

# fontcvtr マニュアル

## フォントコンバータの操作方法

2009-02-18

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、**厳重な取り扱い、管理を行ってください。**

# 目次

1	はじめに .....	6
1.1	fontcvtrについて .....	6
1.2	fontcvtrでできること .....	6
1.3	フォントのライセンス .....	7
2	ファイル構成 .....	7
3	操作説明 .....	8
3.1	GUI .....	8
3.2	変換 .....	9
3.2.1	基本操作 .....	9
3.2.2	フォントリソース作成の流れ .....	9
3.3	入力指定 .....	10
3.3.1	BMP .....	10
3.3.2	NITROフォント .....	10
3.3.3	Windowsフォント .....	11
3.3.4	LCフォント .....	12
3.4	出力指定 .....	14
3.4.1	BMP .....	14
3.4.2	NITROフォント .....	15
3.5	フィルタ指定 .....	16
4	文字フィルタファイル .....	17
4.1	概要 .....	17
4.2	構造 .....	17
	使用法 .....	18
4.3	DTD .....	18
5	文字順序ファイル .....	19
5.1	概要 .....	19
5.2	構造 .....	19
5.3	使用法 .....	20
5.4	DTD .....	21
5.5	付属文字順序ファイル .....	21
6	BMP画像フォーマット .....	22
6.1	BMP画像の構造 .....	22
6.1.1	ブロック .....	22
6.1.1.1	グリッドエリア .....	23
6.1.1.2	余白エリア .....	23
6.1.1.3	セル .....	23
6.1.1.4	幅線領域 .....	23

6.1.2	セルと幅線 .....	23
6.1.2.1	グリフイメージ .....	24
6.1.2.2	幅線 .....	24
6.1.2.3	出力されないグリフ .....	24
6.1.3	配置情報 .....	24
6.1.3.1	ベースライン .....	26
6.1.3.2	アセンダライン、ディセンダライン .....	26
6.1.4	縦横識別マーカー .....	28
6.2	出力されるBMPのフォーマット .....	28
6.3	入力可能なBMPのフォーマット .....	28
7	縦書き/縦持ちフォント .....	29
7.1	縦書きフォント .....	29
7.1.1	縦書きフォントとは .....	29
7.1.2	縦書きフォントの作り方 .....	29
7.1.3	縦書きフォントのBMP画像出力 .....	29
7.2	縦持ちフォント .....	30
7.2.1	縦持ちフォントとは .....	30
7.2.2	縦持ちフォントの作り方 .....	30
7.3	縦書きと縦持ちの組み合わせ .....	30
8	使用上の注意 .....	31
8.1	Windowsフォントから変換する際の注意 .....	31
8.2	fontcvtr内部でのUnicode .....	31
8.3	文字順序ファイル記述上の注意 .....	32
8.4	文字順序ファイルでのUnicode .....	32
9	警告/エラーメッセージ .....	33
9.1	警告メッセージ .....	33
9.2	エラーメッセージ .....	33
10	Xerces-C++について .....	39

## コード

コード 4-1	sample.xlft .....	17
コード 5-1	cp1252.xlft .....	19

## 表

表 4-1	文字フィルタ定義要素 .....	18
表 5-1	文字順序定義要素 .....	20
表 5-2	付属文字順序ファイル .....	21
表 6-1	配置情報点の解釈 .....	27
表 8-1	文字コードの変換が発生する箇所 .....	31

表 8-2	ShiftJISからUnicodeへ 2 対 1 で対応している文字 .....	32
表 9-1	警告メッセージ説明 .....	33
表 9-2	エラー種別説明 .....	34
表 9-3	エラーメッセージ説明 .....	34



図 1-1	フォント関連リソースの関係 .....	6
図 2-1	ファイル構成 .....	7
図 3-1	fontcvtrメインダイアログ .....	8
図 3-2	基本的なフォントリソース作成の流れ .....	9
図 3-3	入力: BMP .....	10
図 3-4	入力: NITROフォント .....	10
図 3-5	入力: Windowsフォント .....	11
図 3-6	入力: LCフォント .....	12
図 3-7	出力: BMP .....	14
図 3-8	セルサイズとグリフ出力位置の指定 .....	15
図 3-9	出力: NITROフォント .....	15
図 3-10	フィルタ指定 .....	16
図 6-1	BMP画像の例(ブロック数: 16x9) .....	22
図 6-2	ブロック模式図 .....	22
図 6-3	セルと幅線模式図 .....	23
図 6-4	セルの中での左右位置 .....	24
図 6-5	すべての配置情報ピクセル .....	25
図 6-6	ベースライン位置模式図 .....	26
図 6-7	アセンダライン、ディセンダラインの位置模式図 .....	26
図 6-8	フォントの高さ .....	27
図 6-9	縦横識別マーカー .....	28
図 7-1	縦書きフォントのBMP出力 .....	29

## 改訂履歴

改訂日	改訂内容
2009-02-18	エラーメッセージの記述を修正。
2008-08-05	6.1.3配置情報を追加。
2008-05-30	NITRO-System の名称変更による修正(NITRO-System を TWL-Sysytem に変更)。
2008-04-08	改訂履歴の書式を変更。 Xerces-C++のライセンス表記を変更。 全体の校正。
2007-11-26	Ver.1.1.3 に対応。
2007-04-25	図表番号のずれを修正。 Windows フォントを入力に用いる際のフォントサイズ指定について追記。
2007-03-14	Ver.1.1.0 に対応。
2005-09-08	誤記修正。
2005-07-11	誤記修正。
2005-05-25	初版。

# 1 はじめに

## 1.1 fontcvtrについて

fontcvtr は、TWL およびニンテンドーDS で使用することができるフォント形式である NITRO フォントの作成を目的とする Windows 用ツールです。

TWL-System G2D の文字描画ライブラリと用いる事により、TWL およびニンテンドーDS 上で NITRO フォントを容易に文字列描画することができます。

## 1.2 fontcvtrでできること

fontcvtr は基本となる機能として Windows 用フォントまたはシャープ株式会社製 LC フォントを NITRO フォントに変換する機能を持ちます。また、これらのフォントを BMP 形式の画像ファイルとして書き出すこともできます。書き出した BMP ファイルを読み込んで NITRO フォントに変換するという事ができますので、元となるフォントを一旦 BMP に変換し微調整した上で NITRO フォントに変換するという使い方が可能です。さらに、BMP 形式でグリフデータを作成する事で一からオリジナルのフォントを作成する事もできます。

図 1-1 にこれらの関係を図示します。

各変換に際しては元となるフォントから必要な文字だけを取り出して、必要最小限のフォントを作ることができます。

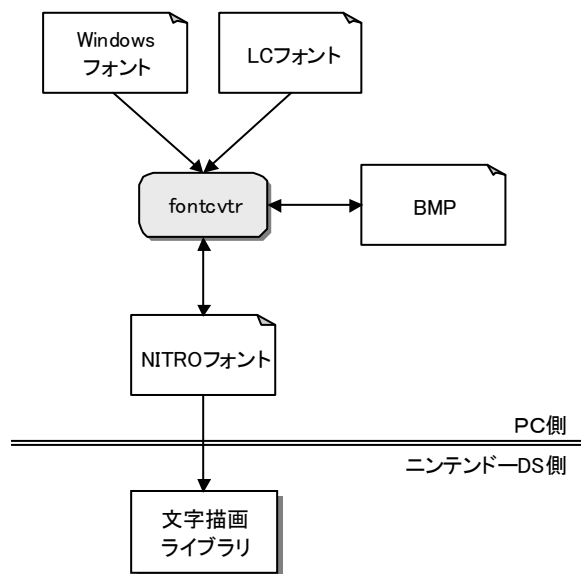


図 1-1 フォント関連リソースの関係

## 1.3 フォントのライセンス

fontcvtr は PC にインストールされている任意のフォントを TWL およびニンテンドーDS で利用できるようにしますが、これらのフォントを利用したソフトを発売するには利用するフォントのライセンスが必要になります。各ゲームソフトごとに適切なライセンスを取得してください。

fontcvtr および TWL-System には LC フォントを含め、いかなるフォントのライセンスも付属しません。

## 2 ファイル構成

fontcvtr とその関連ファイルは TwlSystem/tools/win/fontcvtr 以下に配置されています。またマニュアルは TwlSystem/docs/G2D 以下に用意されています。これらのフォルダに含まれる fontcvtr 関連のファイル構成を図 2-1 に示します。

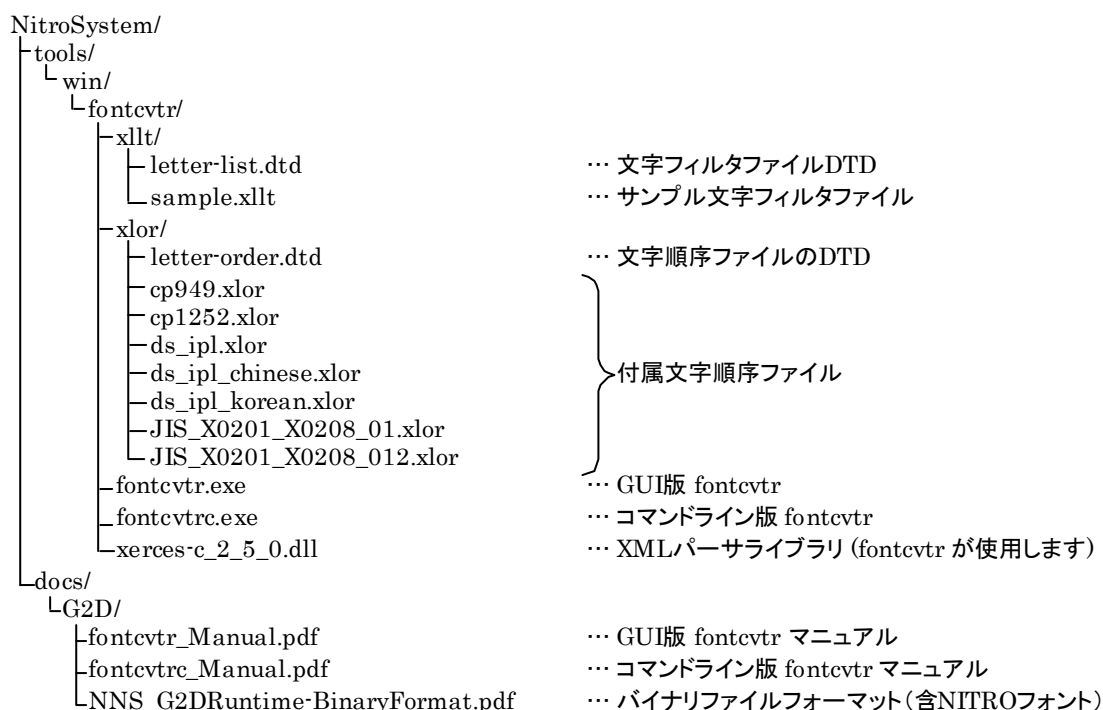


図 2-1 ファイル構成

文字フィルタファイルについては「4 文字フィルタファイル」を、文字順序ファイルについては「5 文字順序ファイル」を参照してください。GUI版fontcvtrマニュアル、すなわち、今お読みのこのマニュアルではGUI版とコマンドライン版のfontcvtrに共通の事項及びGUI版のfontcvtrに固有の事項を扱っています。コマンドライン版fontcvtrに固有の事項についてはコマンドライン版fontcvtrマニュアルを参照してください。

図 2-1には示されていませんがGUI版fontcvtrを起動するとfontcvtr.exeが存在するディレクトリにfontcvtr.iniという設定ファイルを作成します。この設定ファイルにはGUI版fontcvtrの表示状態が自動的に記録されていき、次回起動時に終了時の状態を復元するために使われます。

## 3 操作説明

### 3.1 GUI

図 3-1はGUI版fontcvtr（以下fontcvtr）のスクリーンショットです。fontcvtrはダイアログベースのGUIアプリケーションで、全ての操作をこの図に示されるダイアログ上で行います。

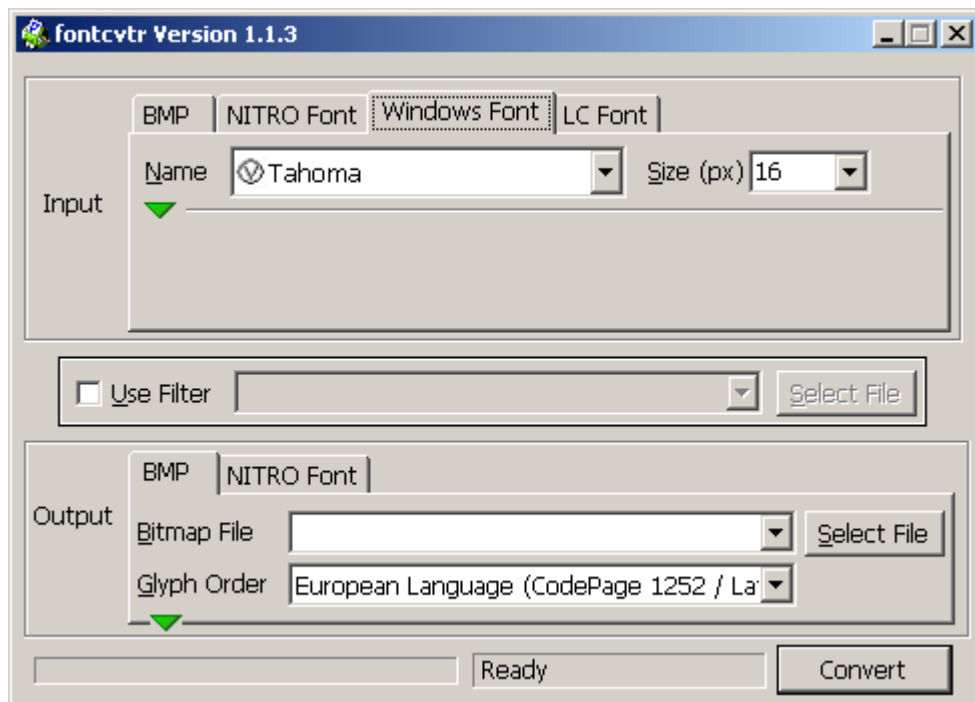


図 3-1 fontcvtr メインダイアログ

メインダイアログは大きく 4 つの部分に分かれます。上半分をしめる「入力指定」、下半分をしめる「出力指定」、そして中央にある「フィルタ指定」と最下部の「状態表示」です。「入力指定」で変換元となるリソースとそのオプションを指定し、「出力指定」で変換先のリソースを指定します。あとは「Convert」ボタンを押すことで変換が行われます。

最下部の「状態表示」ではプログレスバーと処理ステータス表示で現在の処理状況を表示します。

fontcvtr の操作全体に共通してファイルを指定すべき欄では Drag and Drop でファイルを指定することが出来ます。



## 3.2 変換

### 3.2.1 基本操作

fontcvtr で変換を行うための基本的な操作手順は以下のようになります。

- (1) 変換元となるリソースのタブを選択し、そのタブで入力リソースの設定を行います。詳しくは「3.3 入力指定」を参照してください。
- (2) 出力先となるリソースのタブを選択し、そのタブで出力リソースの設定を行います。詳しくは「3.4 出力指定」を参照してください。
- (3) 必要に応じてフィルタを指定します。フィルタを利用しない場合は「Use Filter」のチェックをはずしておきます。詳しくは「3.5 フィルタ指定」を参照してください。
- (4) 「Convert」ボタンを押します。
- (5) 変換が実行されます。入力リソースや出力リソースおよびそれらの設定に異常があるとメッセージウインドウに Warning として表示されます。fontcvtr で補正不可能な異常があると Error として表示され変換が失敗します。Warning のみや何も表示されなければ変換は成功です。警告/エラーメッセージについては「9 警告/エラーメッセージ」を参照してください。

### 3.2.2 フォントリソース作成の流れ

まず「Windows フォント→BMP」または「LC Font→BMP」の変換でベースとなるフォントを BMP へ書き出します。これを必要に応じて修正し、「BMP→NITRO フォント」の変換で NITRO フォントへと変換します。この時にフィルタを通してアプリケーションに必要な文字のみを NITRO フォントへと格納します。

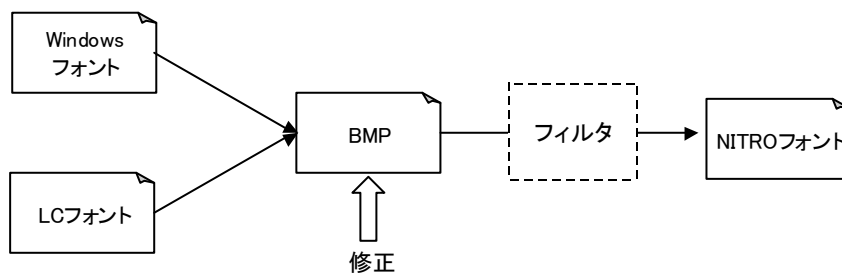


図 3-2 基本的なフォントリソース作成の流れ

## 3.3 入力指定

### 3.3.1 BMP

図 3-3は入力指定のBMPタブです。「6.BMP画像フォーマット」に従ったBMP画像を入力元として指定することができます。

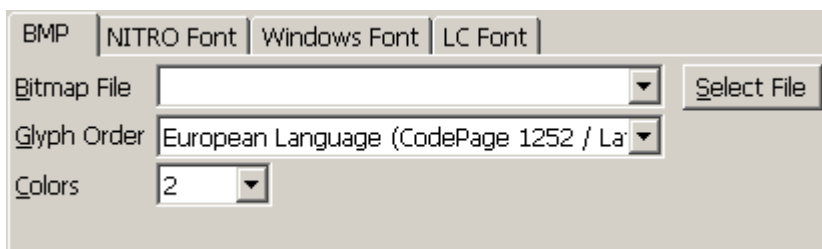


図 3-3 入力:BMP

- **Bitmap File**  
変換元の BMP ファイルを指定します。
- **Glyph Order**  
BMP中にどのような順番で文字が並んでいるかを選択します。詳しくは「5 文字順序ファイル」を参照してください。
- **Colors**  
BMP 中で有効な色数を指定します。これは出力されるフォントの色数の指定でもあります。  
ここで指定する数以上の色が使われている場合、それらの色は抜き色として扱われます。例えば Colors に 4 を指定する場合、BMP カラーパレット中の 0~3 番の色が出力され、4 番以降の色は抜き色として扱われます。

### 3.3.2 NITROフォント

図 3-4は入力指定のNITROフォントタブです。fontcvtrによって作成されたNITROフォントを入力として指定することができます。

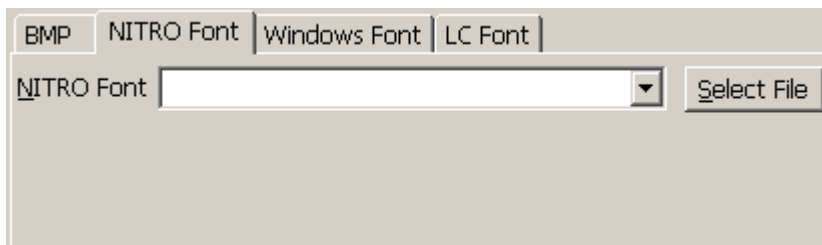


図 3-4 入力:NITRO フォント

- **NITRO Font**  
変換元の NITRO フォントを指定します。

### 3.3.3 Windowsフォント

図 3-5は入力指定のWindowsフォントタブです。Windowsにインストールされているフォントを入力元として指定します。

Windows フォントを入力元とすると Unicode(UCS2)の全範囲の中でフォントに存在する文字が出力側に渡されます。出力側が BMP の場合は文字順序ファイルにより出力文字が制限されますが、NITRO フォントの場合は文字フィルタを指定しないとフォントに存在する全ての文字が出力されることになります。フォントによっては数万文字が出力されますので注意が必要です。

Windows フォントタブでは初期状態で「Name」と「Size」の項目のみが表示されています。緑色の三角アイコンをクリックする事で、それ以外の項目の表示／非表示を切り替える事が出来ます。

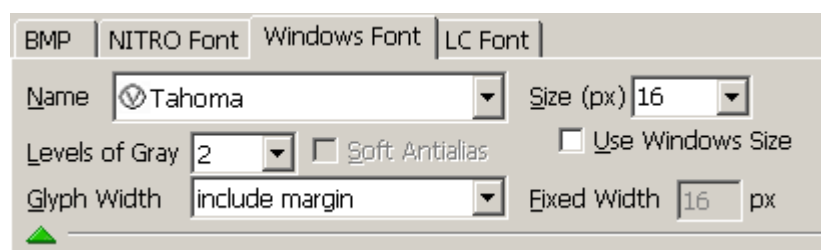


図 3-5 入力:Windows フォント

- **Name**

変換元とするフォントを指定します。一覧には PC にインストールされているフォントが表示されます。背景色がツールチップ色のはラスタフォント(ビットマップフォント)で、白色のものはベクタフォント(アウトラインフォント)です。

フォント名の先頭に@がつくものは縦書き用のフォントです。縦書き用のフォントを入力とすると出力では縦書き用の出力を行うようになります。

- **Size**

変換元フォントのサイズをピクセル単位で指定します。フォントによっては厳密にこの値にならないものがあります。特にラスタフォントに対して小さいサイズを指定した場合に顕著です。

- **Use Windows Size**

チェックをつけると「Size」で指定されるフォントサイズを一般的な Windows ソフトウェアと同様に解釈するようにします。これにより fontcvtr の出力を他の Windows ソフトウェアの出力と組み合わせる場合に出力サイズをあわせることができます。ただし、一般的な Windows ソフトウェアの多くはフォントサイズをポイント単位で扱っているのに対し、「Size」はピクセル単位であることに注意して下さい。

チェックをつけていない状態では fontcvtr はアクセント記号付き文字のような背の高い文字が指定されたサイズとなるように出力するフォントサイズを調整します。これに対して一般的な Windows ソフトウェアでは通常のアルファベットの高さが指定されたサイズとなるようにフォントサイズを調整します。このため一般的にはチェックをつける場合はつけない場合に比べて出力される文字の大きさは大きくなります。

- **Levels of Gray**

出力する階調数を指定します。たとえば「2」だと白と黒の 2 色のみになります。ラスタフォントでは「2」に固定され選択できません。

- **Soft Antialias**

多階調出力(2 階調以外の出力)を行うと出力されるグリフにアンチエイリアス処理が施されますが、このアンチエ

イリアスをよりやわらかくします。この処理によってフォントによっては出力されるグリフの品質が改善することがあります。

- **Glyph Width**

Windows フォントの各文字が持つ左右の空白幅を、出力する文字幅に含めるかどうかを指定します。

「glyph only」であれば左右の空白を含めません。文字幅とグリフ幅が一致します。

「glyph only (keep space)」は基本的に「glyph only」と同様ですが、空白文字については「include margin」を指定した場合と同じ文字幅になります。

「include margin」であれば左右の空白を含めます。

「fixed width」では全ての文字で次項の「Fixed Width」で指定した文字幅になるようにします。

「glyph only」で出力する場合、空白文字に関してはグリフがないために文字幅が 0 として出力され、そのまま使用すると空白文字が存在しないかのように表示されてしまいます。空白文字を使用する場合は、代わりに「glyph only (keep space)」を指定するか、一旦画像ファイルに出力のうえ、空白文字の文字幅を直接調整する必要があります。

- **Fixed Width**

文字幅をピクセル単位で指定します。「Glyph Width」で「fixed width」を選択した場合のみ指定できます。

### 3.3.4 LCフォント

図 3-6は入力指定のLCフォントタブです。シャープ株式会社製LCフォントを入力元として指定します。

LC フォントの入手・利用については弊社業務部までお問い合わせください。

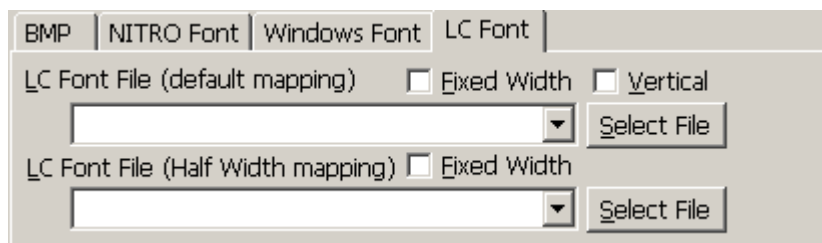


図 3-6 入力:LC フォント

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

変換元の LC フォントファイルを指定します。

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

変換元の LC フォントファイルを指定します。default mapping との違いは半角の文字コードで出力する点です。LC フォントには全角の文字コードで文字が格納されていますが、これを半角の文字コードに置き換えて出力に渡します。これにより ASCII 文字として LC フォントのアルファベットや数字を利用する事ができるようになります。

default mapping と Half Width mapping を同時に指定した場合は 2 つのフォントを 1 つにまとめて変換します。default mapping に全角の LC フォントを、Half Width mapping に半角の LC フォントを指定することで ASCII から第二水準の漢字まで(文字順序ファイルの JIS\_X0201\_X0208\_012.xlor 相当)をカバーできるようになります。

- **Fixed Width**

LC フォントは全ての文字で文字幅が等しい等幅フォントですが、**Fixed Width** のチェックをはずすと、変換時にグリフの左右のスペースを取り除きプロポーションアルフォントとして出力します。これは Windows フォントの入力指定において **Glyph Width** に **glyph only** を選択したのと同様の効果です。ただし LC フォントでは文字の密着を防ぐため全ての文字に 1 ピクセルの左スペースがつきます。

- **Vertical**

LCフォントを縦書き用のフォントとして扱うようにします。詳しくは「7.1 縦書きフォント」を参照してください。

## 3.4 出力指定

出力指定の各タブでは初期状態では最低限指定しなければならない項目のみが表示されています。緑色の三角アイコンをクリックする事で、それ以外の項目の表示／非表示を切り替える事が出来ます。

### 3.4.1 BMP

図 3-7は出力指定のBMPタブです。入力元のフォントデータを「6.BMP画像フォーマット」に従ったBMP画像として出力します。ここで出力したBMP画像はそのまま入力指定のBMPで使う事ができます。

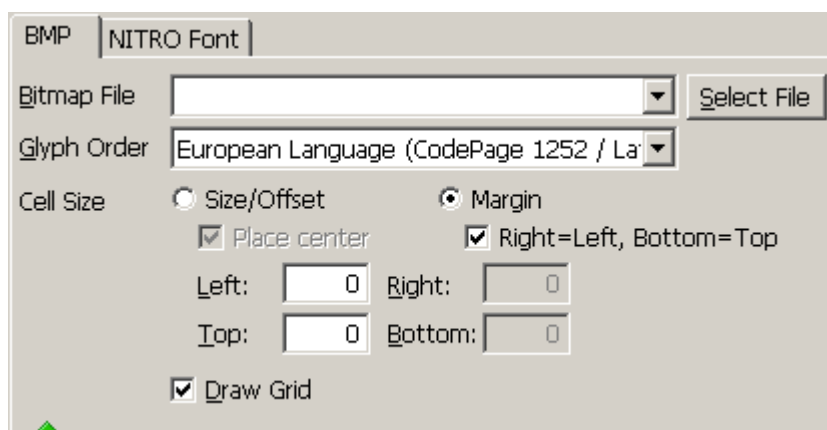


図 3-7 出力:BMP

- **Bitmap File**  
変換先の BMP ファイルを指定します。
- **Glyph Order**  
BMPにどのような並びで文字を出力するかを選択します。詳しくは「5 文字順序ファイル」を参照してください。
- **Size/Offset と Margin**  
「Size/Offset」の場合は図 3-8左のようなパラメータでセルの大きさとグリフ出力位置を指定します。「Margin」の場合は図 3-8右のようなパラメータでセルの大きさとグリフ出力位置を指定します。
- **Place center**  
チェックをつけるとグリフがセルの中心にくるように「Left」、「Top」の値を自動設定します。「Size/Offset」を選択している場合にチェックを付ける事ができます。
- **Right=Left, Bottom=Top**  
チェックをつけると「Right」を「Left」と、「Bottom」を「Top」と同じ値にします。「Margin」を選択している場合にチェックを付ける事ができます
- **Left, Top, Right, Bottom**  
「Margin」が選択されている場合は図 3-8右の各パラメータを指定します。  
「Size/Offset」が選択されているとそれぞれが「Width」、「Height」、「Left」、「Top」に変化し、図 3-8左の各パラメータを指定します。
- **Draw Grid**  
チェックをつけるとグリッドを描画します。

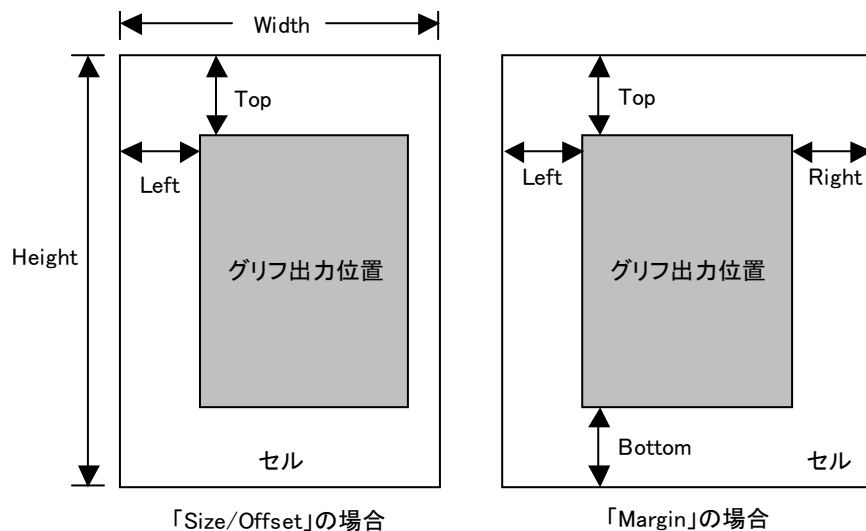


図 3-8 セルサイズとグリフ出力位置の指定

### 3.4.2 NITROフォント

図 3-9 は出力指定のNITROフォントタブです。入力元のフォントデータを文字描画ライブラリが利用するフォントソースへと変換します。最終的にはこの形式で出力する事になります。

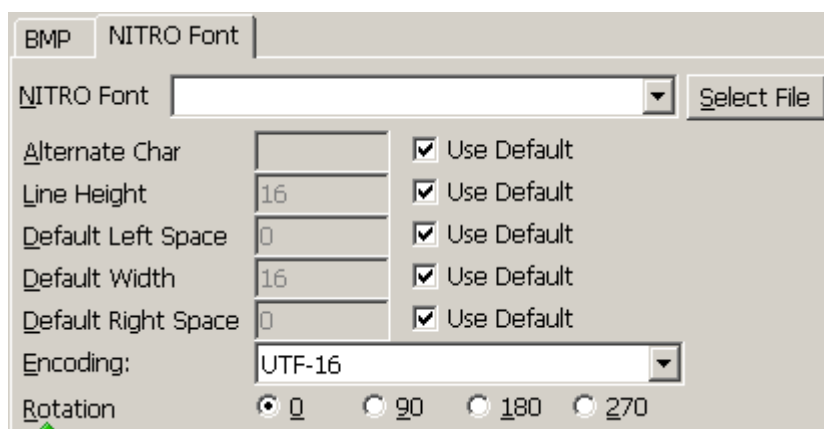


図 3-9 出力:NITRO フォント

- **NITRO Font**

出力先の NITRO フォントファイルを指定します。

- **Alternate Char**

代替文字を指定します。文字描画ライブラリではフォントにない文字の代わりとして、ここで指定した文字を表示します。文字の指定は文字コードを指定するか、もしくは直接文字を入力して指定します。入力が 1 文字だけであれば直接文字が指定されていると判断します。文字コードで指定する場合は C 言語の指定法で 8,10,16 進数が利用できます。「Use Default」時は最も文字コードが小さい文字(多くの場合は半角スペース)が指定されたものとみなします。

- **Line Height**  
1 行の高さを指定します。文字列描画ライブラリでは基本の行の高さとしてこの値が使われます。有効な値は 0 ～255 です。「Use Default」時はフォントの高さと同じ値が指定されたものとみなします。
- **Default Left Space**  
デフォルトの左スペース幅を指定します。有効な値は-128～127 です。「Use Default」時は 0 が指定されたものとみなします。デフォルトの左スペース幅は文字固有の左スペース幅を持たない文字が使用します。
- **Default Width**  
デフォルトのグリフ幅を指定します。有効な値は 0～最大グリフ幅です。「Use Default」時はフォント中の最大グリフ幅が指定されたものとみなします。デフォルトのグリフ幅は文字固有のグリフ幅を持たない文字が使用します。
- **Default Right Space**  
デフォルトの右スペース幅を指定します。「Use Default」時は 0 が指定されたものとみなします。デフォルトの右スペース幅は文字固有の右スペース幅を持たない文字が使用します。
- **Use Default**  
それぞれの入力欄の右にある「Use Default」にチェックをつけると対応するパラメータにデフォルト値が使われます。実際に使われる値については上記の各項目の説明を参照してください。ただし、入力側が NITRO フォントの場合は NITRO フォント中の対応する設定値がデフォルトとして使われます。
- **Encoding**  
出力するフォントが対応する文字エンコーディング形式を指定します。
- **Rotation**  
縦持ちフォントを作成する場合の回転量を指定します。それぞれ度単位(度数法)での反時計回りの回転角度を表します。「0」を選択すると通常のフォントを作成します。詳しくは「7.2 縦持ちフォント」を参照してください。

## 3.5 フィルタ指定

図 3-10 はフィルタ指定欄を抜き出したものです。フィルタ指定で文字フィルタファイルを指定することで文字フィルタファイルで定義されている文字以外の文字を出力しないようにします。これにより必要な文字のみをもったコンパクトな NITRO フォントを作成する事ができます。文字フィルタを使用しない場合は入力側が持つ全ての文字を出力側に渡します。

文字フィルタファイルについては「4 文字フィルタファイル」を参照してください。



図 3-10 フィルタ指定

- **Use Filter**  
チェックをつけると変換時にフィルタを適用します。



## 4 文字フィルタファイル

### 4.1 概要

文字フィルタファイルは fontcvtr が出力する文字を指定するためのファイルです。文字フィルタファイルを使うことでアプリケーションが必要とする文字のみを NITRO フォントに格納する事ができます。

文字フィルタファイルは XML で記述されており拡張子.xlft をもちます。

### 4.2 構造

文字フィルタファイルの XML 構造は TWL-System XML ファイル規約 (以下 NNSXML 規約) に従っています。ここでは NNSXML 規約では規定されない、文字フィルタファイルのルート要素と body 要素内の構造について説明します。TWL-System XML ファイル規約については TwlSystem/docs/Readme/DataFormatRule.pdf を参照してください。

例として文字フィルタファイルのサンプル sample.xlft の内容をコード 4-1 に示します。この文字フィルタファイルを使用すると「fontcvrあいうえかきくけ任天堂」の 18 文字と半角スペースの計 19 文字のみが出力されるようになります。

```
<?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>xlftサンプル</title>
    <comment>文字フィルタファイルのサンプルです</comment>
  </head>

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

#### コード 4-1 sample.xlft

文字フィルタのルート要素は letter-list であり、NNSXML 規約で規定される head 要素と body 要素のみを包含します。

head 要素とその包含要素は全て NNSXML 規約で規定されているためここでは説明を省略します。

body 要素は letter 要素のみを持ち、letter 要素の内容に含まれる文字が出力される文字となります。ただし letter 要素の内容のうち、空白文字 (半角スペースやタブ文字) は無視され、出力される文字には常に半角スペースが含まれるものとして扱います。

以上を表 4-1 にまとめます。

表 4-1 文字フィルタ定義要素

要素名	包含可能要素	説明
	属性	
letter-list	head, body	文字フィルタを定義します。文字フィルタファイルのルート要素です。
	version (必須)	必須属性である version には現在は 1.0 を指定します。
letter	なし	NITRO フォントに格納する文字を定義します。文字の指定は格納したい文字を直接記す事で行います。
	なし	

## 使用法

文字フィルタファイルの使用法については「3.5 フィルタ指定」を参照してください。

### 4.3 DTD

DTD (Document Type Definition) とはXML文書の構造定義を規則に従って記述したもので、記述される内容は表 4-1と同内容です。

文字フィルタファイルの DTD は fontcvtr.exe と同じフォルダの xslt フォルダ内にある letter-list.dtd に記述されています。

## 5 文字順序ファイル

### 5.1 概要

文字順序ファイルは fontcvtr が入出力する BMP ファイル中の文字の並びを定義します。具体的には BMP 画像内での横方向文字数、縦方向文字数、そして文字の描画順序を定義します。文字順序ファイル内で位置が定義されていない文字は出力されないため、文字順序ファイルは文字フィルタとしても働きます。

文字順序ファイルは XML で記述されており拡張子.xlor をもちます。

### 5.2 構造

文字順序ファイルの XML 構造も文字フィルタファイルと同様に NNSXML 規約に従っています。

付属のcp1252.xlorの内容(一部省略)をコード 5-1に示します。

```
<?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 ~(略)~ 8859-1 (Latin-1).</comment>
  </head>
  <body>
    <area width="16" />

    <order>
      <sp/> ! &quot; # $ % & amp; ' ( ) * + , - . /
      0 1 2 3 4 5 6 7 8 9 : ; &lt; = &gt; ?
      @ 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/> ~(略)~ &#x00FE; &#x00FF;
    </order>
  </body>
</letter-order>
```

#### コード 5-1 cp1252.xlor

文字順序定義のルート要素は letter-order であり、letter-order 要素は NNSXML 規約で規定される head 要素と body 要素のみを包含します。

head 要素とその包含要素は全て NNSXML 規約で規定されているためここでは説明を省略します。

body 要素は area 要素と order 要素のみを持ち、area 要素で BMP 画像内での縦横方向文字数を、order 要素で文字の描画順序を指定します。

**area** 要素は内容を持たず縦横方向の文字数を定義する **height** と **width** の 2 つの属性を持ちます。2 つの属性はともに省略可能で **width** を省略した場合は 16 が指定されたものとみなし、**height** を省略した場合には全ての文字を出力するのに必要十分な値が指定されたものとみなします。

**order** 要素はその内容として文字の羅列をもち、この文字の羅列された順序が BMP 画像中での文字の順序となります。BMP 画像中では左上の文字が 1 番目となり、その右隣の文字が 2 番目の文字、以後右端へ向かって順に並びます。BMP 画像の右端まできたら 2 行目に移り、同様に左端から右端へと並びます。**order** 要素では空白文字(半角スペースやタブ文字)は無視されるため、半角スペースを出力する場合には代わりに **sp** 要素を用いる必要があります。また **null** 要素を用いる事で出力位置を 1 つ飛ばす事が出来ます。

以上を表 5-1にまとめます。

**表 5-1 文字順序定義要素**

要素名	包含可能要素	説明
	属性	
letter-order	head, body	文字順序を定義します。文字順序ファイルのルート要素です。
	version (必須)	必須属性である version には現在は 1.0 を指定します。
area	なし	BMP 中の縦横の文字出力数を定義します。属性 width は横方向の文字数を定義します。省略時は 16 が指定されたものとして扱います。属性 height は縦方向の文字数を定義します。省略された場合は order 要素で指定された文字を出力するのに必要十分な値が指定されたものとして扱います。
	width (省略可能)	
	height (省略可能)	
order	sp, null	出力する文字とその順序を定義します。文字の指定は出力する順序に文字を並べる事によって行います。同じ文字が 2 度以上現れてはいけません。
	なし	
sp	なし	order 要素内で半角スペースを指定します。半角スペースは直接記述しても無視されるため、この要素を用います。
	なし	
null	なし	文字を出力しない事を指定します。すなわち 1 文字分出力位置を飛ばします。
	なし	

## 5.3 使用法

fontcvtr は起動時に fontcvtr.exe と同じフォルダにある xlor フォルダ内で拡張子が xlor のファイル(xlor¥\*.xlor)を文字順序ファイルとして読み込み、その title 要素の内容を Glyph Order の項目に選択肢として表示します。このため文字順序ファイルを使用するには fontcvtr.exe と同じフォルダにある xlor フォルダ内に拡張子 xlor をつけて文字順序ファイルを保存しなければなりません。

文字順序ファイルを xlor フォルダに追加した後で fontcvtr を起動すると新しく追加した文字順序ファイルも Glyph Order の項目に表示されるようになります。この追加された項目を BMP ファイルの入出力時に選択する事で追加した文字順序ファイルが BMP ファイルの入出力に用いられるようになります。

文法違反などの理由で起動時に読み込みに失敗した文字順序ファイルは Glyph Order の項目に表示されず利用する事が出来ません。この場合、警告メッセージが表示されますので、メッセージに従って文字順序ファイルを修正してください。

## 5.4 DTD

DTD は fontcvtr.exe と同じフォルダの xlor フォルダ内にある letter-order.dtd に記述されています。

## 5.5 付属文字順序ファイル

fontcvtrには 7 つの文字順序ファイルが付属しています(表 5-2エラー! 参照元が見つかりません。)。これら付属の文字順序ファイルは出力する文字以外の仕様は共通であり、文字コード順に横 16 文字出力となっています。

文字順序ファイルでは文字コードを指定するだけであるため、フォントごとの文字の形の違いを吸収できません。例えば入力として Windows フォントで日本語フォントを選択し、出力を BMP で文字順序ファイルを cp1252.xlor とした場合、フォントによってはバックスラッシュが出力されるべき場所に円記号(¥)が出力されます。

表 5-2 付属文字順序ファイル

ファイル名	fontcvtr での表示
説明	
cp1252.xlor	European Language (CodePage 1252 / Latin-1)
英数字と欧州文字の一部を出力します。これは Windows コードページ 1252 と呼ばれるもので、ISO 8859-1 (Latin-1)及び ASCII の全ての文字を含んでいます。	
ds_ipl.xlor	DS-IPL font table
ニンテンドーDS の IPL フォントと同等の文字を出力します。cp1252.xlor と JIS_X0201_X0208_01.xlor に加えて、絵文字用に Unicode の外字用領域をあわせたものとなっています。ただし cp1252.xlor と JIS_X0201_X0208_01.xlor では同じ文字コードの文字が含まれるため JIS_X0201_X0208_01.xlor の側の該当文字を<null/>に置き換えています。また欧州文字と日本語文字が含まれるため NITRO フォントとして出力する時には Encoding を UTF-16 または UTF-8 にする必要があります。	
ds_ipl_chinese.xlor	DS-IPL font table for Chinese
中国向けニンテンドーDS の IPL フォントと同等の文字を出力します。ds_ipl.xlor の日本語漢字部分を中国語漢字 (GB2312)に置き換えたものです。	
ds_ipl_korean.xlor	DS-IPL font table for Korean
韓国向けニンテンドーDS の IPL フォントと同等の文字を出力します。ds_ipl.xlor の日本語漢字部分をハングル文字 (KS X 1001)に置き換えたものです。	
JIS_X0201_X0208_01.xlor	Japanese Level 1 (JIS X 0208)
第一水準漢字までの日本語文字を出力します。これには半角英数字及び半角カナが含まれます。	
JIS_X0201_X0208_012.xlor	Japanese Level 1,2 (JIS X 0208)
JIS_X0201_X0208_01.xlor に加え第二水準漢字を出力します。	
cp949.xlor	Korean (CodePage 949 / UHC)
英数字とハングル、漢字を出力します。これは Windows コードページ 949 または UHC(Unified Hangul Code)と呼ばれるものです。	

## 6 BMP画像フォーマット

### 6.1 BMP画像の構造

fontcvtr が扱う BMP 画像はブロック及びセルと呼ぶ 2 つの構造をもっています。BMP 画像は格子状に隙間なく並んだブロックで構成され、個々のブロックの中にセルがあります。

#### 6.1.1 ブロック

BMP画像中には文字順序ファイルで指定される幅×高さの個数のブロックが隙間なく敷き詰められています。(図 6-1)

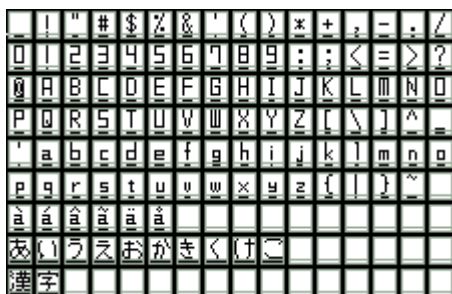


図 6-1 BMP 画像の例(ブロック数: 16x9)

1 つのブロックの中にはセルと幅線が 1 つずつ含まれます。図 6-2は幅 14 ピクセル高さ 16 ピクセルのブロックの模式図です。太い実線がブロックの境界を、細い破線がピクセルの境界を表します。

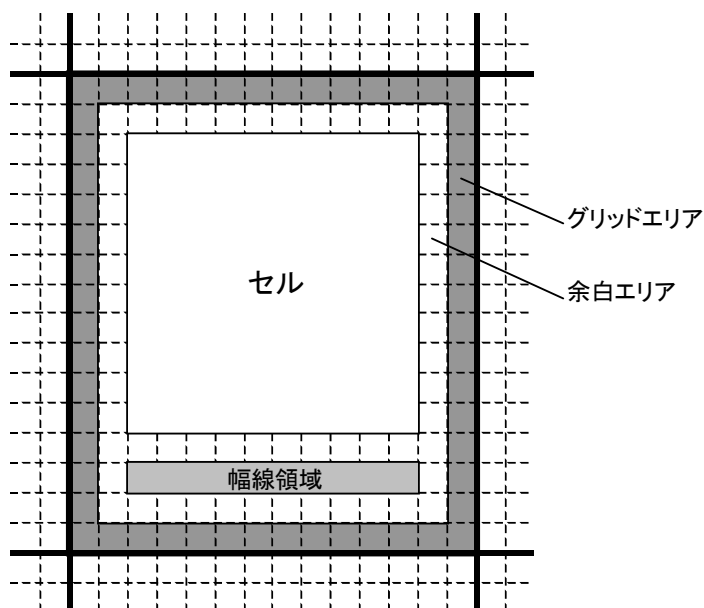


図 6-2 ブロック模式図

### 6.1.1.1 グリッドエリア

ブロック境界の内側 1 ピクセルの部分はグリッドエリアと呼び、図ではブロック境界に沿うように濃い灰色に塗られた部分です。

BMP 出力時に **Draw Grid** をチェックすると、グリッドエリアは黒で塗りつぶされて隣り合うブロックのグリッドエリアとあわせた幅 2 ピクセルのグリッドがひかれることになります。**Draw Grid** をチェックしなかった場合は白で塗りつぶされます。

BMP 入力時にはこのエリアは単に無視されます。

### 6.1.1.2 余白エリア

グリッドエリアのさらに内側 1 ピクセルの領域は余白エリアです。このエリアはグリッドとセルや幅線がくっついてしまわないように設けられています。またセルと幅線の間にも 1 ピクセルの余白エリアが存在します。

BMP 出力時に **Draw Grid** をチェックすると、余白エリアは鈍い緑で塗りつぶされます。**Draw Grid** をチェックしなかった場合は白で塗りつぶされます。

BMP 入力時にはこのエリアが単一色で塗りつぶされている事をチェックします。2 色上の色が検出された場合は警告が表示されます。

### 6.1.1.3 セル

余白エリアの内側上部はセルと呼びます。セルはグリフイメージの描画エリアであり、この領域内にグリフイメージが描画されます。幅線の高さが 1 ピクセル固定であるため、

セルの大きさは常に

セルの幅 = ブロックの幅 - 4 ピクセル  
セルの高さ = ブロックの高さ - 6 ピクセル

となります。

### 6.1.1.4 幅線領域

余白エリアの内側下部は幅線と呼ぶ太さ 1 ピクセルの線が引かれる領域です。幅線は文字幅と、グリフイメージの相対的な位置を定義します。

## 6.1.2 セルと幅線

図 6-3(左)に幅 10 ピクセル高さ 10 ピクセルのセルと幅線領域の模式図を示します。

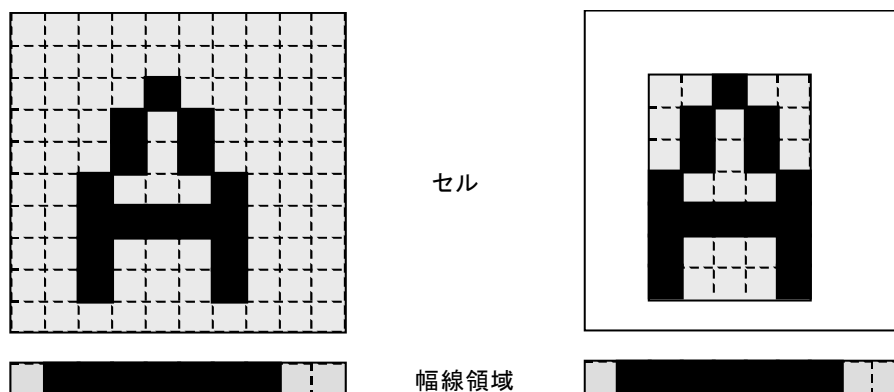


図 6-3 セルと幅線模式図

### 6.1.2.1 グリフイメージ

セル内部にはグリフイメージのみが配置されます。fontcvtrはセル内に存在する白以外のドットを含む最小の矩形を検出し、それをグリフイメージとして扱います。このためグリフイメージの周囲の白色部分は無視され出力されるフォントに影響を与えません。図 6-3(左)ではグリフイメージは幅 5 ピクセル高さ 7 ピクセルの矩形画像として抽出されます(図 6-3(右))。

幅線との位置関係が同じであればセルの中でグリフイメージを左右に移動しても出力されるフォントは変化しません。例えば図 6-4では左右どちらの書き方でも図 6-3と同じフォントとして出力されます。

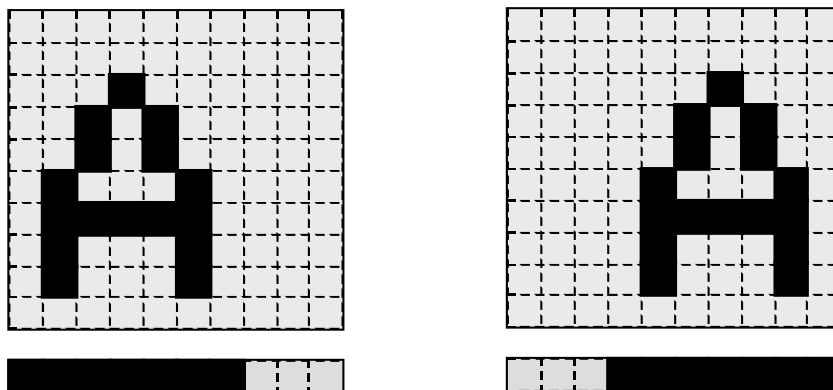


図 6-4 セルの中での左右位置

### 6.1.2.2 幅線

幅線領域には幅線と呼ぶ 1 つの線分のみが描かれます。線分が途中で途切れて 2 つになっていたりするとエラーとなります。

幅線は文字幅と文字の左右(前後)に空けるスペースの幅を規定します。

まず、幅線の横幅がそのまま文字幅となります。文字幅はグリフイメージの幅より小さくする事ができ、その場合には前の文字と重なって表示される事になります。図 6-3(左)では文字幅は 7 ピクセルです。

次に、幅線の左端とグリフイメージの左端との差が文字の左スペース幅となります。文字描画ライブラリで文字描画を行う時には左スペース幅分の隙間が空いてグリフが描画されます。図 6-3(左)では左スペースは 1 ピクセルとなり、文字描画時には指定された座標の 1 ピクセル右に幅 5 ピクセルのグリフが描画されることになります。

同様に幅線の右端とグリフイメージの右端との差が文字の右スペース幅となります。図 6-3(左)では右スペースも 1 ピクセルとなります。

### 6.1.2.3 出力されないグリフ

セルに白以外のドットが含まれず、幅線の幅も 0 である場合、このグリフは出力に渡されません。これを利用して文字フィルタを使用せずに出力するグリフを制御する事も出来ます。しかし、逆に文字フィルタでは出力する指定であるのにグリフが出力に渡されない場合は警告が表示されます。

## 6.1.3 配置情報

画像の一番左上のブロックには文字列を描画するときに文字列の配置の基準となる値を指定する 1 ピクセルの点がいっくつか含まれます。配置の基準となる値とはベースライン、アセンダライン、ディセンダライン、フォント幅の 4 つで、この



4 つを 5 つの点で指定します。これらの値はフォント全体で共通となり、文字ごとに指定することはできません。現在の G2D ライブラリではフォント幅は使用されません。

すべての点はグリッドエリアに存在し、グリッドが描画される場合は白色、グリッドが描画されない場合は黒色です。ベースライン位置を表すピクセルは必ず存在する必要がありますが、残りの 4 点は省略することができます。これら 5 点すべてを含む場合のブロックを図 6-5 に示します。

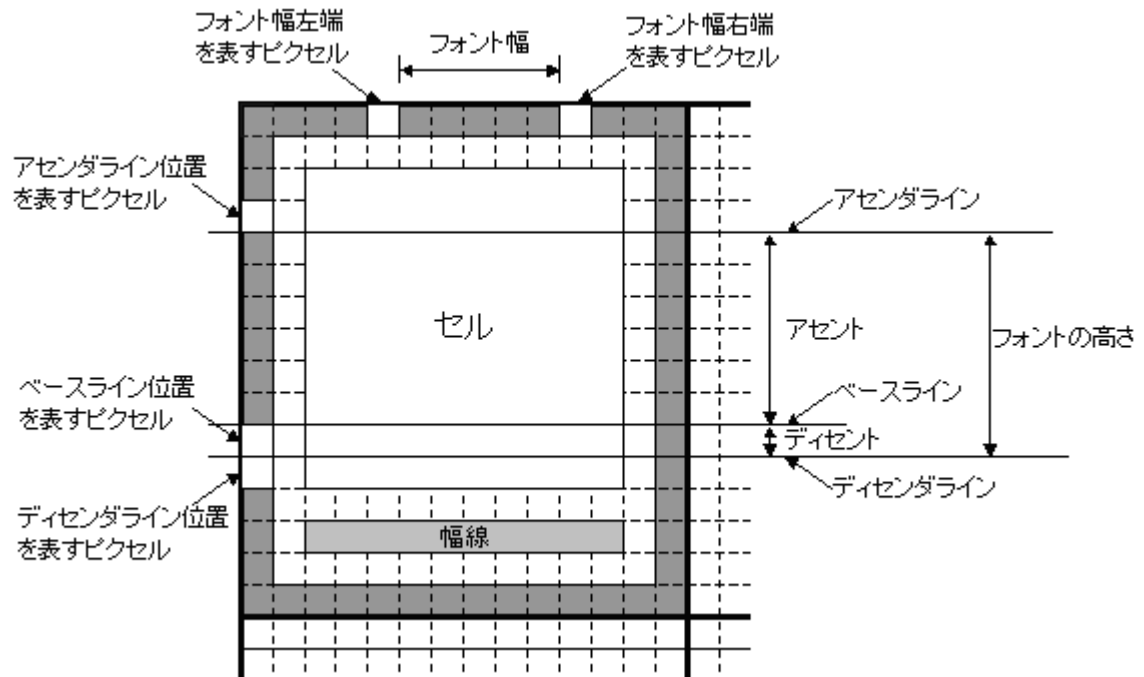


図 6-5 すべての配置情報ピクセル

### 6.1.3.1 ベースライン

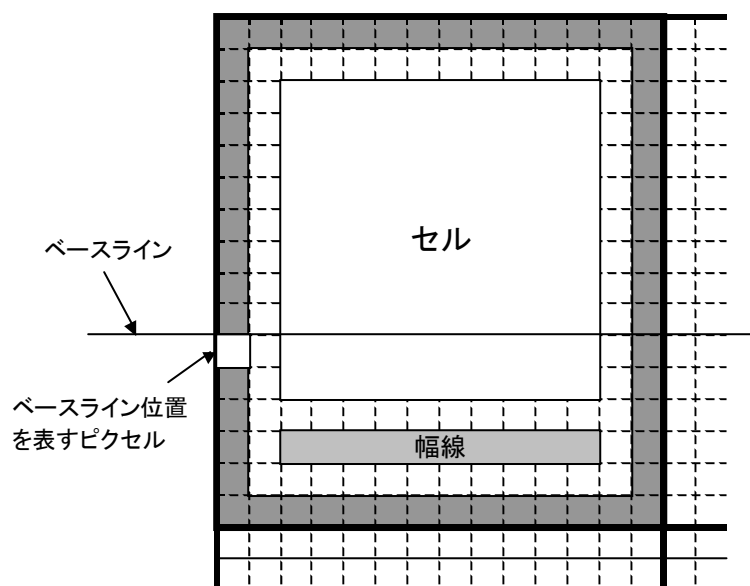


図 6-6 ベースライン位置模式図

ベースライン位置は1つの点で表されグリッドエリアの左側領域に存在します。このピクセルの上端がベースライン位置になります。すなわちベースラインはピクセルとピクセルの間に存在します。

### 6.1.3.2 アセンダライン、ディセンダライン

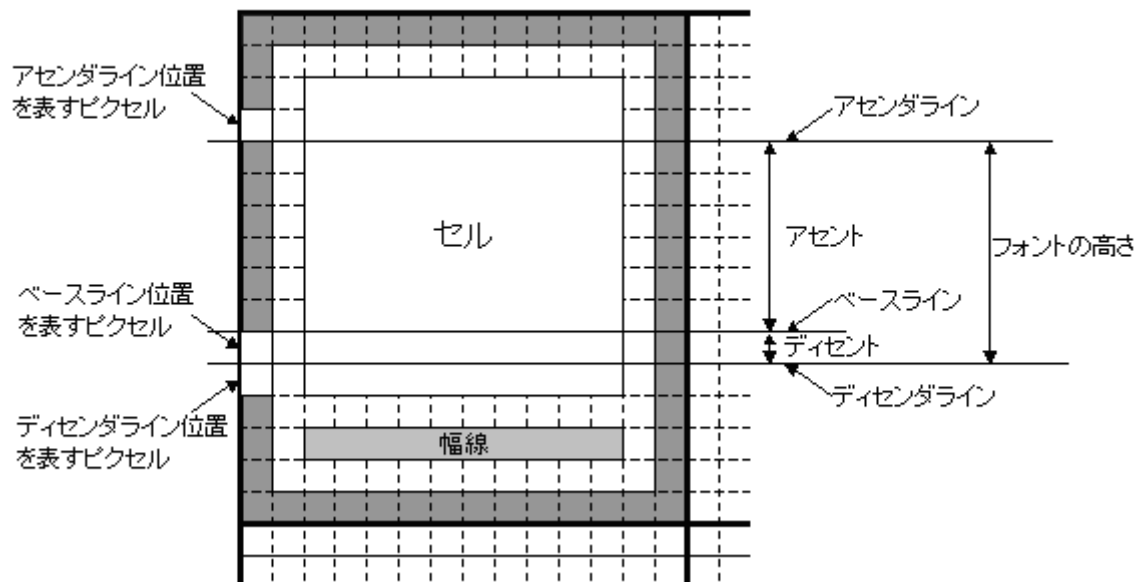


図 6-7 アセンダライン、ディセンダラインの位置模式図

アセンダライン、ディセンダラインはそれぞれ 1 つの点で表されグリッドエリアの左側領域に存在します。アセンダライン、ディセンダライン共にピクセルとピクセルの間に存在し、アセンダラインはアセンダライン位置を表すピクセルの下端が、

ディセンダラインはディセンダライン位置を表すピクセルの上端がそれぞれの位置になります。

アセンダラインは必ずベースラインの上方に位置し、ディセンダラインは必ずベースラインの下方に位置します。またディセンダライン位置を表すピクセルはベースライン位置を表すピクセルと重ねる事ができ、この場合ディセントは 0 になります。このためアセント、ディセントは共に 0 以上の値となります。

これらの点は省略する事ができ、点の数に応じて表 6-1 のように解釈されます。点が 1 つの場合、すなわちアセンダライン位置とディセンダライン位置を指定しなかった場合は、全ての文字をベースラインを合わせて重ね合わせたときに占める領域の上端がアセンダライン、下端がディセンダラインとして扱われます。例えば図 6-8 の 2 文字からなるフォントではフォントの高さは 9 ピクセルとなります。

fontcvtr が出力する画像ではアセンダライン位置、ディセンダライン位置を表すピクセルは、描かれる場合と描かれない場合があります。Windows フォントを入力とする変換では常に描かれず、brfnt を入力とする変換では常に描かれます。Image を入力とする場合は入力の画像にアセンダライン位置、ディセンダライン位置を表すピクセルが含まれていれば描かれ、含まれていなければ描かれませんが、アセンダライン位置、ディセンダライン位置を表すピクセルが描かれるときに出力する画像のセルサイズがこれらのピクセルを表現するのに小さすぎる場合はセルサイズが自動的に拡大されます。

アセンダラインは文字列の上端として、ディセンダラインは文字列の下端としてあつかわれます。またはアセンダラインとディセンダラインの距離をフォントの高さと呼び、縦方向に拡大縮小する場合の基準サイズとなります。

表 6-1 配置情報点の解釈

点の数	上から 1 点目	上から 2 点目	上から 3 点目
0	エラー		
1	ベースラインを表す点	-	
2	アセンダラインを表す点	ベースラインとディセンダラインを表す点	-
3	アセンダラインを表す点	ベースラインを表す点	ディセンダラインを表す点
4 以上	エラー		

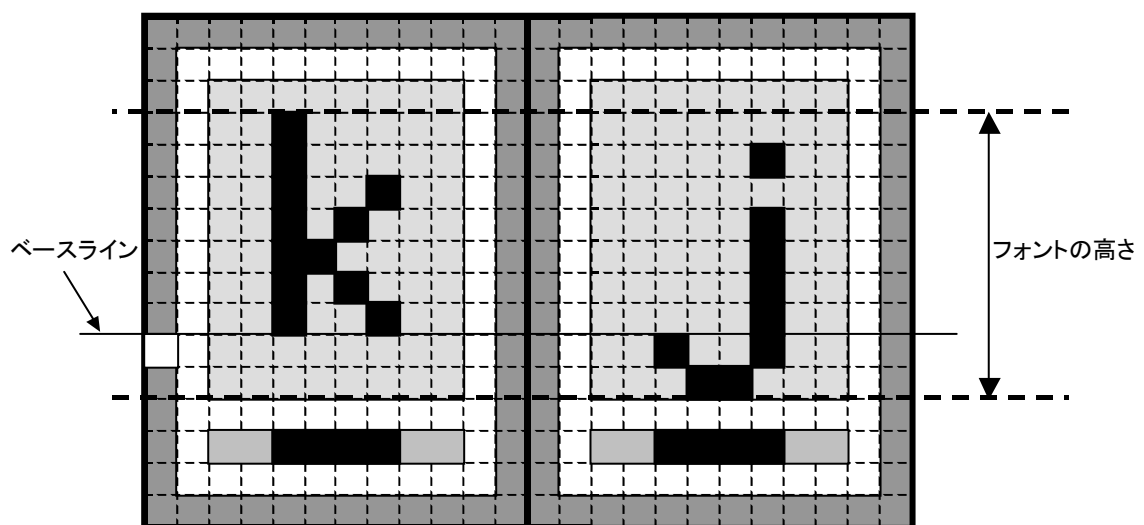


図 6-8 フォントの高さ

### 6.1.4 縦横識別マーカー

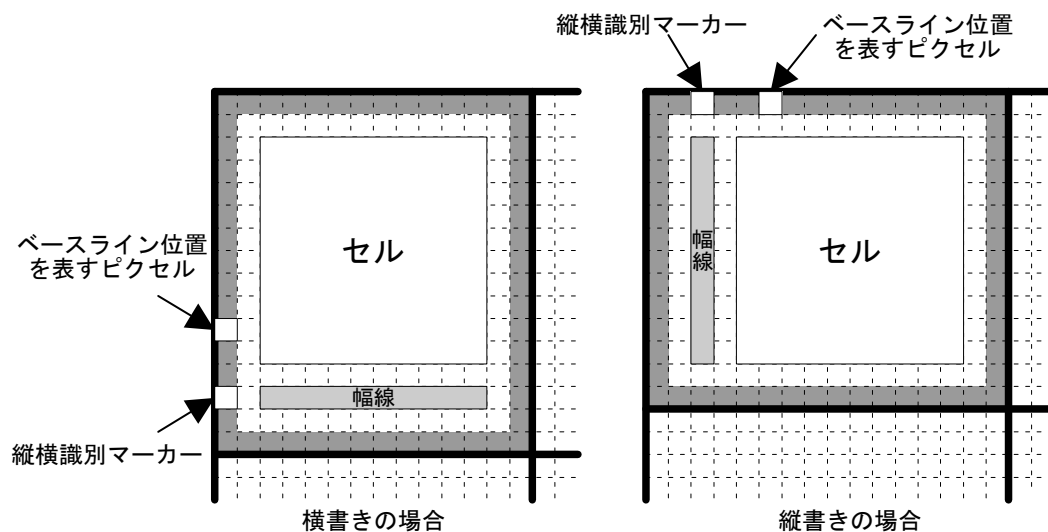


図 6-9 縦横識別マーカー

BMP画像の一番左上のブロックには横書きか縦書きかを識別するための1ピクセルの点も含まれます。この点は横書きの場合は幅線と同じ行の左端に、縦書きの場合は幅線と同じ列の上端に存在し、グリッドが描画される場合は白色、グリッドが描画されない場合は黒色です。過去のバージョンとの互換性のため縦横識別マーカーが存在しない場合は横書きであると判定します。縦書きについては「7.1 縦書きフォント」を参照してください。

## 6.2 出力されるBMPのフォーマット

fontcvtr で出力される BMP は1ピクセルあたり8ビット(256色)のインデックスカラーBMPです。

BMP のカラーパレットには出力されるフォントの階調数分のグレースケールが白から黒へ向かう順で出力されます。たとえば4階調であれば RGB=(255,255,255), (170,170,170), (85, 85, 85), (0, 0, 0) がこの順でカラーパレットに作成されます。これに続いて余白エリアを塗りつぶすための色が出力されます。

## 6.3 入力可能なBMPのフォーマット

fontcvtr に入力可能な BMP は1ピクセルあたり1, 4, 8ビット(2, 16, 256色)のインデックスカラーBMPです。

また

ブロックの幅 \* ブロックの横方向の数 = BMPの幅  
 ブロックの高さ \* ブロックの縦方向の数 = BMPの高さ

が成り立たなければなりません。すなわち BMP 中にブロックは隙間なく並び、余白があってははいけません

実際には文字順序ファイルによりブロックの横(縦)方向の数が、BMP ファイルにより BMP の幅(高さ)が決定し、そこからブロックの幅(高さ)が計算されます。そして、このブロックの幅(高さ)が整数でなければなりません。

fontcvtr では BMP が持つ色情報は全て無視され、そのカラーインデックスのみが意味を持ちます。読み込まれた BMP 画像中のカラーインデックスはそのまま NITRO フォント中のグリフイメージに受け継がれるため、TWL およびニンテンドーDS 上で BMP 画像と同じ出力を得るためには BMP 画像の持つカラーパレットと同じ順序、同じ色のカラーパレットを使用して文字描画を行う必要があります。ただし白色は例外で、白色は BMP 読み込み時にカラーインデックス0番の色と入れ替えられます。これは fontcvtr が白色を抜き色として扱っており、TWL およびニンテンドーDS においてカラーインデックス0番が抜き色を表すためです。

## 7 縦書き/縦持ちフォント

### 7.1 縦書きフォント

#### 7.1.1 縦書きフォントとは

縦書きフォントとは TWL-System G2D の文字列描画関数にて容易に縦書きを行うために特殊な加工が施された NITRO フォントです。

#### 7.1.2 縦書きフォントの作り方

縦書きフォントを作るには入力指定で縦書きのフォントを指定します。入力指定で縦書きのフォントが指定されていると、出力されるフォントは自動的に縦書きフォントとなります。それぞれの入力指定で縦書きを指定する方法は以下のとおりです。

- Windows フォント  
変換元のフォントにフォント名が「@」で始まるフォントを選択します。
- LC フォント  
LC フォントの入力指定のオプションで「Vertical」オプションを有効にします。
- BMP 画像  
縦書きフォントを BMP 画像出力したものを入力として指定します。
- NITRO フォント  
入力となる NITRO フォントに縦書きフォントを指定します。

#### 7.1.3 縦書きフォントのBMP画像出力

入力側が縦書き指定となっている場合にBMP画像に出力すると縦書きフォント用のBMP画像が出力されます(図 7-1)。縦書きフォント用のBMP画像は通常のBMP画像をブロック単位で時計回りに 90 度回転させたようなフォーマットとなります。

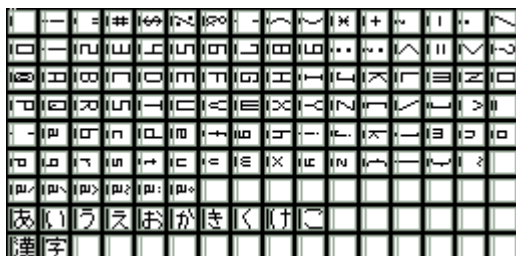


図 7-1 縦書きフォントの BMP 出力

横書きフォントを出力した BMP では幅線で文字幅を指定しますが、縦書きフォントを出力した BMP では幅線で文字の高さを指定することになります。またベースラインも横ではなく縦に存在します。

縦書きフォント用の BMP 画像を入力として指定すると出力は縦書きになります。

## 7.2 縦持ちフォント

---

### 7.2.1 縦持ちフォントとは

---

TWL およびニンテンドーDS を画面の長辺が縦になるように持つ持ち方を縦持ちと呼ぶ事にします。また、縦持ちにしたときに正しく読めるように文字列を描画する事を縦持ち描画と呼ぶ事にします。

縦持ちフォントとは TWL-System G2D の文字列描画関数にて容易に縦持ち描画をするために特殊な加工が施された NITRO フォントです。

### 7.2.2 縦持ちフォントの作り方

---

縦持ちフォントを作るには NITRO フォントを出力するときに、「Rotation」オプションで「0」以外を指定するだけです。「Rotation」オプションでは「0」、「90」、「180」、「270」が選択でき、それぞれ反時計回りに 0 度、90 度、180 度、270 度回転させた持ち方の場合のフォントを出力します。

「Rotation」オプションで「0」を選択した場合は通常のフォントを出力します。

## 7.3 縦書きと縦持ちの組み合わせ

---

NITRO フォントを出力するときに入力で縦書きのフォントを指定し、同時に「Rotation」オプションで「0」以外を指定すると縦持ち用の縦書きフォントを作る事ができます。よって、縦書きか横書きか、4 通りの持ち方のどれに対応するかで、通常のフォントを含め全部で 8 通りのバリエーションが存在する事になります。

## 8 使用上の注意

### 8.1 Windowsフォントから変換する際の注意

Windows2000および Windows XP には自動フォントリンクという機能があります。これはフォントに含まれていない文字の表示が必要となった時に自動的に他のフォントを使って文字の表示を行う機能です。例えば Windows2000 および Windows XP に標準でインストールされているフォント Tahoma は日本語を含まないフォントですが、メモ帳などでフォントを Tahoma に設定した状態で日本語を表示する事ができます。この機能は通常の PC の利用においては便利ですが、商用デザインにおいてはライセンスされていないフォントを商品に使用してしまう可能性があります。

fontcvtr では自動フォントリンク機能を無効化し、指定されたフォントに含まれている文字のみを出力します。そのため Tahoma のような日本語文字を含まないフォントでは日本語文字は出力されません。

### 8.2 fontcvtr内部でのUnicode

fontcvtr は内部での文字コードの処理に Unicode を使用しています。そのため Unicode で表現できない文字は扱う事が出来ません。また各エンコーディングの変換には Windows の機能を使用していますので Windows の変換ルールにもとづいて変換が行われます。例えば Shift JIS で文字コード 0x8160 の「〜」は、本来の Unicode の定義では U+301C となりますが、Windows では U+FF5E となります。fontcvtr でも同様に「〜」は U+FF5E として扱われます。

fontcvtrの各入出力で発生する文字コードの変換を表 8-1に示します。

表 8-1 文字コードの変換が発生する箇所

箇所	変換の内容
入力 BMP	文字順序ファイルを参照し、セルの位置から Unicode へ割り当てます。
入力 NITRO	NITRO フォントに格納されているエンコーディングから Unicode へ変換します。
入力 Windows	Unicode で U+0000 から U+FFFF までの文字を出力します。
入力 LC	Shift JIS から Unicode へ変換します。
出力 BMP	文字順序ファイルを参照し、Unicode からセルの位置へ割り当てます。
出力 NITRO	Unicode から指定されたエンコーディングに変換します。
文字フィルタファイル	ロード時に Unicode へ変換します。
文字順序ファイル	ロード時に Unicode へ変換します。

### 8.3 文字順序ファイル記述上の注意

同一の文字を複数回指定した文字順序ファイルを作成し、BMP の入力を使用して変換を行うと、同一の文字に割り当てられているセルのうち 1 つのセルのグリフのみが読み取られ、他のセルのグリフ無視されるという状況が発生します。このような状況では無視されているセルのグリフを書き換えてもフォントには反映されず原因のわかりにくいバグとなってしまいます。

fontcvtr ではこのような状況を発見しやすくするため、文字順序ファイルで同一の文字を複数回指定し、かつ、それぞれに異なるグリフが描画されている場合には警告を表示します。文字順序ファイルにはできるだけ同じ文字を複数回定義しないようにして下さい。

### 8.4 文字順序ファイルでのUnicode

fontcvtrは内部でUnicodeを使用していますので文字順序ファイルの文字も内部ではUnicodeに変換して使用しています。このときに、Unicode以外の文字セットでは異なる文字コードに割り当てられている複数の文字がUnicodeでは 1 つの文字コードにまとめて割り当てられている事があります。例えば表 8-2の 9 組の文字はWindowsで使われるShift JISでは 2 つの異なる文字コードに割り当てられていますがUnicodeでは 1 つの文字コードに割り当てられています。

これと前節の「文字順序ファイル記述上の注意」とをあわせて考えると、文字順序ファイル上では一見全て異なる文字コードであるのに、あるセルのグリフの変更が出力されるフォントに反映されないという事態が発生する事がわかります。ほとんどの場合、このようなUnicodeで 1 つにまとめられている文字群は同じグリフとなっていますので、使用する文字を 1 つに絞ってしまう事で回避する事が出来ます。

なお、文字順序ファイルで同一の文字を複数回指定した場合の警告は Unicode に変換した後に判定しますので上記の場合であっても警告が表示されます。

表 8-2 ShiftJIS から Unicode へ 2 対 1 で対応している文字

文字	Shift JIS で通常使われる文字コード	Shift JIS で機種依存文字領域の文字コード	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 警告/エラーメッセージ

### 9.1 警告メッセージ

fontcvtrの警告メッセージには表 9-1のものが 있습니다。表 9-1以外にも起動時に読み込みに失敗した文字順序ファイルがある場合、そのエラーメッセージを警告として表示します。

警告は異常の可能性を指摘するだけでするので変換は行われます。

**表 9-1 警告メッセージ説明**

警告メッセージ	説明
A character ({文字}: U+{文字コード}) included in the Filter is not include in the INPUT font.	{文字}は文字フィルタで出力する指定となっていますが入力側のフォントに含まれていません。そのため出力も行われません。
A character code('{文字}': U+{文字コード}) has multiple glyphs. Second and later glyphs are ignored.	{文字}に対して複数のグリフが対応しています。2 番目以降のグリフは無視されます。文字順序ファイルで同じ文字が複数回定義されている事が原因です。詳しくは「8.3 文字順序ファイル記述上の注意」を参照してください。
BMP position ({座標 x}, {座標 y}) is <null/> in {文字順序ファイルのパス}, but the cell include non-white color.	BMP 上の{座標}を含むセルは文字順序ファイルで<null/>となっていますが、白以外の色を含んでいます。グリフがはみ出して描画されている可能性があります。
Can't represent output character('{文字}': U+{文字コード}) in the specified character set.	{文字}を指定されたエンコーディングで表現できなかったため出力されませんでした。{文字}を出力するためには他のエンコーディングを使用する必要があります
More than one color detected in margin area at ({座標 x}, {座標 y}).	BMP 上の{座標}を含むセルの余白エリアに 2 色以上の色が見つかりました。グリフがはみ出して描画されている可能性があります。

### 9.2 エラーメッセージ

fontcvtr のエラーメッセージは大きく 5 種類にカテゴリー分けされています。表 9-2にエラーカテゴリーのリストを、表 9-3にInternal Error以外のエラーメッセージとその説明を記します。なお、表 9-3で { と } で囲まれている部分は実際のエラーメッセージでは置き換えられて表示されます。

エラーが発生するとその時点で変換は中止され、出力は行われません。

表 9-2 エラー種別説明

エラー種	説明
BMP Error	BMP ファイルとして指定されたファイルのエラーです。
NFTR Error	NITRO フォントファイルとして指定されたファイルのエラーです。
XML Error	XML ファイル(文字順序ファイル、文字フィルタファイル)として指定されたファイルのエラーです。
Parameter Error	その他の入力パラメータが不正である事を示すエラーです。
Internal Error	通常、発生しないエラーです。fontcvtr のバグである可能性があります。

表 9-3 エラーメッセージ説明

エラーメッセージ	説明
BMP Error: BMP has no baseline info.	BMPファイルがベースライン情報を持っていません。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: BMP height({BMP の高さ}) is not multiply of vertical number of blocks({縦方向ブロック数})	BMP画像の高さが文字順序ファイルで指定されている縦ブロック数の倍数になっていません。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: BMP height({BMP の高さ}) too small. Must be >= {BMP 縦方向最小値}	BMP画像の高さが小さすぎます。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: BMP width({BMP の幅}) is not multiply of horizontal number of blocks({横方向ブロック数})	BMP画像の幅が文字順序ファイルで指定されている横ブロック数の倍数になっていません。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: BMP width({BMP の幅}) too small. Must be >= ({BMP 横方向最小値})	BMP画像の幅が小さすぎます。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: Two or more width line found at ({座標 x}, {座標 y})	座標(x,y)を含むセルの幅線領域に 2 つ以上の線分があります。詳しくは「6.BMP画像フォーマット」を参照してください。
BMP Error: Unsupported BMP bpp({BMP の bpp}).	BMP 画像の 1 ピクセルあたりのビット数(bpp)がサポートされない値です。fontcvtr では 1,4,8bpp のみをサポートします。
BMP Error: Color white not found in BMP.	BMP 画像に白色が含まれていません。fontcvtr では白色を抜き色として扱うので必ず白色が含まれている必要があります。

Cannot open a file ({ファイルのパス}).	ファイルが開けません。表示されているファイルパスが正しいか確認してください。
Cannot read from a file ({ファイルのパス}).	ファイルを読めません。他のアプリケーションがファイルをロックしている可能性があります。
Cannot write to a file ({ファイルのパス}).	ファイルに書き込めません。他のアプリケーションがファイルをロックしている可能性があります。
Input file({ファイルのパス}) has illegal image size({BMP データサイズ} (expect {想定される BMP データサイズ})).	BMP ファイルに格納されている画像データのサイズが不正です。BMP ファイルが壊れているか、BMP ファイルでない可能性があります。
Input file({ファイルのパス}) has illegal image size({BMP の幅} x {BMP の高さ}).	BMP ファイルに格納されている画像の縦横サイズが不正です。BMP ファイルが壊れているか、BMP ファイルでない可能性があります。
Input file({ファイルのパス}) has unsupported bpp({BMP の bpp}).	BMP ファイルの 1 ピクセルあたりのビット数が不正な値です。BMP ファイルが壊れているか、BMP ファイルでない可能性があります。
Input file({ファイルのパス}) is not bmp file.	BMP ファイルとして指定されたファイルは BMP ファイルではありません。
Input file({ファイルのパス}) is unsupported BMP.	指定された BMP ファイルは fontcvtr で扱えない形式です。具体的には圧縮されている BMP や複数面をもつ BMP がサポートされていません。
NFTR File Error:  Little endian byte order support only	NITRO フォントファイルのもつバイトオーダーマークがビッグエンディアンを示しています。fontcvtr はリトルエンディアンのみに対応します。また NITRO フォントファイルが壊れているか、NITRO フォントファイルでない可能性があります。
NFTR File Error:  The file is not NFTR file	NITRO フォントファイルとして指定されたファイルは NITRO フォントファイルではありません。
NFTR File Error:  Unknown header size(={ファイルのヘッダサイズ}, must be {正しいヘッダサイズ})	NITRO フォントファイルのヘッダサイズが不正です。NITRO フォントファイルが壊れているか、NITRO フォントファイルでない可能性があります。
NFTR File Error:  Unsupported version(={ファイルのバージョン}, supports {サポートしているバージョン})	サポートされないバージョンの NITRO フォントです。
NFTR File Error:  Number of blocks is too small(={ファイル中のデータブロック数})	NITRO フォントファイル中のデータブロック数が不正です。NITRO フォントファイルが壊れているか、NITRO フォントファイルでない可能性があります。

<p>NFTR File Error:</p> <p>Can't represent input character(0x{文字コード}) in the Unicode character set.</p>	<p>NITRO フォントに Unicode で表現できない文字が含まれています。fontcvtr では Unicode で表現できない文字は扱う事ができません。もしくは NITRO フォント内の文字コードテーブルと現在のローカル文字セットが異なります。</p>
<p>Parameter Error:</p> <p>LC Font file path required</p>	<p>LC フォントファイルが指定されていません。</p>
<p>Parameter Error:</p> <p>Can't represent AlternateChar('{ 代替文字 }': U+{代替文字のユニコードでの文字コード}) in the local character set.</p> <p>Please use Unicode character set or other AlternateChar.</p>	<p>代替文字として指定された文字はローカル文字コードでは表現出来ません。NITRO フォントを Unicode で出力するか、別の代替文字を指定してください。</p>
<p>Parameter Error:</p> <p>Can't represent AlternateChar(0x{代替文字の文字コード}) in the Unicode character set.</p> <p>Please use other AlternateChar.</p>	<p>代替文字として指定された文字は Unicode に含まれません。fontcvtr では Unicode で表現できない文字は扱う事ができません。別の代替文字を指定してください。</p>
<p>Parameter Error:</p> <p>DefaultLeftSpace(={入力された値}) out of range[-128 ~ 127]</p>	<p>デフォルトの左スペース幅が指定可能な範囲を超えています。指定可能な値は-128～127です。</p>
<p>Parameter Error:</p> <p>DefaultWidth(={入力された値}) out of range[{最小値} ~ {最大値}]</p>	<p>デフォルトの文字幅が指定可能な範囲を超えています。指定可能な値は{最小値}～{最大値}です。</p>
<p>Parameter Error:</p> <p>DefaultRightSpace(={入力された値}) out of range[{最小値} ~ {最大値}]</p>	<p>デフォルトの右スペース幅が指定可能な範囲を超えています。指定可能な値は{最小値}～{最大値}です。</p>
<p>Parameter Error:</p> <p>Directory({ディレクトリのパス}) does not exist.</p>	<p>入力ファイルまたは出力ファイルとして指定されたパスが存在しません。</p>
<p>Parameter Error:</p> <p>File not specified</p>	<p>ファイルが指定されていません。</p>
<p>Parameter Error:</p> <p>File({ファイルのパス}) does not exist.</p>	<p>指定されたファイルが存在しません。パスが正しいか確認してください。</p>
<p>Parameter Error:</p> <p>Invalid Font Size({指定されたフォントサイズ})</p>	<p>フォントサイズの指定が不正です。</p>

Parameter Error: Invalid alternate character.(0x0000)	代替文字に 0x0000 は指定できません。
Parameter Error: Invalid fixed width({指定された文字幅})	等幅文字幅の指定が不正です。
Parameter Error: LineHeight(={指定された値}) out of range[0 ~ 255]	行の高さの指定が不正です。指定可能な値は 0～255 です。
Parameter Error: Max glyph width(={グリフ幅}) over 255.	グリフ幅が 255 ピクセルを超えました。グリフ幅は 255 以下でなければなりません。
Parameter Error: Max char width(={文字幅}) over 255.	文字幅が 255 ピクセルを超えました。文字幅は 255 ピクセル以下でなければなりません。
Parameter Error: Font height(={フォントの高さ}) over 255.	フォントの高さが 255 ピクセルを超えました。フォントの高さは 255 ピクセル以下でなければなりません。
Parameter Error: Baseline position(={ベースライン位置}) over 127.	ベースライン位置が 127 ピクセルを超えました。ベースライン位置は-128 以上 127 以下でなければなりません。
Parameter Error: Baseline position(={ベースライン位置}) under -128.	ベースライン位置が-128 ピクセルを下回りました。ベースライン位置は-128 以上 127 以下でなければなりません。
Parameter Error: No glyphs in output.	出力すべきグリフが 1 つもありません。入力に文字が全く含まれていないか、もしくは入力に含まれる全ての文字がフィルタされています。西欧文字フォントで日本語文字のみを出力しようとしている、文字順序ファイルに含まれる文字と文字フィルタファイルに含まれる文字が全く違うものになっている、などが考えられます。
Parameter Error: Specified AlternateChar('指定された代替文字' : {文字コード(ローカル)}) is not included in the Font	指定された代替文字はフォントに含まれていません。代替文字に指定する文字は出力するフォントに含まれている必要があります。
Parameter Error: Specified AlternateChar('指定された代替文字' : {文字コード(Unicode)}) is not included in the Font	指定された代替文字はフォントに含まれていません。代替文字に指定する文字は出力するフォントに含まれている必要があります。
Parameter Error: Specified file({ファイルのパス}) is read only.	出力用に指定されたファイルは読み取り専用指定されているため書き込めません。
Parameter Error: Specified path({ファイルのパス}) is a directory.	ファイルを指定しなければならないところでディレクトリが指定されています。

Parameter Error: Filter file({ファイルのパス}) does not exist.	指定されたフィルタファイルは存在しません。パスが正しいか確認してください。
Parameter Error: Input file({ファイルのパス}) is not LC Font file.	LC フォントファイルとして指定されたファイルは LC フォントファイルではありません。
Parameter Error: Cell Width (= {指定した値}) must be > 0.	セル幅の指定は 0 より大きくなければなりません。
Parameter Error: Cell Height (= {指定した値}) must be >= 0.	セル高の指定は 0 より大きくなければなりません。
Parameter Error: Cell Margin Left (= {指定した値}) must be >= 0.	セルの左マージンは 0 以上でなければなりません。
Parameter Error: Cell Margin Right (= {指定した値}) must be >= 0.	セルの右マージンは 0 以上でなければなりません。
Parameter Error: Cell Margin Top (= {指定した値}) must be >= 0.	セルの上マージンは 0 以上でなければなりません。
Parameter Error: Cell Margin Bottom (= {指定した値}) must be >= 0.	セルの下マージンは 0 以上でなければなりません。
Parameter Error: Order file is not selected.	文字順序ファイルが選択されていません。
Parameter Error: "Use Filter" checked, but filter file path not specified.	「Use Filter」にチェックがついていますが、文字フィルタファイルのパスが入力されていません。
Parameter Error: Output encoding is not selected	出力する NITRO フォントのエンコーディングが選択されていません。
XML Error: Invalid filter file	不正な文字フィルタファイルです。
XML Error: Invalid order file.	不正な文字順序ファイルです。
XML Error: Invalid order height (= {文字順序ファイルで指定された高さ}).	文字順序ファイルで指定されている高さの値が不正です。
XML Error: Invalid order width (= {文字順序ファイルで指定された幅}).	文字順序ファイルで指定されている幅の値が不正です。

## 10 Xerces-C++について

この fontcvtr は、Apache Software Foundation (<http://www.apache.org/>) により開発された Xerces-C++を使用しています。Xerces-C++の NOTICE ファイルとライセンスファイルのコピーは、下記のディレクトリに収録しています。

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

Microsoft、Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

LC フォントはシャープ株式会社の商標です。

その他、記載されている会社名、製品名等は、各社の登録商標または商標です。

© 2004-2009 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。