

# KATANA

## ミドルウェアライブラリ

### 基本仕様書

1998年06月30日

Ver . 0.50

1998年08月03日

Ver . 0.55

1998年10月02日

Ver . 1.00

作成

セガ・エンタープライゼス

システム研究開発部

CRI ミドルウェアチーム

マルチメディア研究室	'98-10-02	保坂	押見	野沢	Number
	DATE	DSGN	CHKD	APRV	

株式会社 C S K 総合研究所

変 更 履 歴

年月日	バージョン	変 更 内 容
1998.06.30	0.50	新規作成。
1998.08.03	0.55	記述ミスを修正。(mwPlyStartFname の引数)
1998.10.02	1.00	上記記述ミスを再度修正 (mwPlyStartFname,mwPlyGetStat の引数) Wave 再生用関数仕様、サンプルを追加。

# 目 次

1. 概 要 .....	1
1.1 目 的 .....	1
1.2 モジュール構成 .....	1
2. ミドルウェアライブラリの内部構成 .....	2
3. ミドルウェアの使用方法 .....	3
3.1 ミドルウェアライブラリの初期化 .....	3
3.2 ミドルウェアライブラリの使用方法 .....	3
3.3 ミドルウェア再生ハンドル .....	3
3.4 ミドルウェア再生ハンドルの動作状態 .....	4
3.5 MPEG Sofdec の再生 .....	5
3.6 Wave ファイルの再生 .....	6
4. データ仕様 .....	7
4.1 定 数 .....	8
4.2 データ型 .....	9
5. 関数仕様 .....	10
5.1 MPEG Sofdec 用関数 .....	11
5.2 Wave 用関数 .....	16
5.3 共通関数 .....	18

## 1. 概 要

### 1.1 目 的

本ライブラリは、動画や音声を容易に再生するためのライブラリである。下記の形式のファイルを GD-ROM よりストリーム再生できる。

( 1 ) Sofdec データファイル

映像を MPEG/Video、音声を Sofdec/Audio により圧縮しマルチプレックスしたデータ。

( 2 ) WAVE データファイル

WAVE フォーマットの非圧縮の PCM 音声データ。

### 1.2 モジュール構成

ミドルウェアライブラリのモジュール構成図を以下に示す。

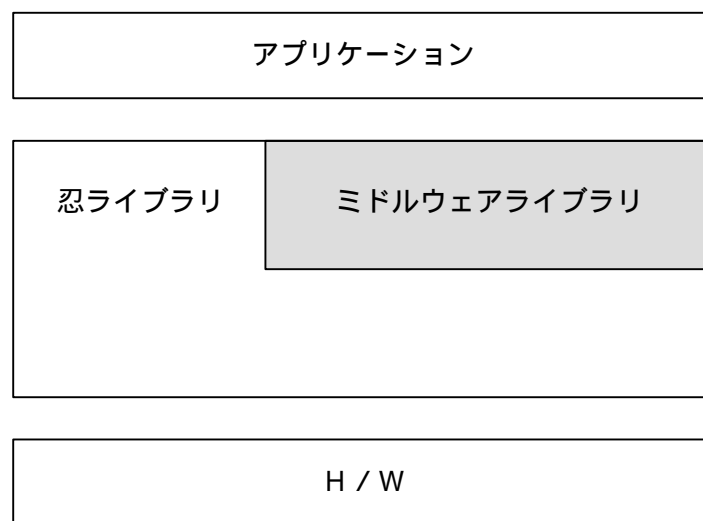


図 1 - 1 モジュール構成

## 2. ミドルウェアライブラリの内部構成

ミドルウェアライブラリは、以下のモジュールから構成されている。アプリケーションは、ミドルウェア API を用いることによって、容易に動画や音声を再生できる。

表 2-1 ミドルウェアライブラリの内部モジュール

モジュール名	説 明
モジュールマネージャ	デコーダに CPU タイムを割り当てる。
ストリームコントローラ	インターリーブされた動画と音声データを読み込む。
ストリームスプリッタ	インターリーブされた動画と音声データを分離し、デコーダに渡す。
ビデオデコーダ	圧縮された動画データを展開し、ビデオレンダラに渡す。
オーディオデコーダ	圧縮された音声データを展開し、オーディオレンダラに渡す。
ビデオレンダラ	展開された動画データを出力する。
オーディオレンダラ	展開された音声データを出力する。

以下に、ミドルウェアライブラリの内部構成図を示す。

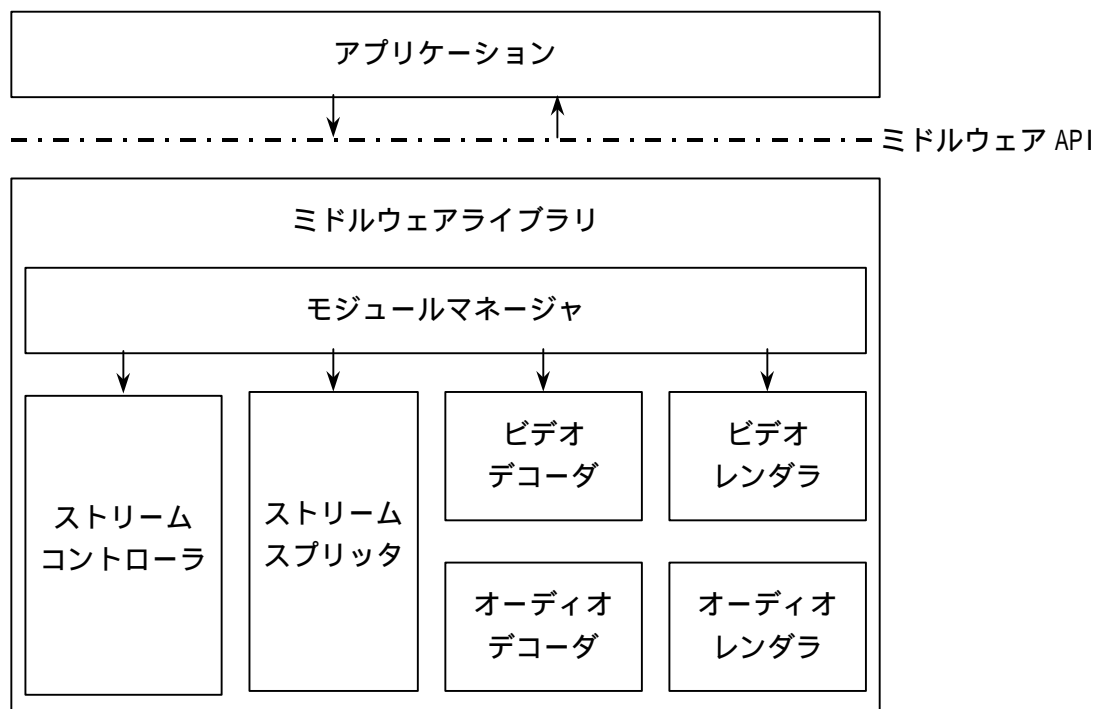


図 2-1 ミドルウェアライブラリの内部構成

### 3. ミドルウェアの使用方法

#### 3.1 ミドルウェアライブラリの初期化

ミドルウェアライブラリの初期化は、各コーデックに対応した初期化関数を呼び出さなければならない。

< Sofdec の場合 >

```
mwPlyPreInitSofdec();  
sbInitSystem(NJD_RESOLUTION_640x480_NTSC1, NJD_FRAMEBUFFER_MODE_ARGB8888, 1);  
mwPlyInitSofdec(NULL);
```

< WAV の場合 >

```
sbInitSystem(NJD_RESOLUTION_640x480_NTSC1, NJD_FRAMEBUFFER_MODE_ARGB8888, 1);  
mwPlyInitWav(MWD_PLY_MN_EXEC_VSYNC);
```

#### 3.2 ミドルウェアライブラリの使用方法

ミドルウェアライブラリの内部状態の更新は、メイン処理用サーバ関数(mwPlyExecServer)を njWaitVSync 関数を呼び出す前に実行する。また、ゲームフレームの開始をミドルウェアライブラリに通知するために、njWaitVSync 関数の直後に mwPlyStartFrame 関数を呼び出さなければならない。

```
while (1) {  
    njWaitVSync();  
    mwStartFrame();      /* ゲームフレームの開始の通知          */  
    :  
    (ゲームの処理)  
    :  
    mwExecServer();      /* デコード処理など          */  
}
```

#### 3.3 ミドルウェア再生ハンドル

各コーデックによって圧縮された音声や動画データは、ミドルウェア再生ハンドルを用いて再生する。まず、各コーデック用のミドルウェアハンドルを生成する。このハンドルの生成には、使用するコーデックに対応した API を用いる。

```
MWPLY plysfd, plywav;
```

```
plysfd = mwPlyCrateSofdec(&cprm);      /* Sofdec データ再生用ミドルウェアハンドル */  
plywav = mwPlyCrateWav(&cprm);         /* WAVE データ再生用ミドルウェアハンドル  */
```

ハンドルの生成には、異なる API を使用するが、再生指示には同じ API を使用することができる。

```
mwPlyStartFname(plysfd, "SAMPLE.SFD"); /* Sofdec データの再生  
*/  
mwPlyStartFname(plywav, "SAMPLE.WAV"); /* WAVE データの再生  
*/
```

mwPlyStart 関数と mwPlyStop 関数により動画や音声の再生開始と再生停止を制御する。

### 3.4 ミドルウェア再生ハンドルの動作状態

ミドルウェアハンドルの動作状態を下記に示す。生成した直後は、STOP 状態となる。再生開始後、状態は、PREP -> PLAYING -> PLAYEND と遷移する。

表 3 - 1 ハンドルの状態

状 態	説 明
STOP	再生が停止している状態
PREP	再生の準備をしている状態
PLAYING	再生中の状態
PLAYEND	再生が終了した状態
ERROR	エラーが発生した状態

状態遷移図を以下に示す。

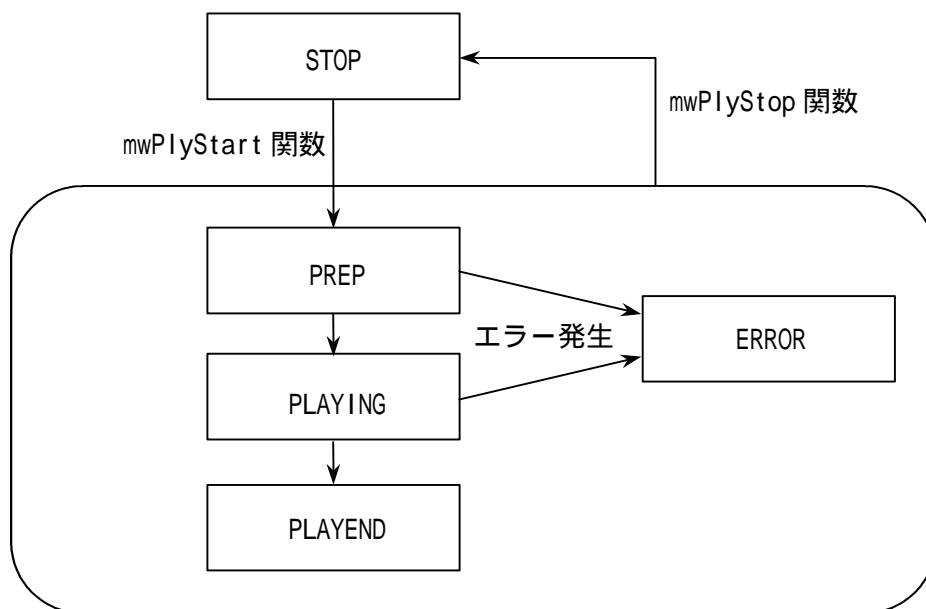


図 3 - 1 状態遷移図

### 3.5 MPEG Sofdec の再生

以下に、MPEG Sofdec を再生するサンプルプログラムを示す。

#### < サンプルプログラム >

/\* アプリケーションメイン関数 \*/

```
void main(void)
{
    MWPLY                ply;                /* ミドルウェア再生ハンドル */
    MWE_PLY_STAT         stat;               /* ハンドルの状態 */
    MWS_PLY_CPRM_SFD     cprm;              /* ハンドル生成のパラメータ構造体 */

    mwPlyPreInitSofdec();                   /* 割り込みスタックなどの設定 */
    sbInitSystem(NJD_RESOLUTION_640x480_NTSC1, NJD_FRAMEBUFFER_MODE_ARGB8888, 1);
    /*
     * 画面やサウンドの初期化
     */
    mwPlyInitSofdec(NULL);                  /* ライブラリの初期化 */
    ply = mwPlyCreateSofdec(&cprm);         /* ハンドルの生成 */
    mwPlyStartFname(ply, "SAMPLE.SFD");     /* 再生開始 */
    for (;;) {
        njWaitVSync();                     /* V-Sync 待ち */
        mwPlyStartFrame();                 /* ゲームフレーム開始の通知 */
        stat = mwPlyGetStat(ply);          /* ハンドルの状態の取得 */
        if ( stat == MWE_PLY_STAT_PLAYEND ) {
            break;
        }
        mwPlyExecServer();                 /* ミドルウェアライブラリの実行 */
    }
    mwPlyStop(ply);                        /* 再生停止 */
    mwPlyDestroy(ply);                     /* ハンドルの消去 */
    mwPlyFinishSofdec();                   /* ライブラリの初期化 */
}
```

### 3.6 Wave ファイルの再生

以下に、Wave ファイルを再生するサンプルプログラムを示す。

#### < サンプルプログラム >

```
/* アプリケーションメイン関数 */
void main(void)
{
    MWPLY          ply;                /* ミドルウェア再生ハンドル */
    MWE_PLY_STAT   stat;               /* ハンドルの状態 */
    MWS_PLY_CPRM   cprm;               /* ハンドル生成のパラメータ構造体 */

    /*
     * 画面やサウンドの初期化
     */
    mwPlyInitWav();                    /* ライブラリの初期化 */
    ply = mwPlyCreateWav(&cprm);        /* ハンドルの生成 */
    mwPlyStartFname(ply, "SAMPLE.WAV"); /* 再生開始 */
    for (;;) {
        njWaitVSync();                 /* V-Sync 待ち */
        mwPlyStartFrame();              /* ゲームフレーム開始の通知 */
        stat = mwPlyGetStat(ply);       /* ハンドルの状態の取得 */
        if ( stat == MWE_PLY_STAT_PLAYEND ) {
            break;
        }
        mwPlyExecServer();              /* ミドルウェアライブラリの実行 */
    }
    mwPlyStop(ply);                    /* 再生開始 */
    mwPlyDestroy(ply);                 /* ハンドルの消去 */
    mwPlyFinishWav();                  /* ライブラリの初期化 */
    for (;;)
}
```

#### 4. データ仕様

ライブラリのデータ一覧を以下に示す。

表 4 - 1 データ一覧

データ名		機 能	番号
定 数			
	MWE_PLY_STAT_~	ハンドルの状態	1.1
	MWE_PLY_FTYPE~	再生するファイルのタイプ	1.2
	MWE_PLY_DTYPE~	動画の表示タイプ	1.3
データ型			
	MWPLY	ミドルウェア再生ハンドル	2.1
	MWS_PLY_CPRM_xxx	ハンドル生成のパラメータ構造体	2.2

## 4.1 定 数

Title	Data Name	Data	No
データ	MWE_PLY_STAT_~	ハンドルの状態	1.1

以下に示す定数は、ハンドルの状態を示す。

定数名	説 明
MWE_PLY_STAT_STOP	停止中
MWE_PLY_STAT_PREP	準備中
MWE_PLY_STAT_PLAYING	再生中
MWE_PLY_STAT_PLAYEND	再生終了
MWE_PLY_STAT_ERROR	エラー状態

Title	Data Name	Data	No
データ	MWE_PLY_FTYPE_~	再生するファイルのタイプ	1.2

以下に示す定数は、ハンドルの状態を示す。

定数名	説 明
MWE_PLY_FTYPE_SFD	MPEG Sofdec (音 + 絵)
MWE_PLY_FTYPE_MPV	MPEG Sofdec (絵だけ)

Title	Data Name	Data	No
データ	MWE_PLY_DTYPE_~	動画の表示タイプ	1.3

以下に示す定数は、ハンドルの状態を示す。

定数名	説 明
MWE_PLY_DTYPE_AUTO	画面の表示サイズを自動的に決定
MWE_PLY_DTYPE_FULL	全画面に表示されるように調整

## 4.2 データ型

Title	Data Name	Data	No
データ	MWPLY	ミドルウェア再生ハンドル	2.1

動画や音声の再生を制御するためのハンドル。

Title	Data Name	Data	No
データ	MWS_PLY_CPRM_SFD	ハンドル生成のパラメータ構造体	2.2

ミドルウェア再生ハンドル生成する時に、設定するパラメータの構造体。メンバーは以下の通り。

メンバー	型 名	説 明
f type	Sint32	再生するファイルのタイプ。 MWE_PLY_FTYPE_ ~ から選択する。( 1 )
dtype	Sint32	動画の表示タイプ MWE_PLY_DTYPE_ ~ から選択する。

1:Ver.1.00 では MWE\_PLY\_FTYPE\_SFD 以外を選択してはならない。

## 5. 関数仕様

ライブラリの関数一覧を以下に示す。

表 5 - 1 関数一覧

関数名	機 能	番号
MPEG SofdecMPEG Sofdec 用関数		
mwPlyPreInitSofdec	システム初期化の設定	1.1
mwPlyInitSofdec	ライブラリの初期化	1.2
mwPlyFinishSofdec	ライブラリの終了処理	1.3
mwPlyCreateSofdec	ハンドルの生成	1.4
mwPlySetDispPos	表示位置の設定	1.5
mwPlySetDispSize	表示サイズの設定	1.6
mwPlySetBright	輝度の設定	1.7
mwPlyGetBright	輝度の取得	1.8
mwPlySetBrightOfst	輝度オフセットの設定	1.9
mwPlyGetBrightOfst	輝度オフセットの取得	1.10
Wave 用関数		
mwPlyInitWav	ライブラリの初期化	2.1
mwPlyFinishWav	ライブラリの初期化	2.2
mwPlyCreateWav	ハンドルの生成	2.3
共通関数		
mwPlyDestroy	ハンドルの消去	3.1
mwPlyStartFname	再生開始	3.2
mwPlyStop	再生停止	3.3
mwPlyGetStat	ハンドルの状態の取得	3.4
mwPlyGetTime	再生サンプル数の取得	3.5
mwPlyStartFrame	ゲームフレーム開始の通知	3.6
mwPlyExecServer	サーバ関数	3.7
mwPlyEntryErrFunc	エラー発生時に呼ばれる関数の登録	3.8

## 5.1 MPEG Sofdec 用関数

MPEG Sofdec の再生に必要な関数である。

Title 関 数	Function Name	Function	No
	mwPlyPreInitSofdec	システム初期化の設定	1.1

[書 式] void mwPlyPreInitSofdec(void);

[入 力] なし

[出 力] なし

[関数値] なし

[機 能] 割り込みスタックの設定をする。

Title 関 数	Function Name	Function	No
	mwPlyInitSofdec	Sofdec ライブラリの初期化	1.2

[書 式] void mwPlyInitSofdec(MWS\_PLY\_INIT\_SFD \*iprm);

[入 力] iprm: 初期化のパラメータ(通常は NULL を設定する)

[出 力] なし

[関数値] なし

[機 能] ライブラリを初期化する。

各種サーバ関数を設定する。サーバ関数には以下の種類のものがある。

( 1 ) メインサーバ

メイン処理内で、mwPlyExecServer が呼び出された時に実行される。

( 2 ) V-Sync サーバ

V-Sync 割り込み時に実行される。

( 3 ) WaitVsync サーバ

njWaitVsync 内で、次のゲームフレームまでの待ち時間に実行される。

Sofdec ライブラリは、このサーバ内でデコードを実行する。

Title 関 数	Function Name mwPlyFinishSofdec	Function Sofdec ライブラリの終了処理	No 1.2
--------------	------------------------------------	-------------------------------	-----------

[書 式] void mwPlyFinishSofdec(void);

[入 力] なし

[出 力] なし

[関数値] なし

[機 能] ライブラリの終了処理をする。

Title 関 数	Function Name mwPlyCreateSofdec	Function ハンドルの生成	No 1.3
--------------	------------------------------------	---------------------	-----------

[書 式] MWPLY mwPlyCreateSofdec(MWS\_PLY\_CPRM\_SFD \*cprm);

[入 力] cprm : ハンドルの生成パラメータ

[出 力] なし

[関数値] ミドルウェア再生ハンドル

[機 能] ハンドルを生成する。

生成パラメータの dtype を指定することにより、表示領域を制御できる。

< 自動調整の場合 >

dtype = MWD\_PLY\_DTYPE\_AUTO : 表示領域を整数倍になるように自動的に調整する。  
640x480 よりも小さい場合は、表示領域は 1 ドット分小さくなる。320x240 の場合、  
639x479 となる。

< フルスクリーン表示の場合 >

dtype = MWD\_PLY\_DTYPE\_FULL : フルスクリーンで表示する。

自動調整の場合と異なり、表示領域が 1 ドット小さくことはなく、640x480 で表示する。

Title 関 数	Function Name	Function	No
	mwPlySetDispPos	表示位置の設定	1.4

[書 式] void mwPlySetDispPos(MWPLY ply, float lx, float ly);

[入 力] ply: ミドルウェアハンドル

lx : X座標

ly : Y座標

[出 力] なし

[関数値] なし

[機 能] 表示位置を設定する。

Title 関 数	Function Name	Function	No
	mwPlySetDispSize	表示サイズの設定	1.5

[書 式] void mwPlySetDispSize(MWPLY ply, float sx, float sy);

[入 力] ply: ミドルウェアハンドル

sx : X座標

sy : Y座標

[出 力] なし

[関数値] なし

[機 能] 表示サイズを設定する。

Title 関 数	Function Name mwPlySetBright	Function 輝度の設定	No 1.6
--------------	---------------------------------	-------------------	-----------

[書 式] void mwPlySetBright(MWPLY ply, Sint32 val);

[入 力] ply: ミドルウェアハンドル

val: 輝度 (0 ~ 255)

[出 力] なし

[関数値] なし

[機 能] 輝度を設定する。デフォルトでは、192 が設定されている。

Title 関 数	Function Name mwPlyGetBright	Function 輝度の取得	No 1.7
--------------	---------------------------------	-------------------	-----------

[書 式] Sint32 mwPlyGetBright(MWPLY ply);

[入 力] ply: ミドルウェアハンドル

[出 力] なし

[関数値] 輝度

[機 能] 輝度を取得する。

Title	Function Name	Function	No
関 数	mwPlySetBrightOfst	輝度オフセットの設定	1.8

[書 式] void mwPlySetBrightOfst(MWPLY ply, Sint32 val);

[入 力] ply: ミドルウェアハンドル  
val: 輝度オフセット (0 ~ 255)

[出 力] なし

[関数値] なし

[機 能] 輝度オフセットを設定する。デフォルトでは、0 が設定されている。  
この値を 255 ~ 0 まで減少させることにより、フェードインすることができる。  
また、反対に 0 ~ 255 まで増加させることにより、フェードアウトすることができる。

Title	Function Name	Function	No
関 数	mwPlyGetBrightOfst	輝度オフセットの取得	1.9

[書 式] Sint32 mwPlyGetBrightOfst(MWPLY ply);

[入 力] ply: ミドルウェアハンドル

[出 力] なし

[関数値] 輝度オフセット

[機 能] 輝度オフセットを取得する。

## 5.2 Wave 用関数

Wave の再生に必要な関数である。

Title 関 数	Function Name	Function	No
	mwPlyInitWav	ライブラリの初期化	2.1

[ 書 式 ] void mwPlyInitWav(Sint32 mode);

[ 入 力 ] mode : サーバ関数呼び出しモード

MWD\_PLY\_AT\_EXEC\_VSYNC : V-Sync 割込みないで、サーバ関数を実行する。

MWD\_PLY\_MN\_EXEC\_VSYNC : アプリケーション側でサーバ関数を呼び出して、サーバ処理を実行する。

[ 出 力 ] なし

[ 関数値 ] なし

[ 機 能 ] ライブラリを初期化する。

: Ver.1.00 では、Vsync 関数の V-Sync 割込み登録をしていないので、ユーザプログラムで Vsync 関数を呼ぶ必要有り。

Title 関 数	Function Name	Function	No
	mwPlyFinishWav	ライブラリの終了処理	2.2

[ 書 式 ] void mwPlyFinishWav(void);

[ 入 力 ] なし

[ 出 力 ] なし

[ 関数値 ] なし

[ 機 能 ] ライブラリの終了処理をする。

Title 関 数	Function Name mwPlyCreateWav	Function ハンドルの生成	No 2.3
--------------	---------------------------------	---------------------	-----------

[ 書 式 ] MWPLY mwPlyCreateWav(MWS\_PLY\_CPRM \*cprm);

[ 入 力 ] cprm : ハンドル生成のパラメータ

[ 出 力 ] なし

[ 関数値 ] ミドルウェア再生ハンドル

[ 機 能 ] ハンドルを生成する。

: Ver.1.00 では、cprm には NULL を指定すること。

### 5.3 共通関数

コーデックの種類に関わらず共通な関数群である。

Title	Function Name	Function	No
関 数	mwPlyDestroy	ハンドルの消去	3.1

[ 書 式 ] void mwPlyDestroy(MWPLY ply);

[ 入 力 ] Ply : ミドルウェア再生ハンドル

[ 出 力 ] なし

[ 関数値 ] なし

[ 機 能 ] ハンドルを消去する。

Title	Function Name	Function	No
関 数	mwPlyStartFname	再生開始	3.2

[ 書 式 ] void mwPlyStartFname(MWPLY ply, Sint8 \*fname);

[ 入 力 ] ply : ミドルウェア再生ハンドル

fname : 再生する動画や音声のファイル名

[ 出 力 ] なし

[ 関数値 ] なし

[ 機 能 ] 動画や音声の再生を開始する。

Title 関 数	Function Name mwPlyStop	Function 再生停止	No 3.3
--------------	----------------------------	------------------	-----------

[書 式] void mwPlyStop(MWPLY ply);  
[入 力] ply : ミドルウェア再生ハンドル  
[出 力] なし  
[関数値] なし  
[機 能] 動画や音声の再生を停止する。

Title 関 数	Function Name mwPlyGetStat	Function ハンドルの状態の取得	No 3.4
--------------	-------------------------------	------------------------	-----------

[書 式] MWE\_PLY\_STAT mwPlyGetStat(MWPLY ply);  
[入 力] ply : ミドルウェア再生ハンドル  
[出 力] なし  
[関数値] ハンドルの内部状態  
[機 能] ミドルウェア再生ハンドルの内部状態を取得する。

定数名	説 明
MWE_PLY_STAT_STOP	停止中
MWE_PLY_STAT_PREP	準備中
MWE_PLY_STAT_PLAYING	再生中
MWE_PLY_STAT_PLAYEND	再生終了
MWE_PLY_STAT_ERROR	エラー状態

Title 関 数	Function Name	Function	No
	mwPlyGetTime	再生サンプル数の取得	3.5

[ 書 式 ] void mwPlyGetTime(MWPLY ply, Sint32 \*ncount, Sint32 \*tscale);

[ 入 力 ] ply : ミドルウェア再生ハンドル

[ 出 力 ] ncount : 再生サンプル数(時刻)

tscale : サンプリング周波数(時刻の単位)(単位 : ヘルツ)

[ 関数値 ] なし

[ 機 能 ] サンプル単位で再生時刻を取得する。

[ 備 考 ]

(a) 実時間(runtime)は、以下の式で計算される。

$runtime = ncount / tscale;$

Title 関 数	Function Name	Function	No
	mwPlyStartFrame	ゲームフレームの開始通知	3.6

[ 書 式 ] void mwPlyStartFrame(void);

[ 入 力 ] なし

[ 出 力 ] なし

[ 関数値 ] なし

[ 機 能 ] ゲームフレームが開始したことを通知する。

Title 関 数	Function Name mwPlyExecServer	Function サーバ関数	No 3.7
--------------	----------------------------------	-------------------	-----------

[書 式] void mwPlyExecServer(void);  
[入 力] なし  
[出 力] なし  
[関数値] なし  
[機 能] ハンドルの内部状態を更新する。

Title 関 数	Function Name mwPlyEntryErrFunc	Function エラー発生時に呼ばれる関数の登録	No 3.8
--------------	------------------------------------	------------------------------	-----------

[書 式] void mwPlyEntryErrFunc(MW\_PLY\_ERRFN errfn, void \*obj);  
[入 力] errfn :エラー関数  
obj :エラーオブジェクト  
[出 力]  
[関数値] なし  
[機 能] エラー発生時に呼ばれる関数を登録する。  
[備 考] エラー関数を登録すると、エラーが発生と共に、登録した関数が呼び出される。  
エラー関数は、第2引数に文字列が渡される。この文字列によりエラーの内容が確認できる。

```
/* ユーザエラー関数 */
void user_error(void *user_obj, char *msg)
{
    for (;;) {
        njPrintC(NJM_LOCATION(3, 3), msg);
    }
}

void main(void)
{
    mwPlyEntryErrFunc(user_error, user_obj);
}
```