

# CGL Reference

## (1) ピクセルフォーマットの操作

### ・ CGLChoosePixelFormat

アトリビュートの配列で指示されたピクセルフォーマットオブジェクトを作成する (Mac OS X 10.0以降)

```
CLError CGLChoosePixelFormat(const CGLPixelFormatAttribute *attribs,  
CGLPixelFormatObj *pix, long *npix);
```

attribs

- ・アトリビュートの配列 (ターミネータはNULL)

pix

- ・作成されたピクセルフォーマットオブジェクトを返す (存在しなければNULL)

npix

- ・ピクセルフォーマットで参照される仮想スクリーンの個数を返す (無ければゼロ)

ピクセルフォーマットを作成するためのアトリビュートには指示することでONにするタイプと直後に整数値を代入するタイプの2種類が存在する。

(a) ピクセルフォーマットオブジェクトのアトリビュートは以下の通り

kCGLPFAAllRenderers

- ・すべてのレンダラーを利用するようにする (デバッグ用なども含む)

kCGLPFADoubleBuffer

- ・ FrontとBackのダブルバッファ・ピクセルフォーマットが考慮される

kCGLPFAStereo

- ・ LeftとRightのステレオバッファ・ピクセルフォーマットが考慮される

kCGLPFAAuxBuffers

- ・ 使用可能とする補助バッファの個数を設定する（負ではない整数）

kCGLPFAColorSize

- ・ カラーバッファのビットサイズ（負ではない整数）

kCGLPFAAlphaSize

- ・ アルファ（透明度）バッファのビットサイズ（負ではない整数）

kCGLPFADepthSize

- ・ デプスバッファのビットサイズ（負ではない整数）

kCGLPFAStencilSize

- ・ ステンシルバッファのビットサイズ（負ではない整数）

kCGLPFAAccumSize

- ・ キュムレータバッファのビットサイズ（負ではない整数）

kCGLPFAMinimumPolicy

- ・ 要求最小バッファ（color、depth、accum）以内のピクセルフォーマットを選択

kCGLPFAMaximumPolicy

- ・ 要求最大バッファ（color、depth、accum）を優先しピクセルフォーマットを選択

kCGLPFAOffScreen

- ・オフスクリーンへのレンダリングを実行可能に（kCGLPFAClosestPolicyがONに）

kCGLPFAScreen

- ・フルスクリーンへのレンダリングを実行可能にする

kCGLPFAAuxDepthStencil

- ・補助バッファはそれぞれにステンシルのデプスサイズを持つ

kCGLPFAColorFloat

- ・カラーバッファのピクセル値として浮動小数点を使う

kCGLPFAMultisample

- ・マルチサンプリングを選択

kCGLPFASupersample

- ・スーパーサンプリングを選択

kCGLPFASampleAlpha

- ・アルファフィルタリングを要求

kCGLPFASampleBuffers

- ・マルチサンプルバッファ数（アトリビュート配列内では指定できない）

kCGLPFASamples

- ・ひとつのマルチサンプルバッファに対してのサンプル数（負ではない整数）

kCGLPFARendererID

- ・利用するレンダラーをID番号で指定する（種類についてはCGLRenderers.hを参照）

kCGLPFASingleRenderer

- ・すべてのスクリーンに対しひとつだけのレンダラーを割り当てる（有用ではない）

kCGLPFANoRecovery

- ・OpenGLのライブラリリカバリー機能をOFFにする（有用ではない）

kCGLPFAAccelerated

- ・ハードウェアアクセラレーションを有効にする

kCGLPFAClosestPolicy

- ・カラーバッファのサイズを優先しピクセルフォーマットを選択する

kCGLPFARobust

- ・フェイラーモードを持たないレンダラーを考慮する（有用ではない）

kCGLPFABackingStore

- ・OpenGLはバックカラーバッファを持ったレンダラーのみ考慮する

kCGLPFAMPSafe

- ・OpenGLはスレッドセーフのレンダラーのみ考慮する（有用ではない）

kCGLPFASingleWindow

- ・ウィンドウに対して描画可能なレンダラーのみが考慮される（有用ではない）

kCGLPFAMultiScreen

- ・マルチスクリーンに描画可能なレンダラーのみが考慮される（有用ではない）

kCGLPFACompliant

- ・OpenGL準拠のレンダラーを使用する（アトリビュート配列内では指定不可）

kCGLPFADisplayMask

- ・使用するスクリーンのビットマスクを指定する（CGDirectDisplay.h参照）

kCGLPFAPBuffer

- ・ピクセルバッファへのレンダリングが行われる

kCGLPFARemotePBuffer

- ・ピクセルバッファにオフラインでレンダリング実行

kCGLPFVirtualScreenCount

- ・仮想スクリーンの個数を返す（アトリビュート配列内では指定不可）

## ・ CGLDestroyPixelFormat

ピクセルフォーマットオブジェクトを破棄する（Mac OS X 10.0以降）

```
CGLError CGLDestroyPixelFormat(CGLPixelFormatObj pix);
```

pix

- ・破棄したいピクセルフォーマットオブジェクト

CGLCreateContext()は、渡されたピクセルフォーマットをコピーしてから使用するの  
で、CGLCreateContext()を実行した後は、すぐさまCGLDestroyPixelFormat()を実  
行する必要がある。

## ・ CGLDescribePixelFormat

ピクセルフォーマットオブジェクトの情報を得る（Mac OS X 10.0以降）

```
CGLError CGLDescribePixelFormat(CGLPixelFormatObj pix, long pix_num,  
CGLPixelFormatAttribute attrib, long *value);
```

pix

- ・ 情報を得たいピクセルフォーマットオブジェクト

pix\_num

- ・ ピクセルフォーマットオブジェクトの仮想スクリーン番号

attrib

- ・ 数値を得たいアトリビュート

value

- ・ 得られた数値

仮想スクリーンの個数を得たい場合には、pix\_numにゼロを代入しておいて、attribにはkCGLPFVirtualScreenCountを渡す。

## (2) ピクセルバッファの操作

- ・ CGLCreatePBuffer

描画対象となるピクセルバッファオブジェクトを作成する (Mac OS X 10.3以降)

```
CGLError CGLCreatePBuffer(long width, long height, unsigned long target,  
unsigned long internalFormat, long max_level, CGLPBufferObj *pbuffer);
```

width

- ・ ピクセルバッファの幅 (単位はピクセル)

height

- ・ ピクセルバッファの高さ (単位はピクセル)

target

- ・ OpenGL定数により対象となるテクスチャのターゲットを指定する

internalFormat

- ・ OpenGL定数によりカラーフォーマットを指定する (GL\_RGBかGL\_RGBA)

max\_level,

- ・ ミップマップ (OpenGL仕様参照) の最大レベルを指定する (必要無い場合はゼロ)

pbuffer,

- ・ 作成されたピクセルバッファオブジェクトが返る

テクスチャのターゲット (target) としては、GL\_TEXTURE\_2D (ノーマル2Dテクスチャ) GL\_TEXTURE\_RECTANGLE\_EXT (一辺が2の階乗サイズではない2Dテクスチャ) GL\_TEXTURE\_CUBE\_MAP (キューブマップテクスチャ) を選択できる。詳しくはOpenGLのテクスチャ仕様を参照。

## ・ CGLDescribePBuffer

指定されたピクセルバッファオブジェクトの各種情報を得る (Mac OS X 10.3以降)

```
GLenum CGLDescribePBuffer(CGLPBufferObj obj, long *width, long *height,  
    unsigned long *target, unsigned long *internalFormat, long *mipmap);
```

obj

- ・ 情報を得たいピクセルバッファオブジェクト

width

- ・ ピクセルバッファの幅が返る (単位はピクセル)

height

- ・ ピクセルバッファの高さが返る (単位はピクセル)

target

- ・対象となるテクスチャのターゲットが返る

internalFormat

- ・カラーフォーマットが返る

mipmap

- ・ミップマップ（OpenGL仕様参照）の最大レベルが返る
- ・ CGLDestroyPBuffer

ピクセルバッファオブジェクトを破棄する（Mac OS X 10.3以降）

```
CLError CGLDestroyPBuffer(CGLPBufferObj pBuffer);
```

pbuffer

- ・破棄するピクセルバッファオブジェクト
- ・ CGLGetPBuffer

コンテキストオブジェクトに関連づけられたピクセルバッファオブジェクトの各種情報を返す（Mac OS X 10.3以降）

```
CLError CGLGetPBuffer(CGLContextObj ctx, CGLPBufferObj *pbuffer,  
    unsigned long *face, long *level, long *screen);
```

ctx

- ・対象となるコンテキストオブジェクト

pbuffer

- ・コンテキストオブジェクトに関係したピクセルバッファオブジェクトが返る



face

- ・ターゲットがGL\_TEXTURE\_CUBE\_MAPなら設定値が返る（他はゼロ）

level

- ・設定されているミップマップレベルが返る

screen

- ・現在の仮想スクリーン番号が返る

## ・ CGLSetPBuffer

コンテキストオブジェクトの描画対象としてピクセルバッファオブジェクトを指定する  
(Mac OS X 10.3以降)

```
CGLError CGLSetPBuffer(CGLContextObj ctx, CGLPBufferObj pBuffer,  
unsigned long face, long level, long screen);
```

ctx

- ・対象となるコンテキストオブジェクト

pbuffer

- ・設定するピクセルバッファオブジェクト

face

- ・ターゲットがGL\_TEXTURE\_CUBE\_MAPなら6種類から選ぶ（他はゼロを設定）

level

- ・描画のためのミップマップレベル

screen

- ・レンダリングに用いられる仮想スクリーン番号

コンテキストオブジェクトのターゲットがGL\_TEXTURE\_CUBE\_MAPの場合、faceで選択できるパラメータは6種類ある（OpenGLの仕様を参照）。

### ・ CGLTexImagePBuffer

ピクセルバッファオブジェクトの内容をテクスチャとしてバインドする（Mac OS X 10.3以降）

```
CLError CGLTexImagePBuffer(CGLContextObj ctx, CGLPBufferObj pbuffer,
    unsigned long source);
```

ctx

- ・対象となるコンテキストオブジェクト

pbuffer

- ・利用するピクセルバッファオブジェクト

source

- ・対象となるOpenGLバッファ（GL\_FRONTやGL\_BACKなど）

CGLTexImagePBuffer()は、OpenGLコマンドのglTexImage2D()と同じ働きをする。

## (3) レンダラー情報

### ・ CGLQueryRendererInfo

指定されたディスプレイに関連したレンダラー情報オブジェクトを得る（Mac OS X 10.0以降）

```
CLError CGLQueryRendererInfo(unsigned long display_mask,
    CGLRendererInfoObj *rend, long *nrend);
```

display\_mask

- ・ CGDisplayIDToOpenGLDisplayMask() で得た OpenGL ディスプレイマスク

rend

- ・ レンダラー情報オブジェクトが返る (対象情報がなければ NULL が返る)

nrend

- ・ レンダラーの個数が返る (対象レンダラーがなければゼロが返る)

display\_mask を得るための CGDisplayIDToOpenGLDisplayMask() については、Core Graphics (CGDirectDisplay.h) を別途参照。

## ・ CGLDestroyRendererInfo

レンダラー情報オブジェクトを破棄する (Mac OS X 10.0以降)

```
CGLError CGLDestroyRendererInfo(CGLRendererInfoObj rend);
```

rend

- ・ 破棄の対象となるレンダラー情報オブジェクト

## ・ CGLDescribeRenderer

レンダラー情報オブジェクトより情報を抽出する (Mac OS X 10.0以降)

```
CGLError CGLDescribeRenderer(CGLRendererInfoObj rend, long rend_num,  
CGLRendererProperty prop, long *value);
```

rend

- ・ 対象となるレンダラー情報オブジェクト

rend\_num

- ・対象となるレンダラー番号（ゼロから個数-1まで）

prop

- ・レンダラーのプロパティの種類

value

- ・返されたプロパティの値

rend-numにゼロを代入してプロパティにkCGLRPRendererCountを与えるとレンダラーの個数を得ることが出来る。

(a) レンダラープロパティの種類は以下の通り

kCGLRPOffScreen

- ・オフスクリーンへの描画をサポートする

kCGLRPFullScreen

- ・フルスクリーンへの描画をサポートする

kCGLRPRendererID

- ・レンダラーのID番号

kCGLRPAccelerated

- ・ハードウェアによるアクセラレーションをサポート

kCGLRPRobust

- ・ビデオカードの仕様によりフェイラーモードをサポートしていなし

kCGLRPBackingStore

- ・バックカラーバッファをサポートする

#### kCGLRPMPSafe

- ・レンダラーはスレッドセーフ（すべてのレンダラーはスレッドセーフ）

#### kCGLRPWindow

- ・ウィンドウへの描画をサポート

#### kCGLRPMultiScreen

- ・マルチディスプレイへの描画をサポート

#### kCGLRPCompliant

- ・レンダラーはOpenGL準拠

#### kCGLRPDisplayMask

- ・レンダラーが処理しているディスプレイのビットマスク（CGDirectDisplay.h参照）

#### kCGLRPBufferModes

- ・フレームバッファの種類（Mono、Stereo、Single、DubleのビットORで示す）

#### kCGLRPColorModes

- ・カラーモード（各フラグのビットをORした値）

#### kCGLRPAccumModes

- ・アキュムレータモード（カラーモードと同等）

#### kCGLRPDepthModes

- ・デプスモード（各フラグのビットをORした値）

#### kCGLRPStencilModes

- ・ステンシルモード（デプスモード同等）

kCGLRPMaxAuxBuffers

- ・レンダラーがサポートする予備バッファの最大数

kCGLRPMaxSampleBuffers

- ・レンダラーがサポートするサンプルバッファの最大数

kCGLRPMaxSamples

- ・ひとつのサンプルバッファにおけるサンプルの最大数

kCGLRPSampleModes

- ・サンプルモード（ビットフィールド）

kCGLRPSampleAlpha

- ・アルファサンプリングをサポートする

kCGLRPVideoMemory

- ・レンダラーがサポートする物理的なビデオメモリの総数（単位はバイト）

kCGLRPTextureMemory

- ・レンダラーがサポートする物理的なテクスチャメモリの総数（単位はバイト）

kCGLRPRendererCount

- ・レンダラーの個数を得る

## (4) コンテキストの操作

### ・ CGLCreateContext

コンテキストオブジェクトを作成する (Mac OS X 10.0以降)

```
CLError CGLCreateContext(CGLPixelFormatObj pix, CGLContextObj share,  
CGLContextObj *ctx);
```

pix

- ・ 新規コンテキストオブジェクトのためのピクセルフォーマット

share

- ・ OpenGLオブジェクトを共有するコンテキストオブジェクト (無い場合はNULL)

ctx

- ・ 作成されやコンテキストオブジェクトが返る (失敗した場合にはNULLが返る)

### ・ CGLDestroyContext

コンテキストオブジェクトを破棄する (Mac OS X 10.0以降)

```
CLError CGLDestroyContext(CGLContextObj ctx);
```

ctx

- ・ 破棄の対象となるコンテキストオブジェクト

### ・ CGLCopyContext

コンテキストオブジェクトの選択された状態パラメータだけをコピーする (Mac OS X 10.0以降)

`CLError CGLCopyContext(CGLContextObj src, CGLContextObj dst, unsigned long mask);`

`src`

- ・ コピー元のコンテキストオブジェクト

`dst`

- ・ コピー先のコンテキストオブジェクト

`mask`

- ・ どの状態パラメータをコピー対象とするか決定するマスク（ビットフィールド）

`mask`は、OpenGLの`glPushAttrib()`コマンドの扱う値と同じで、すべての状態パラメータをコピーする場合には`GL_ALL_ATTRIB_BITS`を渡す（詳しくはOpenGLの仕様を別途参照）。

## ・ CGLSetCurrentContext

指定したコンテキストオブジェクトをカレントに設定する（Mac OS X 10.0以降）

`CLError CGLSetCurrentContext(CGLContextObj ctx);`

`ctx`

- ・ カレントに設定するコンテキストオブジェクト

## ・ CGLGetCurrentContext

カレントコンテキストオブジェクトを得る（Mac OS X 10.0以降）

`CGLContextObj CGLGetCurrentContext(void);`

カレントコンテキストオブジェクトが設定されていない場合にはNULLが返る。



## ・ CGLEnable

コンテキストオブジェクトのオプションをONにする (Mac OS X 10.0以降)

```
CGLError CGLEnable(CGLContextObj ctx, CGLContextEnable pname);
```

ctx

- ・ 対象となるコンテキストオブジェクト

pname

- ・ ONにしたいオプションの種類

(a) コンテキストオプションの種類は以下の通り

kCGLCESwapRectangle

- ・ CGLFlushDrawable()を実行した時のスワップ用矩形の有効をON/OFFする

kCGLCERasterization

- ・ 2Dや3DプリミティブのラスタライゼーションをON/OFFする (デバッグ用)

kCGLCEStateValidation

- ・ CGLSetVirtualScreen()が実行されるたびにコンテキストの状態をチェックする

## ・ CGLDisable

コンテキストオブジェクトのオプションをOFFにする (Mac OS X 10.0以降)

```
CGLError CGLDisable(CGLContextObj ctx, CGLContextEnable pname);
```

ctx

- ・ 対象となるコンテキストオブジェクト

pname

- ・ ONにしたいオプションの種類
- ・ CGLIsEnabled

コンテキストオブジェクトのオプションがON/OFFどちらかなのかを確認する (Mac OS X 10.0以降)

```
CGLError CGLIsEnabled(CGLContextObj ctx, CGLContextEnable pname,  
long *enable);
```

ctx

- ・ 対象となるコンテキストオブジェクト

pname

- ・ ON/OFFを確認したいオプションの種類

enable

- ・ もし指定のオプションがONであればTRUEが返る

- ・ CGLSetParameter

指定したコンテキストオブジェクトにパラメータ値を設定する (Mac OS X 10.0以降)

```
CGLError CGLSetParameter(CGLContextObj ctx, CGLContextParameter pname,  
const long *params);
```

ctx

- ・ 対象となるコンテキストオブジェクト

pname

- ・ 設定するコンテキストパラメータ

params

- ・ 設定するパラメータ

(a) コンテキストパラメータの種類は以下の通り

kCGLCPSSwapRectangle

- ・ スワップ用矩形をON/OFFする (4つのパラメータは矩形の左、下、幅、高さ)

kCGLCPSSwapInterval

- ・ バッファスワップの同期の値を設定する (ゼロで同期しない、1で同期する)

kCGLCPClientStorage

- ・ コンテキストに関するアプリケーション独自の32bit値 (ポインタなど) を保存できる
- ・ CGLGetParameter

指定したコンテキストオブジェクトからコンテキストパラメータの値を得る (Mac OS X 10.0以降)

```
CLError CGLSetParameter(CGLContextObj ctx, CGLContextParameter pname,
const long *params);
```

ctx

- ・ 対象となるコンテキストオブジェクト

pname

- ・ 得たいコンテキストパラメータ

params

- ・ 返されたパラメータ

## (5) 描画対象オブジェクトの操作

- ・ CGLSetOffScreen

描画対象オブジェクトとしてオフスクリーン・メモリバッファを指定する (Mac OS X 10.0以降)

```
CGLError CGLSetOffScreen(CGLContextObj ctx, long width, long height,  
long rowbytes, void *baseaddr);
```

ctx

- ・ 対象となるコンテキストオブジェクト

width

- ・ オフスクリーンの幅 (単位はピクセル)

height

- ・ オフスクリーンの高さ (単位はピクセル)

rowbytes

- ・ メモリバッファの横一列のバイト数

baseaddr

- ・ 確保されたメモリバッファのベースアドレス

CGLChoosePixelFormat()でピクセルフォーマットオブジェクトを作成する時にアトリビュートとしてkCGLPFAOffScreenを渡しておく必要がある。

## ・ CGLGetOffScreen

描画対象オブジェクトとして指定されているオフスクリーン・メモリバッファの情報を返す (Mac OS X 10.0以降)

```
CGLError CGLGetOffScreen(CGLContextObj ctx, long *width, long *height,  
long *rowbytes, void **baseaddr);
```

ctx

- ・ 対象となるコンテキストオブジェクト

width

- ・ オフスクリーンの幅 (単位はピクセル)

height

- ・ オフスクリーンの高さ (単位はピクセル)

rowbytes

- ・ メモリバッファの横一列のバイト数

baseaddr

- ・ 保されたメモリバッファのベースアドレス

## ・ CGLSetFullScreen

描画対象オブジェクトとしてフルスクリーン・グラフィックデバイスを指定する (Mac OS X 10.0以降)

```
CGLError CGLSetFullScreen(CGLContextObj ctx);
```

ctx

- ・ 対象となるコンテキストオブジェクト

CGLChoosePixelFormat()でピクセルフォーマットを作成する時には、アトリビュートとしてkCGLPFAFullScreenを渡しておく必要がある。ソフトウェアレンダラーなどフルスクリーンをサポートしていないレンダラーの場合には、CGLChoosePixelFormat()でNULLが返されるので利用付加を判断することが可能である。

- CGLClearDrawable

指定されたコンテキストオブジェクトから設定されている描画対象オブジェクトを外す (Mac OS X 10.0以降)

```
CLError CGLClearDrawable(CGLContextObj ctx);
```

ctx

- 対象となるコンテキストオブジェクト

- CGLFlushDrawable

ダブルバッファのバックからフロントへ画像をコピーする (Mac OS X 10.0以降)

```
CLError CGLFlushDrawable(CGLContextObj ctx);
```

ctx

- 対象となるコンテキストオブジェクト

ダブルバッファを用いるには、CGLChoosePixelFormat()でピクセルフォーマットを作成する時にアトリビュートとしてkCGLPFADoubleBufferを渡しておく必要がある。

## (6) 仮想スクリーンの管理

- CGLSetVirtualScreen

コンテキストオブジェクトに仮想スクリーンを設定する (Mac OS X 10.0以降)

```
CLError CGLSetVirtualScreen(CGLContextObj ctx, long screen);
```

ctx

- ・対象となるコンテキストオブジェクト

screen

- ・設定する仮想スクリーンの番号

もしシングルディスプレイを利用していればゼロが返る (-1が返るとエラー)

### ・ CGLGetVirtualScreen

コンテキストオブジェクトに関連した仮想スクリーンを得る (Mac OS X 10.0以降)

```
CLError CGLGetVirtualScreen(CGLContextObj ctx, long *screen);
```

ctx

- ・対象となるコンテキストオブジェクト

screen

- ・仮想スクリーンの番号を得る

通常この操作は、aglSetDrawable()やaglUpdateContext()で自動に実行される。

## (7) ライブラリーのグローバルオプション

### ・ CGLSetOption

グローバルオプションの値を設定する (Mac OS X 10.0以降)

```
CLError CGLSetOption(CGLGlobalOption pname, long param);
```

pname

- ・グローバルオプションの種類

param

- ・グローバルオプションの値

(a) グローバルオプションの種類は以下の通り

kCGLGOFormatCacheSize

- ・ピクセルフォーマットのキャッシュサイズ指定 (デフォルトは5)

kCGLGOClearFormatCache

- ・ピクセルフォーマットのキャッシュクリア指定 (1でメモリ使用量最低)

kCGLGORetainRenderers

- ・レンダラーをメモリに常駐させるかどうかを指定 (ゼロ以外で常駐、ゼロでノーマル)

kCGLGOResetLibrary

- ・TRUEでCGLライブラリーは初期状態にリセットされる

kCGLGOUseErrorHandler

- ・TRUEでCGLのエラーはCore Graphics APIへ引き継がれる

- ・ CGLGetOption

グローバルオプションの値を得る (Mac OS X 10.0以降)

```
CGLError CGLGetOption(CGLGlobalOption pname, long *param);
```

pname

- ・グローバルオプションの種類



param

- ・ グローバルオプションの値が返る

## ・ CGLGetVersion

CGLライブラリのメジャーとマイナーバージョン番号を得る (Mac OS X 10.0以降)

```
void CGLGetVersion(long *majorvers, long *minorvers);
```

majorvers

- ・ メジャーバージョン番号

minorvers

- ・ マイナーバージョン番号

## ・ CGLErrorString

CGLエラーコードに準拠した文字列を返す (Mac OS X 10.0以降)

```
const char *CGLErrorString(CGLEError error);
```

error

- ・ CGLエラーコード

---

本ドキュメントの履歴

オリジナル2005年11月9日 要約2005年12月15日 v1.00