

# **PlayStation®Edge DXT**

## **ライブラリ リファレンス**

# 目次

---

はじめに .....	3
このドキュメントについて .....	4
<b>SPU関数</b> .....	<b>5</b>
edgeDxtCompress1 .....	6
edgeDxtCompress1a .....	7
edgeDxtCompress3 .....	8
edgeDxtCompress5 .....	9
edgeDxtDecompress1 .....	10
edgeDxtDecompress3 .....	11
edgeDxtDecompress5 .....	12

# はじめに

## このドキュメントについて

### 目的

このドキュメントは、Edge ライブラリの Edge DXT コンポーネントの API リファレンスです。このコンポーネントは、SPU 上でテクスチャデータを効率的に RSX®圧縮形式に圧縮するために使われます。

### 対象読者と前提条件

このドキュメントは、PlayStation®3 用の高性能アプリケーションを書こうとしている PlayStation®3 デベロッパのため書かれたものです。デベロッパは、以下のような対象を熟知していることを前提にしています。

- C と C++
- PlayStation®3 のハードウェア
- SCE の標準ライブラリ関数

### 関連ドキュメント

このリファレンスと、以下のドキュメントを併用することにより、Edge ライブラリの使用法やリファレンスについての完全な情報を得ることができます。

- PlayStation® Edge ライブラリ概要
- PlayStation® Edge ジオメトリライブラリ クイックスタートガイド
- PlayStation® Edge ジオメトリライブラリ リファレンス
- PlayStation® Edge オフラインツール用ジオメトリライブラリ リファレンス
- PlayStation® Edge アニメーションライブラリ リファレンス
- PlayStation® Edge オフラインツール用アニメーションライブラリ リファレンス
- PlayStation® Edge Zlib ライブラリ リファレンス
- PlayStation® Edge LZMA リファレンス
- PlayStation® Edge LZ0 ライブラリ リファレンス
- PlayStation® Edge Post ライブラリ リファレンス

### 表記法

このドキュメントでは、以下のような表記法を使います。

記法	意味
等幅フォント	プログラミングコードおよびリテラル (処理命令、レジスタ名、データ型、イベント、ファイル名など) を表します。また、関数、構造体、マクロなどの名前を表すこともあります。
等幅フォント+太字	構造体や関数の定義の中でのみ、構造体や関数の名前を示します。
等幅フォント+斜体	引数、パラメータ、変数を表します。
<a href="#">青字 + 下線</a>	ハイパーリンクを表します (青色で表示されるのは、カラープリンタもしくははオンラインの場合だけです)。

# SPU関数

# edgeDxtCompress1

ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 フォーマットに圧縮する

## 定 義

```
#include <edge/dxt/edgedxt.h>
void edgeDxtCompress1
(
    void* outputBlocks,
    const void* inputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

<i>outputBlocks</i>	圧縮後の出力データが書き込まれるローカルストアのポインタ 8 バイト境界にアラインメントされている必要がある
<i>inputPixels</i>	入力ピクセルデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	入力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	圧縮するピクセルデータ中の 4x4 のブロック数

## 解 説

この関数は、ローカルストア (LS) の中で、ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 テクスチャフォーマットに圧縮します。

*inputPixels* の色成分の順序は、ARGB であると仮定されます。たとえば、CELL\_GCM\_SURFACE\_A8R8G8B8 型のタイリングされていない表面データは、ローカルストアに移動したら、直接圧縮することができます。それ以外のフォーマットは、圧縮の前に、ARGB の順序に並べ替える必要があります。

この関数は、圧縮速度を高速に保つために、常に 4 色の DXT1 エンコーディングを使います。したがって、入力ピクセル中のアルファデータはすべて無視されます。出力ブロック中のアルファ値は常に 255 の定数値になります。アルファを維持する必要があるピクセルデータの場合には、[edgeDxtCompress1a](#)、[edgeDxtCompress3](#)、[edgeDxtCompress5](#) の各圧縮関数を利用してください。

この関数は、*outputBlocks* の LS アドレスに、正確に  $8 * blockCount$  バイトのテクスチャデータを書き込みます。

## 注 意

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

# edgeDxtCompress1a

ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 フォーマットに圧縮する

## 定 義

```
#include <edge/dxt/edgedxt.h>
void edgeDxtCompress1a
(
    void*                outputBlocks,
    const void*          inputPixels,
    uint32_t             rowStride,
    uint32_t             blockCount
);
```

## 引 数

<i>outputBlocks</i>	圧縮後の出力データが書き込まれるローカルストアのポインタ 8 バイト境界にアラインメントされている必要がある
<i>inputPixels</i>	入力ピクセルデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	入力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	圧縮するピクセルデータ中の 4x4 のブロック数

## 説 明

この関数は、ローカルストア (LS) の中で、ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 テクスチャフォーマットに圧縮します。

*inputPixels* の色成分の順序は ARGB と仮定されます。たとえば、CELL\_GCM\_SURFACE\_A8R8G8B8 型のタイリングされていない表面データは、ローカルストアに移動したら、直接圧縮することができます。それ以外のフォーマットは、圧縮の前に、ARGB の順序に並べ替える必要があります。

この関数は、アルファが 1 ビットの出力画像を生成します。出力アルファ値は、入力アルファ値が 0x80 以上の場合には、255 に設定され、それ以外の場合には、0 に設定されます。

この関数は、*outputBlocks* の LS アドレスに、正確に  $8 * blockCount$  バイトのテクスチャデータを書き込みます。

## 注 意

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

# edgeDxtCompress3

ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT23 フォーマットに圧縮する

## 定 義

```
#include <edge/dxt/edgedxt.h>
void edgeDxtCompress3
(
    void* outputBlocks,
    const void* inputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

<i>outputBlocks</i>	圧縮後の出力データが書き込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>inputPixels</i>	入力ピクセルデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	入力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	圧縮するピクセルデータ中の 4x4 のブロック数

## 解 説

この関数は、ローカルストアの中で、ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT23 テクスチャフォーマットに圧縮します。

*inputPixels* の色成分の順序は、ARGB と仮定されます。たとえば、CELL\_GCM\_SURFACE\_A8R8G8B8 型のタイリングされていない表面データは、ローカルストアに移動したら、直接圧縮することができます。それ以外のフォーマットは、圧縮の前に、ARGB の順序に並べ替える必要があります。

この関数は、*outputBlocks* の LS アドレスに、正確に  $16 * blockCount$  バイトのテクスチャデータを書き込みます。

## 注 意

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。



---

# edgeDxtCompress5

---

ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT45 フォーマットに圧縮する

## 定 義

---

```
#include <edge/dxt/edgedxt.h>
void edgeDxtCompress5
(
    void* outputBlocks,
    const void* inputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

---

<i>outputBlocks</i>	圧縮後の出力データが書き込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>inputPixels</i>	入力ピクセルデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	入力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	圧縮するピクセルデータ中の 4×4 のブロック数

## 解 説

---

この関数は、ローカルストアの中で、ピクセルデータを CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT45 テクスチャフォーマットに圧縮します。

*inputPixels* の色成分の順序は、ARGB と仮定されます。たとえば、CELL\_GCM\_SURFACE\_A8R8G8B8 型のタイリングされていない表面データは、ローカルストアに移動したら、直接圧縮することができます。それ以外のフォーマットは、圧縮の前に、ARGB の順序に並べ替える必要があります。

この関数は、*outputBlocks* の LS アドレスに、正確に  $16 * blockCount$  バイトのテクスチャデータを書き込みます。

## 注 意

---

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

---

# edgeDxtDecompress1

---

CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 フォーマットからテクスチャデータを伸張する

## 定 義

---

```
#include <edge/dxt/edgedxt.h>
void edgeDxtDecompress1
(
    const void* inputBlocks,
    void* outputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

---

<i>inputBlocks</i>	圧縮されたテクスチャデータが読み込まれるローカルストアのポインタ 8 バイト境界にアラインメントされている必要がある
<i>outputPixels</i>	出力ピクセルデータが書き込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	出力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	伸張するピクセルデータ中の 4x4 のブロック数

## 解 説

---

この関数は、ローカルストアの中で CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT1 テクスチャフォーマットで格納されたテクスチャデータを伸張します。

*outputPixels* 中の色成分は、ARGB の順序で書き込まれます。

## 注 意

---

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

---

# edgeDxtDecompress3

---

CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT23 フォーマットからテクスチャデータを伸張する

## 定 義

---

```
#include <edge/dxt/edgedxt.h>
void edgeDxtDecompress3
(
    const void* inputBlocks,
    void* outputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

---

<i>inputBlocks</i>	圧縮されたテクスチャデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>outputPixels</i>	出力ピクセルデータが書き込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	出力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	伸張するピクセルデータ中の 4x4 のブロック数

## 解 説

---

この関数は、ローカルストアの中で CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT23 テクスチャフォーマットで格納されたテクスチャデータを伸張します。

*outputPixels* 中の色成分は、ARGB の順序で書き込まれます。

## 注 意

---

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

---

# edgeDxtDecompress5

---

CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT45 フォーマットからテクスチャデータを伸張する

## 定 義

---

```
#include <edge/dxt/edgedxt.h>
void edgeDxtDecompress5
(
    const void* inputBlocks,
    void* outputPixels,
    uint32_t rowStride,
    uint32_t blockCount
);
```

## 引 数

---

<i>inputBlocks</i>	圧縮されたテクスチャデータが読み込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>outputPixels</i>	出力ピクセルデータが書き込まれるローカルストアのポインタ 16 バイト境界にアラインメントされている必要がある
<i>rowStride</i>	出力ピクセルデータの各行の間のバイト数 16 バイト境界にアラインメントされている必要がある
<i>blockCount</i>	伸張するピクセルデータ中の 4x4 のブロック数

## 解 説

---

この関数は、ローカルストアの中で CELL\_GCM\_TEXTURE\_COMPRESSED\_DXT45 テクスチャフォーマットで格納されたテクスチャデータを伸張します。

*outputPixels* 中の色成分は、ARGB の順序で書き込まれます。

## 注 意

---

パラメータアラインメント制限の違反があった場合、そのパラメータの下位ビットは無視されます。

Filename: Edge\_DXT\_Library-Reference\_j.doc  
Directory: C:\Documents and Settings\MTESHIMA\My Documents\Working  
Files\7-19-10\Edge 1.2.0  
Template: C:\projects\sony\templates\libref\_V1.1\_j.dot  
Title: Edge\_DXT\_Library-Reference\_j  
Subject:  
Author: SCE Document Group  
Keywords:  
Comments:  
Creation Date: 7/19/2010 2:05:00 PM  
Change Number: 3  
Last Saved On: 7/19/2010 2:05:00 PM  
Last Saved By: mteshima  
Total Editing Time: 5 Minutes  
Last Printed On: 7/19/2010 2:09:00 PM  
As of Last Complete Printing  
Number of Pages: 12  
Number of Words: 2,917 (approx.)  
Number of Characters: 4,698 (approx.)