

- [使い方](#)
 - [オプション](#)
 - [gdi++.iniの例](#)
 - [オプション解説](#)
 - [HookChildProcesses](#)
 - [HintingMode](#)
 - [AntiAliasMode](#)
 - [ForceChangeFont](#)
 - [TextTuning、TextTuningR、TextTuningG、TextTuningB](#)
 - [NormalWeight](#)
 - [BoldWeight](#)
 - [ItalicSlant](#)
 - [EnableKerning](#)
 - [MaxHeight](#)
 - [Individual](#)
 - [Exclude](#)
 - [ExcludeModule](#)
 - [UseInclude](#)
 - [IncludeModule](#)
 - [オプション解説\(マニア向け\)](#)
 - [LoadOnDemand](#)
 - [UseMapping](#)
 - [UseManager](#)
 - [LcdFilter](#)
 - [Shadow](#)
 - [CacheMaxFaces、CacheMaxSizes、CacheMaxBytes](#)
 - [その他](#)
 - [縦書き？](#)
-

使い方

1. [gdi++.dll\(FreeType版\)のアーカイブ](#)を入手し、適当なローカルフォルダに解凍。
2. [freetype.dll](#)を入手し、freetype.dllを
XPの場合: gdi++.dllと同じフォルダ
2kの場合: PATHの通ったフォルダ(system32とか)
にコピー。
3. 下の[オプション](#)を参考にgdi++.iniを編集して、gdi++.dllの動作を設定。
4. gdi++.exeに対象exeをドロップして起動。

期待した結果が得られない場合、gdi++.iniの当該箇所を編集し直す。たぶん結構な試行錯誤が必要。

オプション

gdi++.iniの例

FreeTypeの設定は[General]に書いても動作する。
先頭が;とか#とか//とかREMとか'の行はコメント扱い。

```
[General]
HookChildProcesses=1
[FreeType]
HintingMode=2
AntiAliasMode=2
ForceChangeFont=
TextTuning=0
TextTuningR=0
TextTuningG=0
TextTuningB=0
NormalWeight=0
BoldWeight=0
ItalicSlant=0
EnableKerning=0
MaxHeight=0
LoadOnDemand=0
UseMapping=0
UseManager=0
UseInclude=0
#LcdFilter=0
#Shadow=1,1,4
CacheMaxFaces=0
CacheMaxSizes=0
CacheMaxBytes=0
[Individual]
メイリオ=0
MS Pゴシック=,0,,
Arial=,,,,,1
MingLiU=0
PMingLiU=0
DF行書体=0
DF特太ゴシック体=0
DF平成ゴシック=0
DF平成ゴシック体W5=0
DF平成明朝体W3=0
DF平成明朝体W7=0
DHP行書体=0
DHP特太ゴシック体=0
DHP平成ゴシックW5=0
DHP平成明朝体W3=0
DHP平成明朝体W7=0
[Exclude]
Marlett
Tahoma
[ExcludeModule]
fontview.exe
[IncludeModule]
```

オプション解説

HookChildProcesses

子プロセスにも自動でgdi++を適用

HintingMode

ヒンティングの設定。

0だとバイトコードインタプリタを使用します(フォントに含まれるヒンティングそのもの)。

1だとヒンティングをしません。

2はFreeTypeのオートヒンティングを使用します。

FreeTypeライブラリのバイトコードインタプリタがOFFになっている場合は0と2が同等になります。

一部のフォントではバイトコードインタプリタによるヒンティングを行わないと字形が崩れます(例：DF平成ゴシック)。

AntiAliasMode

-1でアンチエイリアスを無効化。

0なら普通のアンチエイリアス(256階調)をかけます。

1はLightモードです。FreeTypeのReferenceによるとMac OS Xっぽい描画になるらしい。

2、3は液晶用。並びが違います。

1を指定したときはHintingModeは2(オートヒンティング)にしてください。Lightモード(1)はノーマルモード(0)と比べて字形の変化が少なくなります。

ForceChangeFont

強制的にそのフォントを使うようになる

ダイアログが見づらくなるが一時的に特定のフォントで確認したい時に使用する

TextTuning, TextTuningR, TextTuningG, TextTuningB

文字の締まりを良くするようなもの。

-1を指定すると無効化できる。(注: 0でも有効になっている)

0-12までの値を指定できます。

TextTuningはAntiAliasModeが0か1の時に使用します。

TextTuningR・TextTuningG・TextTuningBはAntiAliasModeが2か3の時に使用します。

文字と背景の境界を調節し、背景側を文字側に侵食させる(背景を強く出す)ような形になっています。より大きな値を指定すると文字がより締まって見えますがアンチエイリアスの効果も落ちるので注意してください。

NormalWeight

標準のウェイトを相対値で指定(-32 ~ +32)。BoldWeightより変化幅が小さい。

BoldWeight

太字のウェイトを相対値で指定(-16 ~ +16)。マイナスにすると細くなる。

ItalicSlant

斜体の傾斜を相対値で指定(-16 ~ +16)。

EnableKerning

0だと文字詰めなし
1だと文字詰めあり
ONにするとIEとかでずれるので注意

MaxHeight

FreeTypeで処理させる最大のフォントサイズ。(ピクセル単位)
0で全てのサイズ。初期値は0。

Individual

フォント別に個別設定する。
書式: フォント=Hinting,AAMode,NormalWeight,BoldWeight,ItalicSlant,Kerning
例のように省略可能。その場合は[FreeType]で設定した共通設定が使われる。

Exclude

除外フォント。本家と違って上限は無し。

ExcludeModule

除外モジュールリスト。本家と違って上限は無し。

UseInclude

0の場合(もしくは未定義の場合)は[ExcludeModule]が有効。(従来と同じ)
1の場合[IncludeModule]が有効。
Managerによる自動適用は使いたいが、実際に適用するのは特定のアプリに限定したいという場合に1を指定してIncludeModuleの方を使う。

IncludeModule

対象モジュールリスト。ExcludeModuleの逆で適用したいモジュールを定義する。

オプション解説(マニア向け)

以下はマニア向けオプションなので使用は自己責任で。

LoadOnDemand

フォントを読み込むのをCreateFont系から要求時(TextOut系など)にする。
FAQにあるManager使用時の問題が解決されるかもしれない。
ハングアップする可能性があるので低スペックマシンでは絶対にONにしない事。

UseMapping

フォント読み取りにMemory Mappingを使う。

UseManager

Managerが使用可能になる。(標準では使用できない)

LcdFilter

FT_Library_SetLcdFilterに渡すパラメータ。
0:None, 1:Default, 2:Light, 16:Legacy
LegacyにするとLibXftと同じアルゴリズムらしい。
2.3.0以降で使用する事。

Shadow

影をつける。
書式: Shadow=x,y,t (x>0, y>0, t>0, パラメータ省略は不可)
オフセット(x,y)に100/t (%)の透過率で描画

CacheMaxFaces, CacheMaxSizes, CacheMaxBytes

FTC_Manager_Newに渡すパラメータ。詳細はFreeTypeのマニュアル参照
パラメータチェックしてないので大きな値を設定すると暴走する可能性あり

その他

- mfdel.exeを使ってManifestを削除すると、XPでも2kと同じ動作にできる。
もちろんgdi++.dllを上書きする度にmfdel.exeを使う必要がある。
- gdippInfo.exeを使うと質問用のテンプレを生成できる。
ただしFreeTypeの情報は自分で埋める。(特にCVSとか)
- gdippEST.exeを使うと設定が簡単に変更できる。
- VC6又はDDKを使ってCVSのFreeTypeを簡単にビルドできるセットがgdi0231にある。
VC8以降を使う場合はgdi0236の.makをコピーする。
- さらに、バッチファイルを実行するだけで全自動でビルドできるセットもある。ビルドに必要なcvsやpatchが予め入っているのでVC持ってるならお勧め。
VC6用: gdi0258.zip (オリジナル)
VC2003用: gdi0265.zip
VC2005EE用: gdi0411.zip
これは「やり方分かってるけど面倒な人向け」の物なので、BATの中身を見て何をやるか理解した上で実行すること。
- 十分なCPU、1GB超のメモリを積んでいるPC向け豪速化設定の例(自己責任)
gdi++.dllがメモリを食い潰すのでManager使うならこんな設定にはしない事。

その1

```
UseMapping=1
CacheMaxFaces=16
CacheMaxSizes=16
CacheMaxBytes=16777216
```

その2

```
UseMapping=1
CacheMaxFaces=8
CacheMaxSizes=8
CacheMaxBytes=1048576
```

縦書き？

1. FreeTypeのソースコードを取ってくる。

2. freetype.dswまたはfreetype.slnをVCで読み込んで

1. "Win32 Release Multithreaded"を選択する。
2. otvalid\otvalid.cを追加する。
3. base\ftlcdfil.cを追加する。
4. ftmodule.hにFT_USE_MODULE(otv_module_class)を追記する。

3. プロジェクトの設定(プロパティ)で

1. 使用するランタイムライブラリを"マルチスレッド(DLL)"にする。
2. 出力ファイル名をfreetypeMT.libにする。

4. ビルド。

5. freetypeMT.libをgdi++.dllのsrcフォルダにコピーする。
6. nmake ftstatic=1と打ってgdi++.dllをビルドする。

出来たgdi++.dllでgdi0344相当の縦書きが使えるようになるはず。
VC6でしか試してないので他はシネ。