in the same way as general Windows software. This feature allows you to match the output size when using the output of fontcvtr together with the output from other Windows software.

When the box is not selected, fontcvtr adjusts the output size so that tall characters, such as those with accent marks, are the specified size. In contrast, general Windows software adjusts the font size so that the height of normal characters are the specified size. Selecting this option, therefore, results in larger character sizes.

- **Levels of Gray**

Specify the number of levels in the gray scale output. If the value is set to 2, the output will be black and white only. This option is not available for raster fonts, because they are always set to 2.

- **Soft Antialias**

When performing multi-tone output (output other than two levels of gray), antialiasing is performed on the glyphs that are output. Selecting this option softens antialiasing, which may improve the quality of the glyphs that are output.

- **Glyph Width**

This option specifies whether the text character width of the output font includes a blank width on either side of each Windows font character. When set to **glyph only**, the blank space to the right and left is not included, and the text character width is the same as the glyph width. The setting, **glyph only (keep space)** is essentially identical to **glyph only**, but the character width for blank spaces is identical to that when **include margin** is specified. To include blank spaces, set **Glyph Width** to **include margin**. When **fixed width** (the default value), is selected, all text characters are output with the text character width set in **Fixed Width** (described in section 3.3.4 LC Font).

When **glyph only** is selected, the character width of a blank space is 0 because there are no glyphs for blank characters. The display will appear as if there are no blank characters. When using blank spaces, either specify **glyph only (keep space)** or directly adjust the text character width after an image file has been output.

- **Fixed Width**

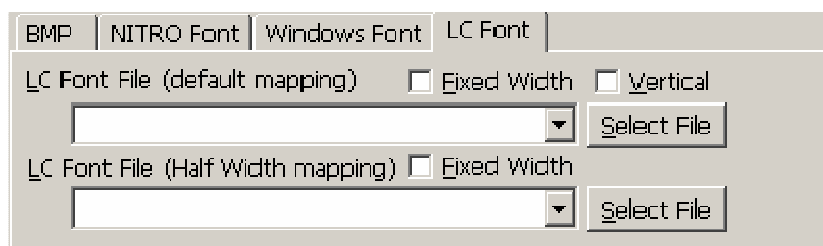
Specify the text character width in pixels. This setting can only be configured when **Glyph Width** is set to **fixed width**.

3.3.4 LC Font

Figure 3-6 shows the **LC Font** tab of the Input Specifications dialog box. This tab is used to specify LC Font as the input.

If you would like to use LC Fonts, contact support@noa.com.

Figure 3-6 Input: LC Font



- **LC Font File (default mapping)**

Specifies the conversion source LC Font file.

- **LC Font File (Half Width mapping)**

Specifies the conversion-source LC Font file. The difference between this and *default mapping* is that the font is output with half-width character codes. Characters are stored with full-width character codes in LC fonts but *Half Width mapping* replaces these with half-width character codes and passes them to the output. This makes it possible to use the alphanumeric characters of the LC Font as ASCII characters.

If *default mapping* and *Half Width mapping* are both specified, the two fonts are combined into a single font for conversion. By using *default mapping* for the full-width LC Font and *Half Width mapping* for the half-width LC Font, you can display all text characters from ASCII to level 2 Japanese kanji characters. (See the Letter Order file `JIS_X0201_X0208_012.xlor`.)

- **Fixed Width**

All LC Font text characters have the same text character width, but if **Fixed Width** is not selected, the spaces on either side of the glyphs are removed during the conversion process and the font is output as a proportional font. This setting has nearly the same effect as selecting **glyph only** for **Glyph Width** when converting a Windows font. However, in the case of the LC Font, every text character has a 1 pixel space attached to its left side to prevent letters from touching.

- **Vertical**

Treats the LC font as a font for vertical writing. For details, see section 7.1 Vertical Fonts.

3.4 Output Specifications

In the initial state, each tab in Output Specifications displays only the settings that need to be configured. By clicking the green triangle, you can show or hide other fields.

3.4.1 BMP

Figure 3-7 shows the **BMP** tab of Output Specifications dialog box. The input source font data is output as a BMP file based on the image settings described in Chapter 6 BMP Image Format. The BMP file output can be used as a BMP entry in the Input Specifications dialog box.

Figure 3-7 Output: BMP

- **Bitmap File**

Specify the conversion-destination BMP file.

- **Glyph Order**

Select the order to output the text characters in the BMP files. For further details, see Chapter 5 Letter Order Files.

- **Size/Offset and Margin**

Specify **Size/Offset** and then specify the cell size and glyph output position with the parameters shown on the left in Figure 3-8. Otherwise, specify **Margin** and then specify the cell size and glyph output position with the parameters shown on the right in Figure 3-8.

- **Place center**

Select this option to center the glyph in the cell. The **Left** and **Top** values are set automatically. This feature is available when **Size/Offset** has been selected.

- **Right=Left, Bottom=Top**

Select this option to equate the **Right** and **Left** values, and the **Bottom** and **Top** values. This feature is available when **Margin** has been selected.

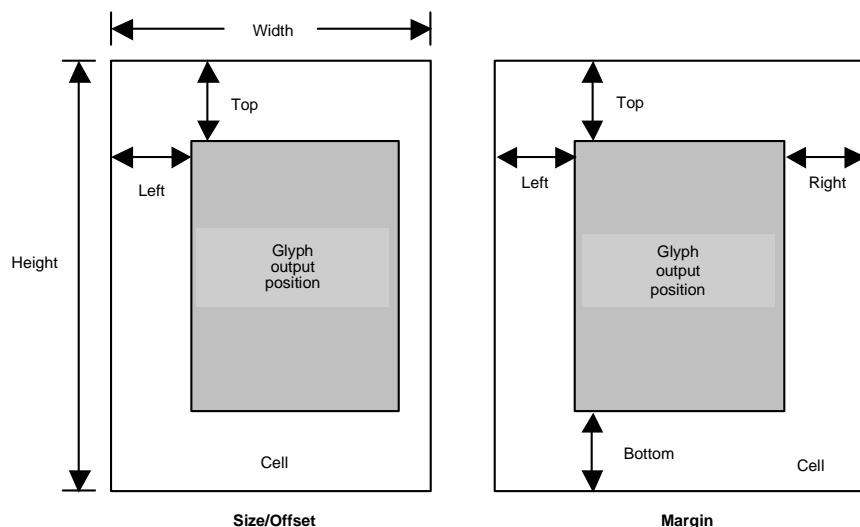
- **Left, Top, Right, Bottom**

Specify the parameters shown on the right in Figure 3-8 when **Margin** has been selected. When **Size/Offset** is selected, these parameters change to **Width**, **Height**, **Left**, and **Top**.

- **Draw Grid**

Select this option to draw a grid.

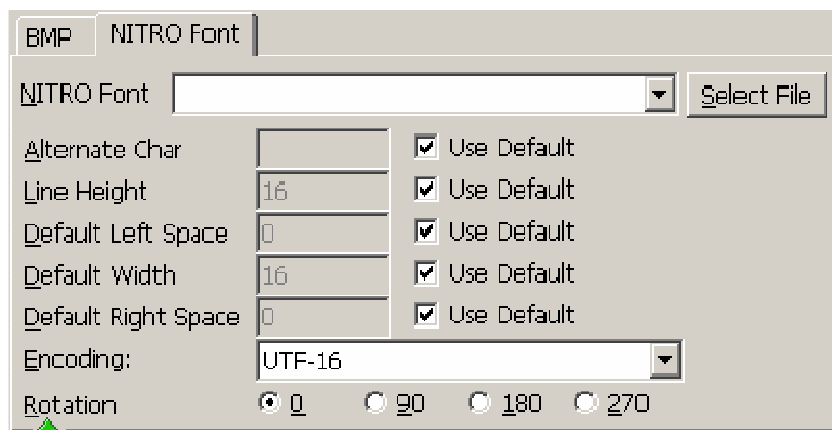
Figure 3-8 Specifying Cell Size and Glyph Output Position



3.4.2 NITRO Font

Figure 3-9 shows the **NITRO Font** tab of Output Specifications dialog box. This tab configures the settings to convert the input source font data into a font resource to be used by the Character Drawing Library. Ultimately, font data is output in this format.

Figure 3-9 Output: NITRO Font



- **NITRO Font**

Specify the output target NITRO font file.

- **Alternate Char**

Specify which alternate text characters the Character Drawing Library should display if the font is not supported. Enter the character code of the text character itself. Character codes are entered according to C-language rules in octal, decimal, or hexadecimal. If only one text character is entered, it is directly mapped to the alternate text character. When **Use Default** is selected, the text character

with the lowest character code is used as the alternate text character. (In most cases, this alternate text character is half-width.)

- **Line Height**

Specify the height of one line. Values range from 0 to 255. The Character Drawing Library uses this value for the basic line height. If **Use Default** is selected, the value is set to the height of the font.

- **Default Left Space**

Specify the default width of the left space for text characters that do not have their own left space. Values range from -128 to 127. When **Use Default** is selected, the value is set to 0.

- **Default Width**

Specify the default glyph width for text characters that do not have their own glyph width. Values range from 0 to the largest glyph width possible for the font. When **Use Default** is selected, the value is set to the largest glyph width possible.

- **Default Right Space**

Specify the default width of the right space for text characters that do not have their own right space. When **Use Default** is selected, the value is set to 0.

- **Use Default**

The **Use Default** check box is located to the right of each option field. Select it to set the option to the default value explained in each of the previous sections. However, note that if a NITRO font is entered in the Input Specifications dialog box, then the default values correspond to the values configured in the NITRO font.

- **Encoding**

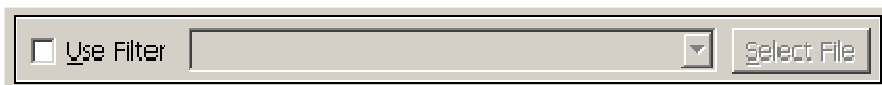
Specify the text character encoding format that corresponds to the output font.

- **Rotation**

Specifies the rotation amount when creating a vertical-style font. The various options indicate the counterclockwise rotation angle in degrees. Selecting 0 will create a normal font. For details, see section 7.2 Vertical-Style Fonts.

3.5 Filter Specification

Figure 3-10 shows the Filter Specification dialog box. If a text character filter file is specified, text characters not included in the file are not output. The filter provides a method to create a compact NITRO font that includes a minimum set of text characters. If a character filter file is not specified, then all text characters that belong to the specified input font are passed to the output. For further details, see Chapter 4 Character Filter Files.

Figure 3-10 Filter Specification

- **Use Filter**

Select this option to apply a filter during the conversion process.

4 Character Filter Files

4.1 Overview

The character filter file specifies which text characters fontcvtr will output. Use this file so the game program includes the minimum set of NITRO font text characters.

Character filter files are written in XML and have an XLLT extension.

4.2 Structure

The XML structure of the character filter file complies with the protocol for TWL-System XML files (NNSXML protocol). To learn more about the NNSXML protocol, see [TwlSystem/docs/Readme/DataFormatRule.pdf](#). This section explains the character filter file structure that is not defined by the NNSXML protocol. Specifically, the section describes the internal structure of the *root* element and the *body* element of the character filter file.

The sample character filter file `sample.xllt` is shown in the following code example. When fontcvtr is run using this character filter file, a total of only 19 text characters are output, including the letters fontcvtr, the kana characters あいうえかきくけ, the kanji characters 任天堂, and a half-width space.

Code 4-1 sample.xllt

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE letter-list SYSTEM "letter-list.dtd">

<letter-list version="1.0">
  <head>
    <create user="nintendo" date="2005-02-18T10:51:13" />
    <title>xllt sample</title>
    <comment>Character Filter file sample</comment>
  </head>

  <body>
    <letter>
      fontcvtr
      あいうえかきくけ
      任天堂
    </letter>
  </body>
</letter-list>
```

The root element of the text character filter is *letter-list*, and this filter contains only two elements: the *head* element defined by the NNSXML protocol and the *body* element. The *head* element and the incorporated elements are all defined by the NNSXML protocol, and therefore, will not be explained. The *body* element incorporates the *letter* element, and the *letter* element contains all the text characters that will be output. Note that blank-space text characters in the *letter* element (for example, the half-width space and the tab characters) are ignored, and the letters that are output always include

a half-width space. This information is described below in Table 4-1 Character Filter Definition Elements.

Table 4-1 Character Filter Definition Elements

Element	Includable Elements	Description
	Attribute	
letter-list	head, body	Defines the text character filter. This element is the root element of the character filter file. The required attribute version is specified currently as 1.0.
	version (required)	
letter	None	Defines the text characters in the NITRO font. Characters are specified by entering the text characters you want to include directly.
	None	

4.3 How to Use

See section 3.5 Filter Specification to learn how to use Character Filter files.

4.4 Document Type Definition

The Document Type Definition (DTD) describes the structural rules for an XML document. It is written as shown in Table 4-1. The DTD for the character filter file is shown in `letter-list.dtd` and is located in the same folder as `fontcvtr.exe` inside the `xllt` folder.

5 Letter Order Files

5.1 Overview

The Letter Order file defines the text character order in the input and output of the BMP files by fontcvtr. The Letter Order file defines the number of text characters on the horizontal and vertical axis in the BMP image and also the order in which the characters are drawn. Since characters not defined in the Letter Order file are not output, this file can also operate as a character filter.

Letter order files are written in XML and named with an XLOR extension.

5.2 Structure

As with character filter files, the XML structure of Letter Order files complies with the NNSXML protocol. The code example below shows a section of the Letter Order file `cp1252.xlor`.

Code 5-1 cp1252.xlor

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE letter-order SYSTEM "letter-order.dtd">

<letter-order version="1.0">
  <head>
    <create user="nintendo" date="2005-02-18T10:51:13" />
    <title>European Language (CodePage 1252 / Latin-1)</title>
    <comment>Windows Code ~(omitted)~ 8859-1 (Latin-1).</comment>
  </head>
  <body>
    <area width="16" />

    <order>
      <sp/> ! " # $ % & ' ( ) * + , - . /
      0 1 2 3 4 5 6 7 8 9 : ; < = > ?
      @ A B C D E F G H I J K L M N O
      P Q R S T U V W X Y Z [ \ ] ^ _
      ` a b c d e f g h i j k l m n o
      p q r s t u v w x y z { | } ~ <null/>

      &#x20AC; <null/> ~(omitted)~ &#x00FE; &#x00FF;
    </order>
  </body>
</letter-order>
```

The root element in the letter order definition is *letter-order*, and this definition contains only two elements: the *head* element defined by the NNSXML protocol and the *body* element. The *head* element and incorporated elements are all defined by the NNSXML protocol, and therefore, will not be explained. The *body* element incorporates the *area* element, and specifies the number of characters on the vertical and horizontal axis of the BMP image and the *order* element that specifies the drawing order of the characters.

The *area* element has no content but has two attributes: *height* and *width*. The attributes *height* and *width* define the number of characters on the vertical and horizontal axis. Both of these attributes can

be omitted. When *width* is omitted, the default value is 16 text characters. When *height* is omitted, the default value is set so that all characters are output.

The *order* element arranges the text characters, and the same order is used to make the letter order in the BMP image. The first text character is placed in the upper-left corner of the BMP image, and the second text character follows immediately to the right. Using the specified letter order, this left-to-right order is used to position the rest of the text characters to fill the first line to the right edge of the BMP image and then move to the second line. The blank-space characters (the half-width and the tab spaces) are ignored in the *order* element, so you must use the *sp* element to output a half-width space. In addition, you can use the *null* element to skip ahead one space in the output. These different elements are listed in Table 5-1.

Table 5-1 Letter Order Definition Elements

Element	Includable Elements	Description
	Attribute	
letter-order	head, body	Defines the order of text characters. This element is the root element of the Letter Order file. The required attribute version is presently specified as 1.0.
	version (required)	
area	None	Defines the number of characters to output on the vertical and horizontal axis of the BMP image. The width attribute defines the number of characters on the horizontal axis. When it is omitted, the default value is 16 text characters. The height attribute defines the number of text characters on the horizontal axis. When it is omitted, the default value is set to output all characters specified in the order element.
	width (can be omitted)	
	height (can be omitted)	
order	sp, null	Defines the output characters and output order. Specify text characters by arranging them in the output order. The same character should not appear more than once.
	None	
sp	None	Specify a half-width space inside the order element. You need to use this element to output a half-width space because direct descriptions of half-width spaces are ignored.
	None	
null	None	Specify no text character output. Provides a way to skip a character's space.
	None	

5.3 How to Use

When fontcvtr starts, it reads the XLOR files in the `xlor` folder (the same folder that contains `fontcvtr.exe`). The XLOR files (Letter Order files) show the *title* element content in each file as a list item in **Glyph Order**. Letter Order files, therefore, must have an XLOR extension and be saved in the `xlor` folder.

If you start fontcvtr after adding a new Letter Order file to the `xlor` folder, the new file appears as a list item in **Glyph Order**. If you select the newly added file as an input and output BMP file, the new Letter Order file is used to input and output BMP files.

If the new Letter Order file fails to load at startup for any reason, the new file will not display in **Glyph Order** and cannot be used. The displayed error message provides troubleshooting assistance.

5.4 DTD

The DTD for the Letter Order file is described in `letter-order.dtd`, which is located in the same folder as `fontcvtr.exe`, inside the `xlor` folder.

5.5 The Included Letter Order files

Seven Letter Order files (see Table 5-2) are included with fontcvtr. Except for the characters that are output, the seven included Letter Order files share the same specifications. Characters are output in the order of their character code, with 16 characters in the horizontal direction.

Because a Letter Order file only specifies character codes, it cannot handle differences in shape for the characters in each font. For example, if a Japanese-language Windows font is the source, a BMP is the output, and the Letter Order file is `cp1252.xlor`, then the yen sign (¥) is output even though the font should show a backslash sign.

Table 5-2 The Included Letter Order Files

Filename	Displayed by fontcvtr
Explanation	
<code>cp1252.xlor</code>	European Language (CodePage 1252 / Latin-1)
Outputs alphanumeric and some European-language characters. This is called Windows code page 1252; it includes ISO 8859-1 (Latin-1) and all ASCII characters.	
<code>ds_ipl.xlor</code>	DS-IPL font table
Outputs the same characters as the IPL font of the Nintendo DS system. In addition to <code>cp1252.xlor</code> and <code>JIS_X0201_X0208_01.xlor</code> , it has a Unicode extended character region for pictographs. However, because characters with the same character codes are contained in <code>cp1252.xlor</code> and <code>JIS_X0201_X0208_01.xlor</code> , the corresponding characters in <code>JIS_X0201_X0208_01.xlor</code> have been replaced with <code><null></code> . In addition, because European and Japanese characters are included, set Encoding to UTF-16 or UTF-8 for NITRO font output.	
<code>ds_ipl_chinese.xlor</code>	DS-IPL font table for Chinese
Outputs the same characters as the IPL font of the Nintendo DS for China. It is the same as the <code>ds_ipl.xlor</code> font table, with Chinese kanji (GB2312) replacing the Japanese kanji portion.	
<code>ds_ipl_korean.xlor</code>	DS-IPL font table for Korean
Outputs the same characters as the IPL font of the Nintendo DS for Korea. It is the same as the <code>ds_ipl.xlor</code> font table, with hangul characters (KS X 1001) replacing the Japanese kanji portion.	
<code>JIS_X0201_X0208_01.xlor</code>	Japanese Level 1 (JIS X 0208)
Outputs Japanese characters, including all level 1 kanji. This also includes half-size alphanumeric characters and half-width kana characters.	
<code>JIS_X0201_X0208_012.xlor</code>	Japanese Level 1,2 (JIS X 0208)
Outputs level 2 kanji in addition to the level 1 kanji of <code>JIS_X0201_X0208_01.xlor</code> .	
<code>cp949.xlor</code>	Korean (CodePage 949 / UHC)
Outputs alphanumeric characters, hangul, and kanji. This is known as Windows CodePage 949 or UHC (Unified Hangul Code).	

6 BMP Image Format

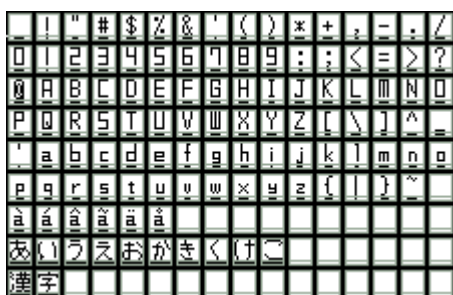
6.1 The Structure of BMP Images

The BMP images handled by fontcvtr have a block and cell structure. The BMP image comprises a grid of blocks, and each block comprises a cell. There are no spaces between blocks.

6.1.1 Block

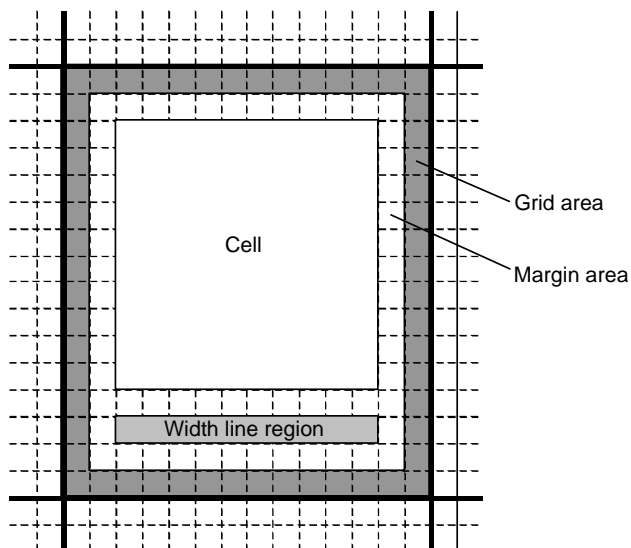
The BMP image blocks are arranged without any spaces between the blocks. The number of blocks in the horizontal and vertical axes is defined in the Letter Order file (see Figure 6-1).

Figure 6-1 Example of BMP Image (Number of Blocks: 16x9)



Each block includes a cell and a width line. The block shown in Figure 6-2 has a width of 14 pixels and a height of 16 pixels. The thick solid line shows the boundaries of the block and the thin dashed lines show the boundaries of each pixel.

Figure 6-2 Schematic Pattern Diagram of Block



6.1.1.1 Grid Area

The border of 1-pixel squares around the inside of the block boundary is called the grid area. In Figure 6-2, this area is shaded in dark gray.

If **Draw Grid** is selected, the grid area fills with black when the BMP file is output, which makes **Draw Grid** look like there is a 2-pixel grid in neighboring blocks. If **Draw Grid** is not selected, this grid area is filled with white.

When BMP is input, this grid area is simply ignored.

6.1.1.2 Margin Area

The 1-pixel squares around the inside of the grid area are called the margin area. This area is set so the grid, cell, and width lines do not touch. There is also a 1-pixel margin area between the cell and the width line.

When the BMP file is output and **Draw Grid** is selected, the margin area is filled with dim green. If **Draw Grid** is not selected, this area is filled with white.

When BMP is input, this area is checked to see if it is filled with a single color. A warning message is displayed if it is filled with more than two colors.

6.1.1.3 Cell

The larger block contained by the margin area is called the cell. The glyph image is drawn in cell the area. Because the height of the width line is set to 1 pixel, the cell size is as follows.

Cell width = Block width - 4 pixels

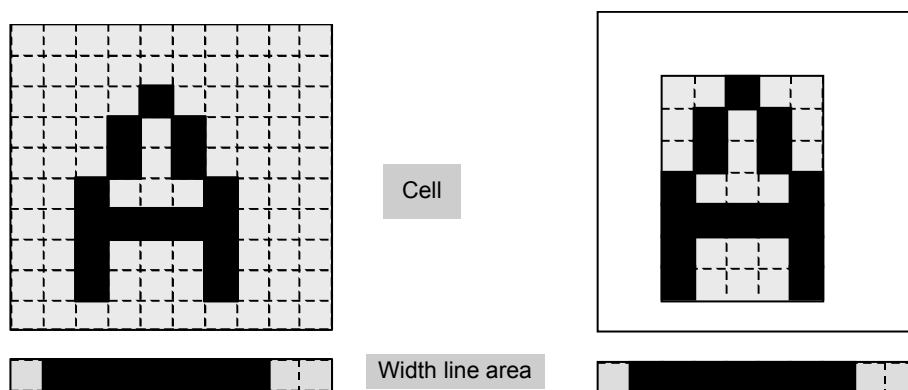
Cell height = Block height - 6 pixels

6.1.1.4 Width Line Region

The width line is the smaller area within the margin area and has a height of 1 pixel that extends horizontally. It defines the width of the text character and the relative location of the glyph image.

6.1.2 Cells and Width Lines

Figure 6-3 shows the pattern diagram of a width line area and a representative cell with a width of 10 pixels and a height of 10 pixels.

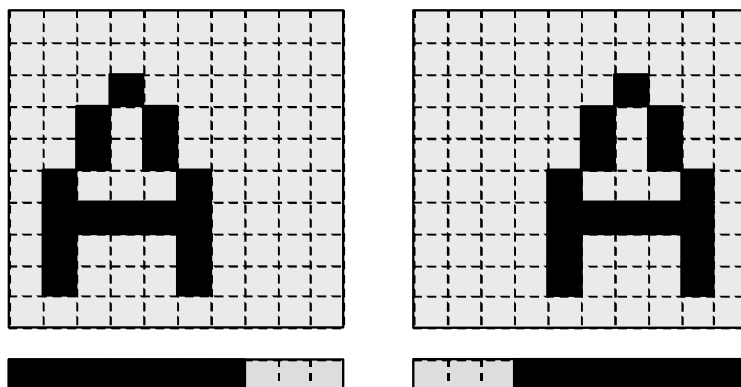
Figure 6-3 Schematic Pattern Diagram of Cell and Width Line

6.1.2.1 Glyph Image

The glyph image is the only thing placed inside the cell.

fontcvtr searches inside the cell for the smallest rectangle that contains dots that are not white and treats this as the glyph image. Thus, white parts around the glyph image are ignored and do not affect the output font. In Figure 6-3, the glyph image on the left is output as the rectangular image on the right, with a width of 5 pixels and a height of 7 pixels.

If the glyph image is located in the same relative location as the width line, then it does not matter whether it is located on the left or right side of the cell. In both cases shown in Figure 6-4, the font will be output just like the font shown in Figure 6-3.

Figure 6-4 Left and Right Placement in Cell

6.1.2.2 Width Line

A single line is drawn in the width-line area. If the line is broken into two segments, an error occurs.

The width line prescribes both the width of the character and the width of the empty space to the left and right of the character.

Because the length of the width line defines the text character width, the text character width can be set to be narrower than the glyph image. When the text character width is reduced, the text character

appears to overlap preceding text characters. The left text character shown in Figure 6-4 has a text character width of 7 pixels.

The left space width is the difference between the left end of the width line and the left edge of the glyph image. When text characters are drawn with the Character Drawing Library, the glyphs are drawn with space equal to this left-space width. The left text character shown in Figure 6-4 has a 1-pixel left-space width. Therefore, when the text character is drawn, the glyph is drawn with a 5 pixel width and is shifted by 1 pixel to the right from the specified coordinate.

Like the left-space width, the right-space width is the difference between the right end of the width line and the right edge of the glyph image. The text character shown on the right of Figure 6-4 has a right-space width of 1 pixel.

6.1.2.3 Glyphs With No Output

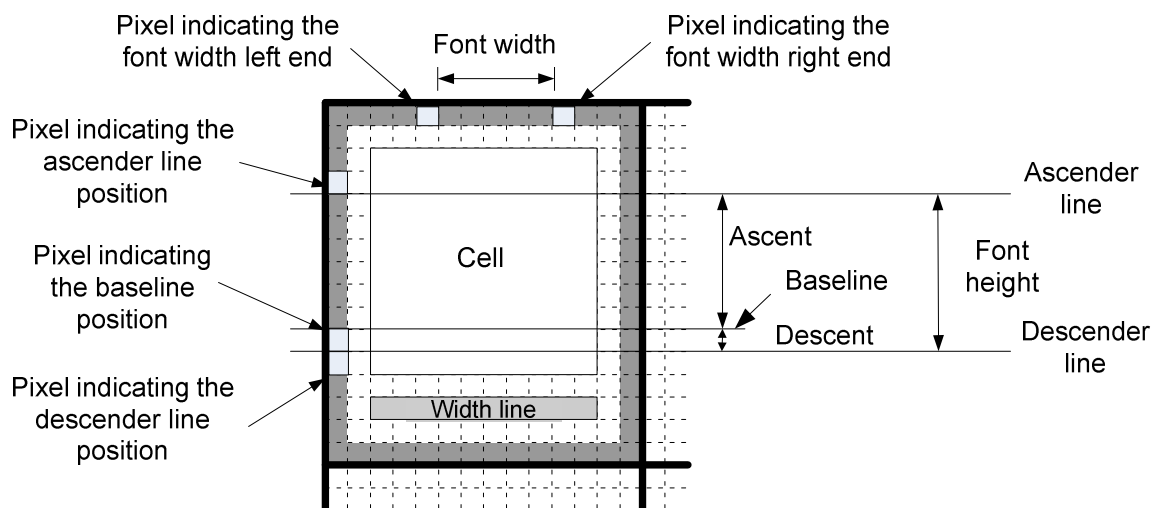
If the cell contains nothing but white dots and the width line is 0, then the glyph is not passed to output. This method can be used to control glyph output without using a text character filter. However, if a text character is specified but not passed to output, a warning message is displayed.

6.1.3 Location Information

In the block at the very top of the image, there are several pixels that specify values used as reference locations for a character string when drawing character strings. Five pixels are used to specify the standard locations for the following four items: the baseline, the ascender line, the descender line, and the font width. These values are shared by the entire font and cannot be specified for each individual character. Font width is not used under the current G2D library.

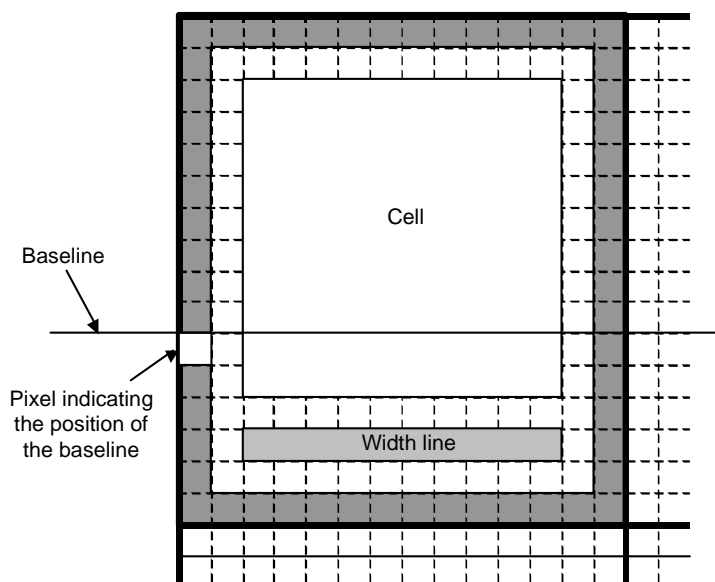
All points exist in the grid area. These are indicated using white when a grid is drawn and in black when no grid is drawn. The pixel indicating the baseline position must exist, but the remaining four points may be omitted. A block including all five points is shown in Figure 6-5.

Figure 6-5 All Location Information Pixels



6.1.3.1 Baseline

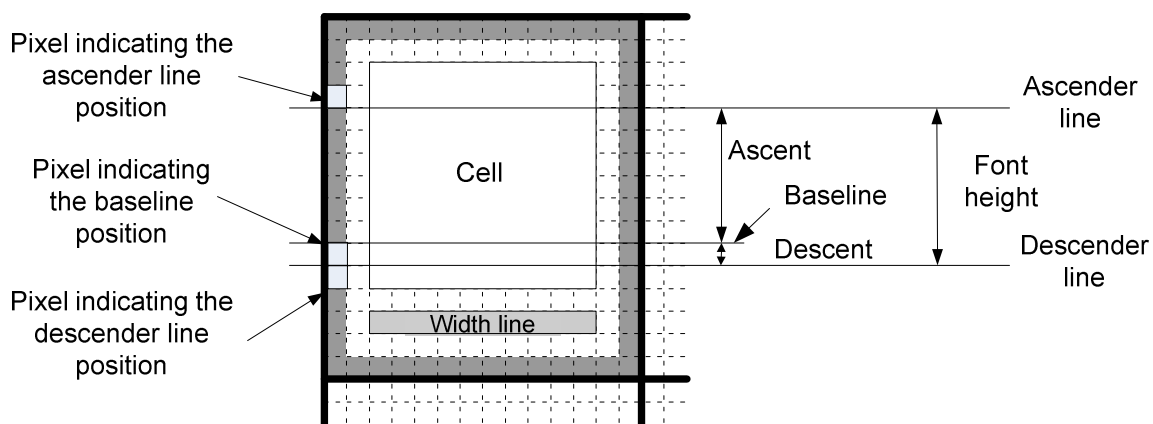
Figure 6-6 Position of Baseline



The baseline position exists on the left side of the grid area, expressed by a single pixel. The top edge of this pixel is the baseline position. In other words, the baseline exists between pixels.

6.1.3.2 The Ascender Line and Descender Line

Figure 6-7 Ascender Line and Descender Line Position Model Information



The ascender line and descender line are each represented by a single point located on the left side of the grid area. Both the ascender line and descender line exist between pixels. The ascender line is located beneath the pixel indicating its position, while the descender line is located above the pixel that indicates its position.

The ascender line must be located above the baseline, and the descender line must be located below the baseline. Also, the pixel indicating the descender line position can overlap the pixel indicating the baseline position. A descent of 0 results in this case. The ascent and descent both have values greater than or equal to 0.

These points may be omitted. Locations can be interpreted based on the number of specified points, as shown in Table 6-1. If one point is specified, the ascender line position and descender line position have not been specified. The top boundary of the tallest character is the ascender line, and the bottom boundary of the deepest descending character defines the descender line. For example, the font height for the two characters shown in Figure 6-8 is 9 pixels.

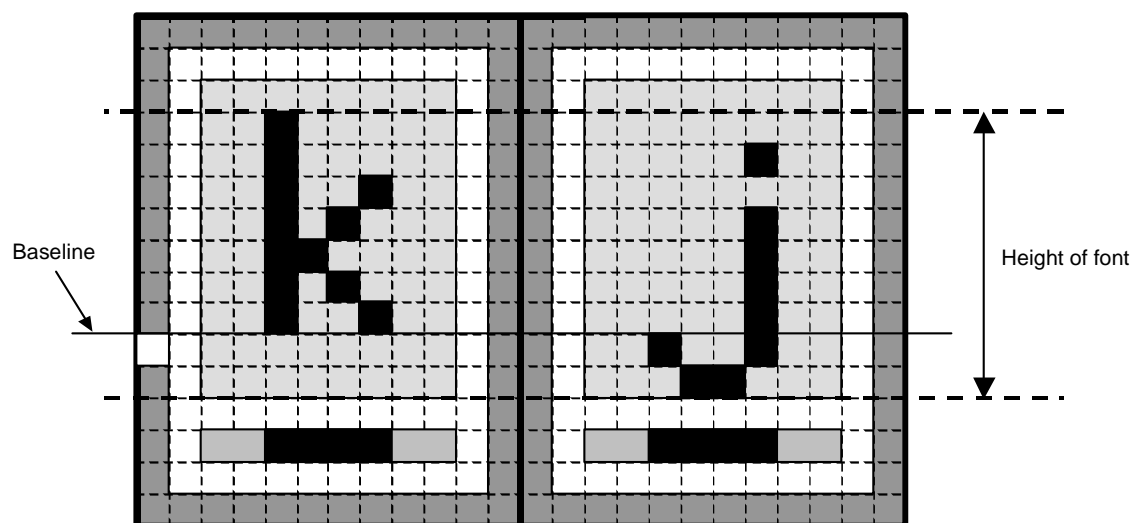
When images are output by fontcvtr, the pixels indicating the ascender line and descender line may or may not be drawn. They are never drawn for conversions that take a Windows font as input and they are always drawn for conversions that take a brfnt file as input. When an image is input, the pixels indicating the ascender line and descender line are drawn if they are included in the image, and not drawn if they are not included. If the size of a cell is too small to show the pixels that define the ascender and descender lines, the cell size is automatically increased to accommodate that font image.

The ascender line is treated as the top edge of a character and the descender line is treated as the bottom edge. Furthermore, the distance between the ascender line and descender line is referred to as the font height. This is used as the standard size to vertically scale characters.

Table 6-1 Interpreting Location Information Points

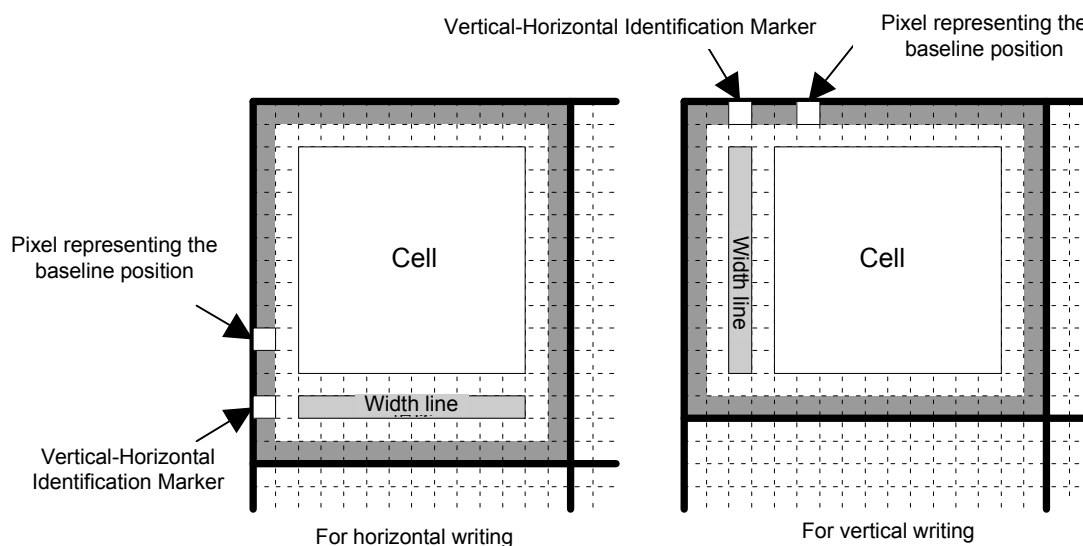
Number of Points	First Point from Top	Second Point from Top	Third Point from Bottom
0	Error		
1	Point indicating the baseline	-	
2	Point indicating the ascender line	Point indicating the baseline and descender line	-
3	Point indicating the ascender line	Point indicating the baseline	Point indicating the descender line
4 or more	Error		

Figure 6-8 Font Height



6.1.4 Vertical-Horizontal Identification Marker

Figure 6-9 The Vertical- Horizontal Identification Marker



The upper left-most block in a BMP image contains a one-pixel point to identify whether the font is used for vertical or horizontal writing. For horizontal writing, this point is located at the left edge of the same row that contains the width line. For vertical writing, this point is located at the top edge of the same column that contains the width line. The point is white if a grid is being drawn, and black if a grid is not being drawn. For compatibility with previous versions, images with no vertical-horizontal identification marker output horizontal writing. For more about vertical writing, see section 7.1 Vertical Fonts.

6.2 BMP Output Fonts

fontcvtr outputs a BMP with an 8-bit color index per pixel (256 colors).

The BMP color palette is output as gray scale from white to black in the gradation levels of the font. For example, if the font has a four-level gradation, the color palette is created with the values RGB=(255,255,255), (170,170,170), (85, 85, 85), (0, 0, 0), set in that order. Following these values is the output for the color that will be used to fill margin areas.

6.3 BMP Input Fonts

For input, fontcvtr accepts BMP color indexes with 1, 4, or 8 bits per pixel (2, 16, or 256 colors). Additional requirements include:

Block width * number of blocks in horizontal direction = BMP width

Block height * number of blocks in vertical direction = BMP height

There must be no spaces between blocks in the BMP, and no margins.

The number of blocks on the horizontal or vertical axis is determined by the Letter Order file, and the BMP width or height is determined by the BMP file. The block width and height are calculated once these values have been determined. Both the width and the height must be integer values.

fontcvtr ignores all color information contained in the BMP file. The only relevant color information is the color index, and when the BMP image is read, the color index is passed on as is by the glyph image contained in the NITRO font. To output the same BMP image on a TWL or Nintendo DS system, you must draw the text characters using the same order as the BMP image's color palette and the same color palette. However, the exception is white. White is switched to the 0th color of the color palette when the BMP is read. This is because the fontcvtr treats white as a transparent color and the 0th color is treated as a transparent color on a TWL and Nintendo DS system.

7 Vertical Fonts and Vertical-Style Fonts

7.1 Vertical Fonts

7.1.1 What are Vertical Fonts?

Vertical fonts are NITRO fonts that have been specially processed to perform vertical writing easily, with the character drawing functions of TWL-System G2D.

7.1.2 How to Create a Vertical Font

To create a vertical font, specify a vertical font in the input specification. If a vertical font is specified in the input specification, the font that is output will automatically be a vertical font. The various methods for specifying a vertical font in the input specification are as follows.

- Windows Fonts

Select an input font name that begins with “@”.

- LC Fonts

Enable the **Vertical** option in the input specifications for the LC font.

- BMP Images

Specify a vertical font that has been output as a BMP image.

- NITRO Fonts

Specify a vertical font for the NITRO font.

7.1.3 Output BMP Images of Vertical Fonts

If you specify a vertical font as input and a BMP image as output, a BMP image is output as shown in Figure 7-1. BMP images used as vertical fonts are formatted the same as normal BMP images, with each block rotated 90° clockwise.

Figure 7-1 BMP Output of a Vertical Font



In BMP blocks output by horizontal fonts, the character *width* is specified by the width line. In BMP blocks output by vertical fonts, the character *height* is specified by the width line. In addition, the vertical font baseline is oriented vertically.

If you specify a BMP image as input for vertical font output, vertical writing is output.

7.2 Vertical-Style Fonts

7.2.1 What are Vertical-Style Fonts?

The term *vertical-style* refers to holding the TWL or Nintendo DS such that the long edge of the screen is vertically oriented. The term *vertical-style drawing* refers to drawing strings so that they can be read correctly in vertical-style.

Vertical fonts are NITRO fonts that have been specially processed to perform vertical-style drawing easily with the character drawing functions of TWL-System G2D.

7.2.2 How to Create Vertical-Style Fonts

To create vertical-style fonts, specify a value other than **0** for the **Rotation** option when outputting a NITRO font. Values of **0**, **90**, **180**, or **270** can be selected, which respectively output fonts for when the DS is held while rotated counterclockwise by 0°, 90°, 180°, or 270°.

If **0** is selected for the **Rotation** option, a normal font is output.

7.3 Combinations of Vertical Writing and Vertical-Style

If a vertical font is specified as an input when outputting a NITRO font and the **Rotation** option is set to a value other than **0** at the same time, it is possible to create a vertical font for use with vertical-style. Accordingly, a total of eight variations exist, including the normal font. The variations depend on which of the four orientations is used and whether the font is vertical or horizontal.

8 Cautions

8.1 Converting Windows Fonts

Both Microsoft Windows 2000 and XP have automatic font linking, which automatically displays text characters in another font when the text characters are unsupported. For example, Tahoma font, which is a standard font in Windows 2000 and XP, does not support Japanese characters. But you can still display Japanese characters in Notepad even when Tahoma is set as the default font. This feature is useful for normal PC applications, but be careful about unlicensed fonts accidentally appearing in your game software.

This risk can be avoided because fontcvtr disables the automatic font linking feature and only outputs text characters included in the specified font. As a result, a font such as Tahoma that does not support Japanese will not output characters correctly.

8.2 Unicode Inside fontcvtr

fontcvtr uses Unicode for the internal processing of character codes and cannot handle text characters not supported by Unicode. Windows features are used to convert the various types of encoding. Therefore, the process must follow Windows conversion rules. For example, the JIS character code 0x8160 for a tilde (～) is defined in Unicode as U+301C. Windows conventions define a tilde as U+FF5E, so fontcvtr handles it as U+FF5E. Table 8-1 shows the character code conversions that fontcvtr generates for various inputs and outputs.

Table 8-1 Places Where Character Codes are Converted

Place	Description of Conversion Process
Input BMP	References the Letter Order file and allocates from cell position to Unicode.
Input NITRO	Converts from the encoding stored in NITRO font to Unicode.
Input Windows	Outputs characters in Unicode from U+0000 to U+FFFF.
Input LC	Converts from Shift_JIS to Unicode.
Output BMP	References the Letter Order file and allocates from Unicode to cell position.
Output NITRO	Converts from Unicode to specified encoding.
Character Filter file	Converts to Unicode when loaded.
Letter Order file	Converts to Unicode when loaded.

8.3 Warning Message About Duplicate Cells in a Letter Order File

Do not define the same character multiple times in a Letter Order file. If you create a Letter Order file that specifies the same text character multiple times and run the conversion process with BMP as the input source, only one glyph from all of the assigned cells is read. The glyphs in the rest of the cells are ignored. This situation can cause a bug that is difficult to pinpoint. Even rewriting the glyph in the ignored cell will not reflect the change in the font.

To help you find duplicate cells, fontcvtr displays a warning message if the same text character appears more than once in the Letter Order file and has a different glyph for each occurrence.

8.4 Unicode in the Letter Order File

Because fontcvtr uses Unicode for internal processing, the text characters in the Letter Order file are also converted internally with Unicode. If the conversion involves a non-Unicode character set, text characters that have been assigned different character codes are assigned a single Unicode character code. For example, each of the nine characters in Table 8-2 are assigned two different Shift_JIS character codes for use with Windows, but when they are converted into Unicode each is assigned just a single Unicode character code.

Even though all the character codes in the Letter Order file might appear to be different, sometimes the font will not reflect the changes made to the glyph in a cell. Because all occurrences of the same text character are grouped and assigned a single Unicode value, all the glyphs for that text character become the same glyph. You can avoid this by using single characters once.

fontcvtr displays a warning message when the same character is specified more than once in the Letter Order file, but the warning is displayed after the Unicode conversion. Thus, the warning is displayed even in the case described above.

Table 8-2 The 2:1 Relationship of Characters Converted from Shift_JIS to Unicode

Character	Character Code Normally Used by Shift_JIS	Shift_JIS Character Codes in Machine-Dependent Character Region	Unicode
U	0x81BE	0x879C	U+222A
∩	0x81BF	0x879B	U+2229
∠	0x81DA	0x8797	U+2220
⊥	0x81DB	0x8796	U+22A5
≡	0x81DF	0x8791	U+2261
≡	0x81E0	0x8790	U+2252
√	0x81E3	0x8795	U+221A
∴	0x81E6	0x879A	U+2235
∫	0x81E7	0x8792	U+222B

9 Warning and Error Messages

9.1 Warning Messages

Table 9-1 lists the warning messages displayed by fontcvtr. In addition to the messages in this table, fontcvtr displays an error message if it fails to read the Letter Order file at startup.

The warnings only indicate a possible problem, so fontcvtr will continue the conversion process.

Table 9-1 Explanation of Warning Messages

Warning Message	Explanation
A character ('{character}': U+{character code}) included in the Filter is not included in the INPUT font.	{Character} is specified for output in the character filter but is not present in the font at input. For this reason, the character output will not be included.
A character code ('{character}': U+{character code}) has multiple glyphs. Second and later glyphs are ignored.	{Character} corresponds to more than one glyph. All glyphs after the first are ignored. The cause is multiple definitions for the same character in the Letter Order file. For details, see section 8.3 Warning Message About Duplicate Cells in a Letter Order File.
BMP position ({coordinate x}, {coordinate y}) is <null/> in {Letter Order file's path}, but the cell includes non-white color.	The BMP cell with the specified coordinates is <null/> in the Letter Order file, but has a non-white color. The drawing of the glyph might protrude.
Can't represent output character ('{character}': U+{character code }) in the specified character set.	{Character} could not be represented with the specified encoding so it was not output. To output {Character} you need to use some other encoding.
More than one color detected in margin area at ({coordinate x}, {coordinate y}).	Two or more colors were detected in the margin of the cell with the coordinates in the BMP. The drawing of the glyph might protrude.

9.2 Error Messages

The error messages displayed by fontcvtr can be divided into five categories. Table 9-2 lists the error categories and Table 9-3 explains the error messages that are not part of the Internal Error category. In Table 9-3, the text enclosed in braces is replaced in the actual error messages.

When an error occurs, the conversion process stops and there is no output.

Table 9-2 Explanation of Error Categories

Error Category	Description
BMP Error	Error in a file specified as a BMP file.
NFTR Error	Error in a file specified as a NITRO font file.
XML Error	Error in a file specified as an XML file (Letter Order file, Character Filter file).
Parameter Error	Errors due to some other invalid input parameter.
Internal Error	These errors usually do not occur. They indicate that there might be a bug in fontcvtr.

Table 9-3 Descriptions of Error Messages

Error Message	Description
BMP Error: BMP has no baseline info.	The BMP does not contain baseline information. For details, see Chapter 6 BMP Image Format.
BMP Error: BMP height ({BMP height}) is not a multiple of vertical number of blocks ({Number of blocks in vertical direction})	The height of the BMP image is not a multiple of the number of vertical blocks specified in the Letter Order file. For details, see Chapter 6 BMP Image Format.
BMP Error: BMP height ({BMP height}) too small. Must be >= {Min. value of BMP in vertical direction}	The height of the BMP image is too small. For details, see Chapter 6 BMP Image Format.
BMP Error: BMP width ({BMP width}) is not a multiple of horizontal number of blocks ({Number of blocks in horizontal direction})	The width of the BMP image is not a multiple of the number of horizontal blocks specified in the Letter Order file. For details, see Chapter 6 BMP Image Format.
BMP Error: BMP width ({BMP width}) too small. Must be >= ({Min. value of BMP in horizontal direction})	The width of the BMP image is too small. For details, see Chapter 6 BMP Image Format.
BMP Error: Two or more width line found at ({Coordinate x}, {Coordinate y})	The width line of the cell in the mentioned coordinates is broken into two or more line segments. For details, see Chapter 6 BMP Image Format.
BMP Error: Unsupported BMP bpp ({BMP's bpp}).	The bpp (bits per pixel) value for the BMP image is not supported. fontcvtr supports only 1, 4, and 8 bpp.
BMP Error: Color white not found in BMP.	There is no white color in the BMP image. fontcvtr treats white as transparent color, so the BMP image must include the color white.
Cannot open a file ({File path}).	The file cannot be opened. Check to see that the displayed file path is correct.
Cannot read from a file ({File path}).	Cannot read the file. Some other application might have locked the file.
Cannot write to a file ({File path}).	The file cannot be written to. Some other application might have locked the file.
Input file ({File path}) has illegal image size({BMP data size} (expect {Expected BMP data size})).	The image data stored in the BMP file is an illegal size. The BMP file might be corrupt or the file might not be a BMP file.
Input file ({File path}) has illegal image size ({BMP width} x {BMP height}).	The image stored in the BMP file has an illegal width and height size. The BMP file might be corrupt or the file might not be a BMP file.

Error Message	Description
Input file ({File path}) has unsupported bpp ({BMP's bpp}).	The BMP file does not have a valid bpp (bits per pixel) value. The BMP file might be corrupt or the file might not be a BMP file.
Input file ({File path}) is not bmp file.	The file that has been specified as a BMP file is not a BMP file.
Input file ({File path}) is unsupported BMP.	The specified BMP file is in a format that fontcvtr cannot handle. fontcvtr does not support compressed BMP and BMP that has multiple planes.
NFTR File Error: Little-endian byte order support only.	The NITRO font file's byte order mark indicates big-endian. fontcvtr supports only little-endian order. The NITRO font file might be corrupt or the file might not be a NITRO font file.
NFTR File Error: The file is not NFTR file.	The file specified as a NITRO font file is not a NITRO font file.
NFTR File Error: Unknown header size (= {File's header size}, must be {Correct header size}).	The NITRO font file has an invalid header size. The NITRO font file might be corrupt or the file might not be a NITRO font file.
NFTR File Error: Unsupported version (= {File version}, supports {Supported version}).	This is an unsupported version of NITRO font.
NFTR File Error: Number of blocks is too small (= {Number of data blocks in file}).	The NITRO font file has an invalid number of blocks. The NITRO font file might be corrupt or the file might not be a NITRO font file.
NFTR File Error: Can't represent input character (0x{Character code}) in the Unicode character set.	The NITRO font includes a character that cannot be represented by Unicode. fontcvtr cannot handle characters that cannot be represented by Unicode. The problem might instead be that the character code table in the NITRO font is different from the current local character set.
Parameter Error: LC Font file path required.	An LC Font file has not been specified.
Parameter Error: Can't represent AlternateChar ('{Alternate character}': U+{Unicode character code of alternate character}) in the local character set. Please use Unicode character set or other AlternateChar.	The character specified as the alternate character cannot be represented by the local character code. Output the NITRO font in Unicode, or specify an alternate character.
Parameter Error: Can't represent AlternateChar (0x{Character code of alternate character}) in the Unicode character set. Please use other AlternateChar.	The character specified as the alternate character is not included in Unicode. fontcvtr cannot handle characters that cannot be represented by Unicode. Specify an alternate character.

Error Message	Description
Parameter Error: DefaultLeftSpace (= {Input value}) out of range [-128 ~ 127].	The default left-space width exceeds the range that can be specified. The value should be set between -128 and 127.
Parameter Error: DefaultWidth (= {Input value}) out of range [{Min. value} ~ {Max. value}].	The default character width exceeds the range that can be specified. The value should be set between {Min. value} and {Max. value}.
Parameter Error: DefaultRightSpace (= {Input value}) out of range [{Min. value} ~ {Max. value}].	The default right-space width exceeds the range that can be specified. The value should be set between {Min. value} and {Max. value}.
Parameter Error: Directory ({Directory path}) does not exist.	The path specified for the input file or for the output file does not exist.
Parameter Error: File not specified.	No file has been specified.
Parameter Error: File ({File path}) does not exist.	The specified file does not exist. Check that the path is correct.
Parameter Error: Invalid Font Size ({Specified font size}).	An invalid font size has been specified.
Parameter Error: Invalid alternate character (0x0000).	0x0000 cannot be specified for the alternate character.
Parameter Error: Invalid fixed width ({Specified character width}).	An invalid monospace character width has been specified.
Parameter Error: LineHeight (= {Specified value}) out of range [0 - 255].	An invalid line height has been specified. The value should be set between 0 and 255.
Parameter Error: Max glyph width (= {Glyph width}) over 255.	The glyph width exceeds 255 pixels. It must be no larger than 255 pixels.
Parameter Error: Max char width (= {Character width}) over 255.	The character width exceeds 255 pixels. It must be no larger than 255 pixels.
Parameter Error: Font height (= {Font height}) over 255.	The font height exceeds 255 pixels. It must be no larger than 255 pixels.
Parameter Error: Baseline position (= {Baseline position}) over 127.	The baseline position exceeds 127 pixels. It should be set between -128 and 127.
Parameter Error: Baseline position (= {Baseline position }) under -128.	The baseline position is set below -128 pixels. It should be set between -128 and 127.

Error Message	Description
Parameter Error: No glyphs in output.	There are no glyphs for output. Either there were no characters in the input source or all characters in the input source have been filtered out. Here are two possible reasons: (1) the system was configured to output only Japanese characters using a Western font, or (2) the characters in the Letter Order file are completely different from the characters in the character filter file.
Parameter Error: Specified AlternateChar ('{ Specified alternate character}' : {Character code (local)}) is not included in the Font.	The specified alternate character is not included in the font. The character specified as the alternate character must be part of the font being output.
Parameter Error: Specified AlternateChar ('{Specified alternate character}' : {Character code (Unicode)}) is not included in the Font.	The specified alternate character is not included in the font. The character specified as the alternate character must be part of the font being output.
Parameter Error: Specified file ({File path}) is read only.	The file specified for output has been configured as a read-only file, so data cannot be written to it.
Parameter Error: Specified path ({File path}) is a directory.	A directory was specified where a file should have been specified.
Parameter Error: Filter file ({File path}) does not exist.	The specified filter file does not exist. Check the path to see if it is correct.
Parameter Error: Input file ({File path}) is not LC Font file.	The file that was specified as a LC Font file is not an LC Font file.
Parameter Error: Cell Width (= {Specified value}) must be > 0.	The cell width must be specified as a value larger than 0.
Parameter Error: Cell Height (= {Specified value}) must be => 0.	The cell height must be specified as a value larger than 0
Parameter Error: Cell Margin Left (= {Specified value}) must be => 0.	The cell's left margin must be specified as a value of 0 or larger.
Parameter Error: Cell Margin Right (= {Specified value}) must be => 0.	The cell's right margin must be specified as a value of 0 or larger.
Parameter Error: Cell Margin Top (= {Specified value}) must be => 0.	The top margin of the cell must be specified as a value of 0 or larger.
Parameter Error: Cell Margin Bottom (= {Specified value}) must be => 0.	The bottom margin of the cell must be specified as a value of 0 or larger.
Parameter Error: Order file is not selected.	No Letter Order file has been selected.

Error Message	Description
Parameter Error: "Use Filter" checked, but filter file path not specified.	Use Filter has been checked, but no path has been entered for the Character Filter file.
Parameter Error: Output encoding is not selected.	No encoding has been selected for the NITRO font that will be output.
XML Error: Invalid filter file.	This is an invalid Character Filter file.
XML Error: Invalid order file.	This is an invalid Letter Order file.
XML Error: Invalid order height (= {The height specified in the Letter Order file}).	An invalid height has been specified by the Letter Order file.
XML Error: Invalid order width (= {The width specified in the Letter Order file}).	An invalid width has been specified by the Letter Order file.

10 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.

`TwlSystem/docs/Xerces-C++/NOTICE`

`TwlSystem/docs/Xerces-C++/LICENSE`

Microsoft and Windows are registered trademarks or trademarks of Microsoft Corporation in the U.S. and other countries.

LC Font is a registered trademark of Sharp Corp.

All other company and product names in this document are the registered trademarks or trademarks of those companies.

© 2004-2009 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.