

# Parallel World ライブラリ 仕様書

## 目次

Parallel World ライブラリ .....	4
Parallel World ライブラリとは .....	4
特徴.....	4
Parallel World ライブラリの概要.....	4
制約事項.....	4
プロジェクト関連.....	5
PWLibCreateProject .....	5
PWLibDeleteProject.....	5
PWLibLoadWaveFile .....	6
PWLibUnloadWaveFile.....	7
PWLibLoadUstFile.....	8
PWLibUnloadUstFile.....	9
PWLibSetProjectPrm .....	10
PWLibGetProjectPrm .....	11
PWLibSetVoiceTrackPrm .....	12
PWLibGetVoiceTrackPrm .....	13
PWLibSetAudioTrackPrm .....	14
PWLibGetAudioTrackPrm.....	15
トラック関連 .....	16
PWLibGetVoiceTrackHandle.....	16
PWLibSetMorphingPrefix .....	17
PWLibConvertVoiceTrack.....	18
周波数表関連 .....	19
PWLibStartCreatingFrqFile .....	19
PWLibWaitCreatingFrqFile .....	19
PWLibStopCreatingFrqFile .....	20
音声合成関連 .....	21
PWLibOpenWaveOut.....	21
PWLibCloseWaveOut.....	21
PWLibWoPlay.....	22
PWLibWoStop.....	22
PWLibWoSeek .....	23
PWLibWoGetStatus .....	24
PWLibWoGetPosition.....	25
PWLibGetProjectLength.....	26
PWLibExportWaveFile .....	27
PWLibSynthesize .....	28
エラーコード .....	29

エラーコード一覧 .....	29
----------------	----

# Parallel World ライブラリ

## Parallel World ライブラリとは

Parallel World の音声合成機能を提供するライブラリです。本ライブラリを使用することで、Parallel World が持つ音声合成機能を利用したソフトウェアを開発することができます。

## 特徴

- ・ UTAU 音源を使用して、リアルタイムに音声を合成することができます。
- ・ ボイストラック、オーディオトラックのミキシングに対応しています。
- ・ 表情音源モーフィングに対応しています。

## Parallel World ライブラリの概要

プロジェクトは 4 つのボイストラックと 8 つのオーディオトラックから構成されます。ボイストラックには 1 つの UST ファイルを、オーディオトラックには 1 つの Wave ファイルをロードすることができます。各トラックに対し、トラックパラメータ (ボリューム、ミュート、ソロ、オフセット) を設定することができます。

UST ファイルのロード後、再生または Wave ファイルを出力する前に、PWLibConvertVoiceTrack 関数を使用してトラックの変換処理を行ってください。(変換処理では、ピッチ等のパラメータのレンダリングや、連続音化などを行っています。)

## 制約事項

現在のバージョンには、以下の制約があります。

- ・ トラックパラメータの offset には対応していません。
- ・ Wave ファイル出力時のフォーマットは、16bit ステレオのみ対応しています。
- ・ 複数のプロジェクトから同時に Wave 出力デバイスを使用できません。
- ・ Unicode には対応していません。

## プロジェクト関連

### PWLibCreateProject

```
int PWLibCreateProject(  
    PWLIB_HPROJECT *hProject    // プロジェクトハンドルが格納される変数へのポインタ  
)
```

#### パラメータ

hProject

プロジェクトハンドルが格納される変数へのポインタを指定します。

#### 戻り値

エラーコード

#### 解説

新規プロジェクトを作成します。hProject にプロジェクトハンドルが格納されます。

PWLibCreateProject 関数で作成したプロジェクトは、PWLibDeleteProject 関数で破棄してください。

### PWLibDeleteProject

```
int PWLibDeleteProject(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

#### 戻り値

エラーコード

#### 解説

プロジェクトを破棄します。

PWLibCreateProject 関数で作成したプロジェクトは、PWLibDeleteProject 関数で破棄してください。

**PWLibLoadWaveFile**

```
int PWLibLoadWaveFile(  
    PWLIB_HPROJECT hProject,      // プロジェクトハンドル  
    int trackNum,                 // オーディオトラック番号  
    const TCHAR *fileName         // ファイル名  
)
```

**パラメータ**

hProject

プロジェクトハンドルを指定します。

trackNum

オーディオトラック番号 (0-7) を指定します。

fileName

Wave ファイル名を指定します。

**戻り値**

エラーコード

**解説**

オーディオトラックに Wave ファイルをロードします。

**PWLibUnloadWaveFile**

```
int PWLibUnloadWaveFile(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int trackNum                // オーディオトラック番号  
)
```

**パラメータ**

hProject

プロジェクトハンドルを指定します。

trackNum

オーディオトラック番号 (0-7) を指定します。

**戻り値**

エラーコード

**解説**

オーディオトラックにロードされた Wave ファイルをアンロードします。

**PWLibLoadUstFile**

```
int PWLibLoadUstFile(  
    PWLIB_HPROJECT hProject,          // プロジェクトハンドル  
    int trackNum,                     // ボイストラック番号  
    const TCHAR *filename,            // ファイル名  
    const TCHAR *voiceRootPath        // Voice Root パス  
)
```

**パラメータ**

hProject

プロジェクトハンドルを指定します。

trackNum

ボイストラック番号 (0-3) を指定します。

fileName

UST ファイル名を指定します。

voiceRootPath

Voice Root パスを指定します。

**戻り値**

エラーコード

**解説**

ボイストラックに UST ファイルをロードします。

voiceRootPath には空文字を指定することができます。空文字を指定した場合、ライブラリが自動的にパスを取得します。



**PWLibUnloadUstFile**

```
int PWLibUnloadUstFile(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int trackNum                // ボイストラック番号  
)
```

**パラメータ**

**hProject**

プロジェクトハンドルを指定します。

**trackNum**

ボイストラック番号 (0-3) を指定します。

**戻り値**

エラーコード

**解説**

ボイストラックにロードされた UST ファイルをアンロードします。

**PWLibSetProjectPrm**

```
int PWLibSetProjectPrm(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int prmNum,                  // パラメータ番号  
    double prm                   // パラメータの値  
)
```

**パラメータ**

**hProject**

プロジェクトハンドルを指定します。

**prmNum**

パラメータ番号を指定します。

**prm**

パラメータの値を指定します。

**戻り値**

エラーコード

**解説**

プロジェクトのパラメータを設定します。設定可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	－	1.0

**PWLibGetProjectPrm**

```
int PWLibGetProjectPrm(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int prmNum,                  // パラメータ番号  
    double *prm                  // パラメータの値  
)
```

**パラメータ**

**hProject**

プロジェクトハンドルを指定します。

**prmNum**

パラメータ番号を指定します。

**prm**

パラメータの値を格納する変数へのポインタを指定します。

**戻り値**

エラーコード

**解説**

プロジェクトのパラメータを取得します。取得可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	-	1.0

## PWLibSetVoiceTrackPrm

```

int PWLibSetVoiceTrackPrm(
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル
    int trackNum,               // トラック番号
    int prmNum,                 // パラメータ番号
    double prm                  // パラメータの値
)

```

## パラメータ

hProject

プロジェクトハンドルを指定します。

trackNum

トラック番号を指定します。

prmNum

パラメータ番号を指定します。

prm

パラメータの値を指定します。

## 戻り値

エラーコード

## 解説

ボイストラックのパラメータを設定します。設定可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	–	1.0
PWLIB_PRM_PAN	パン	–1.0 (L) ~ 1.0 (R)	0.0
PWLIB_PRM_MUTE_FLAG	ミュートフラグ	0, 1	0
PWLIB_PRM_SOLO_FLAG	ソロフラグ	0, 1	0
PWLIB_PRM_OFFSET	オフセット	–	0

## PWLibGetVoiceTrackPrm

```

int PWLibGetVoiceTrackPrm(
    PWLIB_HPROJECT hProject,      // プロジェクトハンドル
    int trackNum,                 // トラック番号
    int prmNum,                   // パラメータ番号
    double *prm                   // パラメータの値
)

```

## パラメータ

hProject

プロジェクトハンドルを指定します。

trackNum

トラック番号を指定します。

prmNum

パラメータ番号を指定します。

prm

パラメータの値を格納される変数へのポインタを指定します。

## 戻り値

エラーコード

## 解説

ボイストラックのパラメータを取得します。取得可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	–	1.0
PWLIB_PRM_PAN	パン	–1.0 (L) ~ 1.0 (R)	0.0
PWLIB_PRM_MUTE_FLAG	ミュートフラグ	0, 1	0
PWLIB_PRM_SOLO_FLAG	ソロフラグ	0, 1	0
PWLIB_PRM_OFFSET	オフセット	–	0

## PWLibSetAudioTrackPrm

```

int PWLibSetAudioTrackPrm(
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル
    int trackNum,               // トラック番号
    int prmNum,                 // パラメータ番号
    double prm                  // パラメータの値
)

```

## パラメータ

hProject

プロジェクトハンドルを指定します。

trackNum

トラック番号を指定します。

prmNum

パラメータ番号を指定します。

prm

パラメータの値を指定します。

## 戻り値

エラーコード

## 解説

オーディオトラックのパラメータを設定します。設定可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	–	1.0
PWLIB_PRM_PAN	パン	–1.0 (L) ~ 1.0 (R)	0.0
PWLIB_PRM_MUTE_FLAG	ミュートフラグ	0, 1	0
PWLIB_PRM_SOLO_FLAG	ソロフラグ	0, 1	0
PWLIB_PRM_OFFSET	オフセット	–	0

## PWLibGetAudioTrackPrm

```

int PWLibGetAudioTrackPrm(
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル
    int trackNum,               // トラック番号
    int prmNum,                 // パラメータ番号
    double *prm                 // パラメータの値
)

```

## パラメータ

hProject

プロジェクトハンドルを指定します。

trackNum

トラック番号を指定します。

prmNum

パラメータ番号を指定します。

prm

パラメータの値を格納される変数へのポインタを指定します。

## 戻り値

エラーコード

## 解説

オーディオトラックのパラメータを取得します。取得可能なパラメータは以下の通りです。

パラメータ番号	内容	範囲	初期値
PWLIB_PRM_VOLUME	ボリューム[倍]	–	1.0
PWLIB_PRM_PAN	パン	–1.0 (L) ~ 1.0 (R)	0.0
PWLIB_PRM_MUTE_FLAG	ミュートフラグ	0, 1	0
PWLIB_PRM_SOLO_FLAG	ソロフラグ	0, 1	0
PWLIB_PRM_OFFSET	オフセット	–	0

## トラック関連

### PWLibGetVoiceTrackHandle

```
int PWLibGetVoiceTrackHandle(  
    PWLIB_HPROJECT hProject,        // プロジェクトハンドル  
    int trackNum,                   // トラック番号  
    PWLIB_HVOICETRACK *hTrack      // トラックハンドル  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

trackNum

トラック番号を指定します。

hTrack

トラックハンドルが格納される変数へのポインタを指定します。

#### 戻り値

エラーコード

#### 解説

ボイストラックのハンドルを取得します。



PWLibSetMorphingPrefix

```
int PWLibSetMorphingPrefix(  
    PWLIB_HVOICETRACK hTrack,    // トラックハンドル  
    int type,                    // 設定項目  
    const TCHAR *prefix          // prefix または suffix 文字列  
)
```

パラメータ

hTrack  
トラックハンドルを指定します。

type  
設定項目を指定します。以下のいずれかを指定します。

PWLIB_MAIN_PREFIX	メインボイス prefix
PWLIB_MAIN_SUFFIX	メインボイス suffix
PWLIB_MORPHING_PREFIX	モーフィングボイス prefix
PWLIB_MORPHING_SUFFIX	モーフィングボイス suffix

prefix  
prefix または suffix 文字列を指定します。

戻り値  
エラーコード

解説  
ボイストラックのモーフィングに使用する prefix または suffix 文字列を設定します。

**PWLibConvertVoiceTrack**

```
int PWLibConvertVoiceTrack(  
    PWLIB_HVOICETRACK hTrack,    // トラックハンドル  
    unsigned int option          // オプション  
)
```

**パラメータ**

hTrack

トラックハンドルを指定します。

option

オプションを指定します。以下のいずれかを指定します。

PWLIB_CNVOPTION_PIT_MODE1	ピッチ Mode1 でレンダリング
PWLIB_CNVOPTION_PIT_MODE2	ピッチ Mode2 でレンダリング

**戻り値**

エラーコード

**解説**

ボイストラックの音声合成のための準備を行います。

## 周波数表関連

### PWLibStartCreatingFrqFile

```
int PWLibStartCreatingFrqFile(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

#### 戻り値

エラーコード

#### 解説

周波数表の生成を開始します。

### PWLibWaitCreatingFrqFile

```
int PWLibWaitCreatingFrqFile(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

#### 戻り値

エラーコード

#### 解説

周波数表の生成が完了するまで待機します。

**PWLibStopCreatingFrqFile**

```
int PWLibStopCreatingFrqFile(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

**パラメータ**

hProject

プロジェクトハンドルを指定します。

**戻り値**

エラーコード

**解説**

周波数表の生成を中断します。

## 音声合成関連

### PWLibOpenWaveOut

```
int PWLibOpenWaveOut(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int deviceId                // デバイス ID  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

deviceId

デバイス ID を指定します。0 を指定すると既定のデバイスを Open します。

#### 戻り値

エラーコード

#### 解説

Wave 出力デバイスを Open します。

### PWLibCloseWaveOut

```
int PWLibCloseWaveOut(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

#### パラメータ

hProject

プロジェクトハンドルを指定します。

#### 戻り値

エラーコード

#### 解説

Wave 出力デバイスを Close します。

**PWLibWoPlay**

```
int PWLibWoPlay(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

**パラメータ**

hProject  
プロジェクトハンドルを指定します。

**戻り値**

エラーコード

**解説**

再生を開始します。

**PWLibWoStop**

```
int PWLibWoStop(  
    PWLIB_HPROJECT hProject    // プロジェクトハンドル  
)
```

**パラメータ**

hProject  
プロジェクトハンドルを指定します。

**戻り値**

エラーコード

**解説**

再生を停止します。

**PWLibWoSeek**

```
int PWLibWoSeek(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int unit,                    // 単位  
    int offset                    // オフセット  
)
```

**パラメータ****hProject**

プロジェクトハンドルを指定します。

**unit**

オフセットの単位を指定します。以下のいずれかの値を指定します。

PWLIB_UNIT_SAMPLE	サンプル
PWLIB_UNIT_MS	ミリ秒

**offset**

プロジェクト先頭からのオフセットを指定します。

**戻り値**

エラーコード

**解説**

再生位置を変更します。

**PWLibWoGetStatus**

```
int PWLibWoGetStatus(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int *status                // 再生状態  
)
```

**パラメータ**

hProject

プロジェクトハンドルを指定します。

status

再生状態が格納される変数へのポインタを指定します。

**戻り値**

エラーコード

**解説**

再生状態を取得します。再生中の場合は 1 が、停止中の場合は 0 が再生状態に格納されます。



**PWLibWoGetPosition**

```
int PWLibWoGetPosition(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int unit,                  // 単位  
    int *position              // 再生位置  
)
```

**パラメータ**

**hProject**

プロジェクトハンドルを指定します。

**unit**

再生位置の単位を指定します。以下のいずれかの値を指定します。

PWLIB_UNIT_SAMPLE	サンプル
PWLIB_UNIT_MS	ミリ秒

**position**

再生位置が格納される変数へのポインタを指定します。

**戻り値**

エラーコード

**解説**

再生位置を取得します。

**PWLibGetProjectLength**

```
int PWLibGetProjectLength(  
    PWLIB_HPROJECT hProject,    // プロジェクトハンドル  
    int unit,                    // 単位  
    int *length                  // プロジェクト全体の長さ(再生時間)  
)
```

**パラメータ****hProject**

プロジェクトハンドルを指定します。

**unit**

長さの単位を指定します。以下のいずれかの値を指定します。

PWLIB_UNIT_SAMPLE	サンプル
PWLIB_UNIT_MS	ミリ秒

**position**

プロジェクト全体の長さ(再生時間)が格納される変数へのポインタを指定します。

**戻り値**

エラーコード

**解説**

プロジェクト全体の長さ(再生時間)を取得します。

**PWLibExportWaveFile**

```
int PWLibExportWaveFile(  
    PWLIB_HPROJECT hProject,        // プロジェクトハンドル  
    const TCHAR *fileName,          // ファイル名  
    int bitDeapth,                  // ビット深度  
    int nChannel                     // チャンネル数  
)
```

**パラメータ**

**hProject**

プロジェクトハンドルを指定します。

**fileName**

Wave ファイル名を指定します。

**bitDeapth**

ビット深度(16)を指定します。

**nChannel**

チャンネル数(2)を指定します。

**戻り値**

エラーコード

**解説**

Wave ファイルを出力します。

**PWLibSynthesize**

```
int PWLibSynthesize(  
    PWLIB_HPROJECT hProject,          // プロジェクトハンドル  
    int index,                        // 先頭からのオフセット[サンプル]  
    int size,                         // サイズ[サンプル]  
    double *waveData                 // Wave データ (double, ステレオ)  
)
```

**パラメータ****hProject**

プロジェクトハンドルを指定します。

**index**

合成する位置(プロジェクト先頭からのサンプル単位のオフセット)を指定します。

**size**

合成するサイズ(サンプル数)を指定します。

**waveData**

Wave データが格納される変数へのポインタを指定します。

**戻り値**

エラーコード

**解説**

音声合成結果(double, ステレオの Wave データ)を取得します。

# エラーコード

## エラーコード一覧

エラーコード	内容
PWLIB_ERROR_NOERROR	エラーなし
PWLIB_ERROR_ARGUMENT	引数が範囲外です。
PWLIB_ERROR_NO_MEMORY	メモリが不足しています。
PWLIB_ERROR_ALREADY_OPENED	既に音声出力デバイスを Open 済みです。
PWLIB_ERROR_ALREADY_CLOSED	既に音声出力デバイスを Close 済みです。
PWLIB_ERROR_OPEN_VOICE_DIR	音源フォルダが見つかりません。
PWLIB_ERROR_OPEN_OTO_FILE	oto.ini ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_MAP_FILE	map ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_WAV_FILE_W	wave ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_WAV_FILE_R	wave ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_UST_FILE	ust ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_FRQ_FILE_W	frq ファイルの Open に失敗しました。
PWLIB_ERROR_OPEN_FRQ_FILE_R	frq ファイルの Open に失敗しました。
PWLIB_ERROR_PROGRAM_FILES	音源が Program Files フォルダに格納されています。
PWLIB_ERROR_WRITE_WAV_FILE	wave ファイルの書き込みに失敗しました。
PWLIB_ERROR_WRITE_FRQ_FILE	frq ファイルの書き込みに失敗しました。
PWLIB_ERROR_WAV_FORMAT	wave ファイルのフォーマットが不正です。
PWLIB_ERROR_FRQ_FORMAT	frq ファイルのフォーマットが不正です。
PWLIB_ERROR_UST_FORMAT	ust ファイルのフォーマットが不正です。
PWLIB_ERROR_OTO_UTF8	oto.ini が対応していない形式 (UTF-8 形式) です。
PWLIB_ERROR_UST_UTF8	ust ファイルが対応していない形式 (UTF-8 形式) です。
PWLIB_ERROR_APPEND_NUM_OVER	アペンド数が制限を超過しています。
PWLIB_ERROR_WAVE_OUT	音声デバイスへの出力に失敗しました。