

VocalShifter ライブラリ

仕様書

AckieSound

<https://ackiesound.ifdef.jp/>

2023.11.05

目次

VocalShifter ライブライリ	4
VocalShifter ライブライリとは	4
ライブラリ関連	5
VslibGetVersion	5
プロジェクト関連	6
VslibCreateProject	6
VslibOpenProject	6
VslibSaveProject	7
VslibDeleteProject	7
VslibGetProjectInfo	8
VslibSetProjectInfo	8
トラック関連	9
VslibGetTrackMaxNum	9
VslibAddTrack	9
VslibCopyTrack	10
VslibUpTrack	11
VslibDownTrack	11
VslibDeleteTrack	12
VslibGetTrackInfo	13
VslibSetTrackInfo	14
VslibGetTrackInfoEx	15
VslibSetTrackInfoEx	16
アイテム関連	17
VslibGetItemMaxNum	17
VslibAddItem	18
VslibAddItemEx	19
VslibCopyItem	20
VslibDeleteItem	21
VslibGetItemInfo	22
VslibSetItemInfo	23
VslibGetCtrlPntInfoEx	24
VslibSetCtrlPntInfoEx	25
VslibGetCtrlPntInfoEx2	26
VslibSetCtrlPntInfoEx2	27
VslibGetEQGain	28
VslibSetEQGain	29
VslibGetHEQGain	30
VslibSetHEQGain	31

VslibGetTimeCtrlPntNum	32
VslibGetTimeCtrlPnt	33
VslibSetTimeCtrlPnt	34
VslibAddTimeCtrlPnt	35
VslibDeleteTimeCtrlPnt	36
VslibGetStretchOrgSec	37
VslibGetStretchEditSec	38
VslibGetStretchOrgSample	39
VslibGetStretchEditSample	40
ミキサー関連	41
VslibGetMixSample	41
VslibGetMixData	42
VslibExportWaveFile	43
単位変換関連	44
VslibCent2Freq	44
VslibFreq2Cent	44
構造体	45
プロジェクト情報	45
トラック情報	45
トラック情報(EX)	45
アイテム情報	46
アイテム制御点情報	46
アイテム制御点情報 2	47
エラーコード	48
エラーコード一覧	48

VocalShifter ライブライ

VocalShifter ライブライとは

VocalShifter の音声解析・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、VocalShifter が持つ音声解析・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。

ライブラリ関連

VslibGetVersion

```
int VslibGetVersion();
```

パラメータ
なし

戻り値
ライブラリのバージョン

解説

ライブラリのバージョンを返します。バージョンが 1.23 なら 123 を返します。

プロジェクト関連

VslibCreateProject

```
int VslibCreateProject(
    HVSPRJ *hVsprj           // 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

HVSPRJ *hVsprj
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(VSERR_NOERR, VSERR_PRM)

解説

新規プロジェクトを作成します。hVsprj にプロジェクトハンドルが格納されます。
VslibCreateProject で作成したプロジェクトは、VslibDeleteProject で破棄してください。

VslibOpenProject

```
int VslibOpenProject(
    HVSPRJ *hVsprj,          // プロジェクトハンドルが格納される変数へのポインタ
    char *fileName            // ファイル名
)
```

パラメータ

HVSPRJ *hVsprj
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。

char *fileName

ファイル名を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM, VSERR_PRJOPEN, VSERR_PRJFORMAT, VSERR_WAVEOPEN,
VSERR_WAVEFORMAT, VSERR_NOMEM)

解説

プロジェクトファイルを開きます。hVsprj にプロジェクトハンドルが格納されます。
VslibOpenProject で作成したプロジェクトは、VslibDeleteProject で破棄してください。

VslibSaveProject

```
int VslibSaveProject(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    char *fileName            // ファイル名
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

char *fileName

ファイル名を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM, VSERR_PRJOPEN, VSERR_NOMEM)

解説

プロジェクトをファイルに保存します。

VslibDeleteProject

```
int VslibDeleteProject(
    HVSPRJ hVsprj           // プロジェクトハンドル
)
```

パラメータ

HVSPRJ hVsprj

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(VSERR_NOERR, VSERR_PRM)

解説

プロジェクトを破棄します。VslibCreateProject または VslibOpenProject で作成したプロジェクトは VslibDeleteProject で破棄してください。

VslibGetProjectInfo

```
int VslibGetProjectInfo(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    VSPRJINFO *vsPrjInfo     // プロジェクト情報
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

VSPRJINFO *vsPrjInfo

プロジェクト情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

プロジェクト情報を取得します。

VslibSetProjectInfo

```
int VslibSetProjectInfo(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    VSPRJINFO *vsPrjInfo     // プロジェクト情報
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

VSPRJINFO *vsPrjInfo

プロジェクト情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

プロジェクト情報を設定します。

トラック関連

VslibGetTrackMaxNum

```
int VslibGetTrackMaxNum(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int *trackMaxNum         // トラック数が格納される変数へのポインタ
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int *trackMaxNum

トラック数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(VSERR_NOERR, VSERR_PRM)

解説

トラック数を取得します。

VslibAddTrack

```
int VslibAddTrack(
    HVSPRJ hVsprj           // プロジェクトハンドル
)
```

パラメータ

HVSPRJ hVsprj

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(VSERR_NOERR, VSERR_PRM, VSERR_MAX)

解説

プロジェクトに新規トラックを追加します。

VslibCopyTrack

```
int VslibCopyTrack(  
    HVSPRJ hVsprjDst,           // コピー先プロジェクトハンドル  
    int *trackNumDst,            // コピー先トラック番号  
    HVSPRJ hVsprjSrc,            // コピー元プロジェクトハンドル  
    int trackNumsrc              // コピー元トラック番号  
)
```

パラメータ

HVSPRJ hVsprjDst

コピー先プロジェクトハンドルを指定します。

int *trackNumDst

コピー先トラック番号が格納される変数へのアドレスを指定します。

HVSPRJ hVsprjSrc

コピー元プロジェクトハンドルを指定します。

int trackNumsrc

コピー元トラック番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM, VSERR_MAX)

解説

プロジェクト内またはプロジェクト間でトラックをコピーします。

VslibUpTrack

```
int VslibUpTrack(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int trackNum              // トラック番号
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラックを前のトラックと入れ替えます。

VslibDownTrack

```
int VslibDownTrack(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int trackNum              // トラック番号
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラックを次のトラックと入れ替えます。

VslibDeleteTrack

```
int VslibDeleteTrack(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int trackNum             // トラック番号  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラックを削除します。

VslibGetTrackInfo

```
int VslibGetTrackInfo(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int trackNum,            // トラック番号  
    VSTrackInfo *vsTrackInfo // トラック情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

VSTrackInfo *vsTrackInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラック情報を取得します。

VslibSetTrackInfo

```
int VslibSetTrackInfo(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int trackNum,            // トラック番号  
    VSTrackInfo *vsTrackInfo // トラック情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

VSTrackInfo *vsTrackInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラック情報を設定します。

VslibGetTrackInfoEx

```
int VslibGetTrackInfoEx(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int trackNum,            // トラック番号  
    VSTRACKINFOEX *vsTrackInfo // トラック情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

VSTRACKINFOEX *vsTrackInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラック情報を取得します。

VslibSetTrackInfoEx

```
int VslibSetTrackInfoEx(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int trackNum,            // トラック番号  
    VTRACKINFOEX *vsTrackInfo // トラック情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int trackNum

トラック番号を指定します。

VTRACKINFOEX *vsTrackInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

トラック情報を設定します。

アイテム関連

VslibGetItemMaxNum

```
int VslibGetItemMaxNum(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int *itemMaxNum          // アイテム数が格納される変数へのポインタ  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int *itemMaxNum

アイテム数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(VSERR_NOERR, VSERR_PRM)

解説

アイテム数を取得します。

VslibAddItem

```
int VslibAddItem(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    char *fileName,          // ファイル名  
    int *itemNum)            // アイテム番号が格納される変数へのポインタ  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

char *fileName

ファイル名を指定します。

int *itemNum

アイテム番号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 (VSERR_NOERR, VSERR_PRM, VSERR_WAVEOPEN, VSERR_WAVEFORMAT, VSERR_FREQ,
VSERR_MAX)

解説

プロジェクトにアイテムを追加します。

VslibAddItemEx

```
int VslibAddItemEx(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    char *fileName,           // ファイル名
    int *itemNum,             // アイテム番号が格納される変数へのポインタ
    int nnOffset,              // 解析する最低音のノートナンバー(C4=60)
    int nnRange,               // 解析する音階の範囲[半音]
    unsigned int option        // 解析オプション
)
```

パラメータ**HVSPRJ hVsprj**

プロジェクトハンドルを指定します。

char *fileName

ファイル名を指定します。

int *itemNum

アイテム番号が格納される変数へのポインタを指定します。

int nnOffset

解析する最低音のノートナンバーを指定します。C1～B7 までを解析する場合は 24 を指定します。

int nnRange

解析する音階の範囲を半音単位で指定します。C1～B7 までを解析する場合は 84 を指定します。

unsigned int option

以下の値の論理和を指定します。

ANALYZE_OPTION_SKIP_PIT_ANALYZE	ピッチを解析しない
ANALYZE_OPTION_WORLD	WORLD で分析する
ANALYZE_OPTION_VOCAL_SHIFTER	VocalShifter 独自方式で分析する

戻り値

エラーコード (VSERR_NOERR, VSERR_PRM, VSERR_WAVEOPEN, VSERR_WAVEFORMAT, VSERR_FREQ, VSERR_MAX)

解説

プロジェクトにアイテムを追加します。

VslibCopyItem

```
int VslibCopyItem(
    HVSPRJ hVspProjDst,          // コピー先プロジェクトハンドル
    int *itemNumDst,              // コピー先アイテム番号
    HVSPRJ hVspProjSrc,          // コピー元プロジェクトハンドル
    int itemNumSrc                // コピー元アイテム番号
)
```

パラメータ

HVSPRJ hVspProjDst

コピー先プロジェクトハンドルを指定します。

int *itemNumDst

コピー先アイテム番号が格納される変数へのアドレスを指定します。

HVSPRJ hVspProjSrc

コピー元プロジェクトハンドルを指定します。

int itemNumSrc

コピー元アイテム番号を指定します。

戻り値

エラーコード (VSERR_NOERR, VSERR_PRM, VSERR_WAVEOPEN, VSERR_WAVEFORMAT, VSERR_FREQ, VSERR_MAX)

解説

プロジェクト内またはプロジェクト間でアイテムをコピーします。

VslibDeleteItem

```
int VslibDeleteItem(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum               // アイテム番号  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

アイテムを削除します。

VslibGetItemInfo

```
int VslibGetItemInfo(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    VSITEMINFO *vsItemInfo // アイテム情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

VSITEMINFO *vsItemInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

アイテム情報を取得します。

VslibSetItemInfo

```
int VslibSetItemInfo(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    VSITEMINFO *vsItemInfo // アイテム情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

VSITEMINFO *vsItemInfo

トラック情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

アイテム情報を設定します。

VslibGetCtrIPntInfoEx

```
int VslibGetCtrIPntInfoEx(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    int pntNum,              // 制御点番号  
    VSCPINFOEX *vsCpInfo    // 制御点情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

制御点番号を指定します。

VSCPINFOEX *vsCpInfo

制御点情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

制御点情報を取得します。

VslibSetCtrlPntInfoEx

```
int VslibSetCtrlPntInfoEx(
```

HVSPRJ hVsprj,	// プロジェクトハンドル
int itemNum,	// アイテム番号
int pntNum,	// 制御点番号
VSCPINFOEX *vsCpInfo	// 制御点情報

```
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

制御点番号を指定します。

VSCPINFOEX *vsCpInfo

制御点情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

制御点情報を設定します。

VslibGetCtrIPntInfoEx2

```
int VslibGetCtrIPntInfoEx(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int pntNum,                // 制御点番号
    VSCPINFOEX2 *vsCpInfo2   // 制御点情報
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

制御点番号を指定します。

VSCPINFOEX2 *vsCpInfo2

制御点情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

制御点情報を取得します。

VslibSetCtrlPntInfoEx2

```
int VslibSetCtrlPntInfoEx(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    int pntNum,              // 制御点番号  
    VSCPINFOEX2 *vsCpInfo2 // 制御点情報  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

制御点番号を指定します。

VSCPINFOEX2 *vsCpInfo2

制御点情報構造体へのポインタを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

制御点情報を設定します。

VslibGetEQGain

```
int VslibGetEQGain (
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int eqNum,                // イコライザ番号(0:EQ1, 1:EQ2)
    int *gain                 // ゲインの値が格納される配列変数(15 要素、0.1dB 単位)
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int eqNum

EQ1なら0を、EQ2なら1を指定します。

int *gain

ゲインの値が格納される配列変数へのポインタを指定します。ゲインの値は1が0.1dBに相当します。15個の要素はそれぞれ以下の周波数におけるゲインです。

25Hz, 40Hz, 63Hz, 100Hz, 160Hz, 250Hz, 400Hz, 630Hz, 1.0kHz, 1.6kHz,
2.5kHz, 4.0kHz, 6.3kHz, 10.0kHz, 16.0kHz

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

EQのゲインを取得します。

VslibSetEQGain

```
int VslibSetEQGain (
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int eqNum,                // イコライザ番号(0:EQ1, 1:EQ2)
    int *gain                 // ゲインの値が格納されている配列変数(15要素、0.1dB 単位)
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int eqNum

EQ1なら0を、EQ2なら1を指定します。

int *gain

ゲインの値が格納されている配列変数へのポインタを指定します。ゲインの値は1が0.1dBに相当します。15個の要素はそれぞれ以下の周波数におけるゲインです。

25Hz, 40Hz, 63Hz, 100Hz, 160Hz, 250Hz, 400Hz, 630Hz, 1.0kHz, 1.6kHz,
2.5kHz, 4.0kHz, 6.3kHz, 10.0kHz, 16.0kHz

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

EQのゲインを設定します。

VslibGetHEQGain

```
int VslibGetEQGain (
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int *gain                 // ゲインの値が格納される配列変数(15 要素、0.1dB 単位)
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int *gain

ゲインの値が格納される配列変数へのポインタを指定します。ゲインの値は 1 が 0.1dB に相当します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

HEQ のゲインを取得します。

VslibSetHEQGain

```
int VslibSetEQGain (
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int *gain                 // ゲインの値が格納されている配列変数(15 要素、0.1dB 単位)
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int *gain

ゲインの値が格納されている配列変数へのポインタを指定します。ゲインの値は 1 が 0.1dB に相当します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

HEQ のゲインを設定します。

VslibGetTimeCtrlPntNum

```
int VslibGetTimeCtrlPntNum(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,              // アイテム番号  
    int *timeCtrlPntNum      // タイミング制御点数が格納される変数へのポインタ  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int *timeCtrlPntNum

タイミング制御点数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(VSERR_NOERR, VSERR_PRM)

解説

タイミング制御点数を取得します。

VslibGetTimeCtrlPnt

```
int VslibGetTimeCtrlPnt(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    int pntNum,              // タイミング制御点番号  
    int *time1,               // 編集前タイミング[サンプル]  
    int *time2                // 編集後タイミング[サンプル]  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

タイミング制御点番号を指定します。

int *time1

編集前タイミングが格納される変数へのポインタを指定します。

int *time2

編集後タイミング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(VSERR_NOERR, VSERR_PRM)

解説

タイミング制御点情報を取得します。

VslibSetTimeCtrlPnt

```
int VslibSetTimeCtrlPnt(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    int pntNum,               // タイミング制御点番号
    int time1,                // 編集前タイミング[サンプル]
    int time2                 // 編集後タイミング[サンプル]
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

タイミング制御点番号を指定します。

int time1

編集前タイミングを指定します。

int time2

編集後タイミングを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

タイミング制御点情報を設定します。

VslibAddTimeCtrlPnt

```
int VslibAddTimeCtrlPnt(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    int time1,               // 編集前タイミング[サンプル]  
    int time2                // 編集後タイミング[サンプル]  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int time1

編集前タイミングを指定します。

int time2

編集後タイミングを指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

タイミング制御点情報を追加します。

VslibDeleteTimeCtrlPnt

```
int VslibDeleteTimeCtrlPnt(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,              // アイテム番号  
    int pntNum                // タイミング制御点番号  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

int pntNum

タイミング制御点番号を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

タイミング制御点情報を削除します。

VslibGetStretchOrgSec

```
int VslibGetStretchOrgSec(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    double time_edt,          // タイムストレッチ後の時間[秒]
    double *time_org           // タイムストレッチ前の時間[秒]
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

double time_edt

タイムストレッチ後の時間を秒単位で指定します。

double *time_org

秒単位のタイムストレッチ前の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(VSERR_NOERR, VSERR_PRM)

解説

タイムストレッチ後の時間に対応するタイムストレッチ前の時間を取得します。

VslibGetStretchEditSec

```
int VslibGetStretchEditSec(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    double time_org,         // タイムストレッチ前の時間[秒]  
    double *time_edt,        // タイムストレッチ後の時間が格納される変数へのポインタ  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

double time_org

タイムストレッチ前の時間を秒単位で指定します。

double *time_edt

秒単位のタイムストレッチ後の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(VSERR_NOERR, VSERR_PRM)

解説

タイムストレッチ前の時間に対応するタイムストレッチ後の時間を取得します。

VslibGetStretchOrgSample

```
int VslibGetStretchOrgSample(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int itemNum,             // アイテム番号  
    double time_edt,         // タイムストレッチ後の時間[サンプル]  
    double *time_org          // タイムストレッチ前の時間[サンプル]  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

double time_edt

タイムストレッチ後の時間をサンプル単位で指定します。

double *time_org

サンプル単位のタイムストレッチ前の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(VSERR_NOERR, VSERR_PRM)

解説

タイムストレッチ後の時間に対応するタイムストレッチ前の時間を取得します。

VslibGetStretchEditSample

```
int VslibGetStretchEditSample(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    int itemNum,              // アイテム番号
    double time_org,          // タイムストレッチ前の時間[サンプル]
    double *time_edt,         // タイムストレッチ後の時間が格納される変数へのポインタ
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int itemNum

アイテム番号を指定します。

double time_org

タイムストレッチ前の時間をサンプル単位で指定します。

double *time_edt

サンプル単位のタイムストレッチ後の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(VSERR_NOERR, VSERR_PRM)

解説

タイムストレッチ前の時間に対応するタイムストレッチ後の時間を取得します。

ミキサー関連

VslibGetMixSample

```
int VslibGetMixSample(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    int *mixSample           // ミックス後サンプル数が格納される変数へのポインタ  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

int *mixSample

ミックス後サンプル数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(VSERR_NOERR, VSERR_PRM)

解説

ミックス後サンプル数を取得します。

VslibGetMixData

```
int VslibGetMixData(
    HVSPRJ hVsprj,           // プロジェクトハンドル
    void *waveData,          // 音声データが格納される変数へのポインタ
    int bit,                 // ビット数(8 または 16)
    int channel,             // チャンネル数(1 または 2)
    int index,               // 取得位置[サンプル]
    int size                 // 取得サイズ[サンプル]
)
```

パラメータ**HVSPRJ hVsprj**

プロジェクトハンドルを指定します。

void *waveData

音声データが格納される変数へのポインタを指定します。

int bit

ビット数を指定します。

int channel

チャンネル数を指定します。

int index

取得位置をサンプル単位で指定します。

int size

取得サイズをサンプル単位で指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM)

解説

ミックス後の音声データを取得します。

VslibExportWaveFile

```
int VslibExportWaveFile(  
    HVSPRJ hVsprj,           // プロジェクトハンドル  
    char *fileName,          // ファイル名  
    int bit,                 // ビット数(8 または 16)  
    int channel,             // チャンネル数(1 または 2)  
)
```

パラメータ

HVSPRJ hVsprj

プロジェクトハンドルを指定します。

char *fileName

ファイル名を指定します。

int bit

ビット数を指定します。

int channel

チャンネル数を指定します。

戻り値

エラーコード(VSERR_NOERR, VSERR_PRM, VSERR_WAVEOPEN)

解説

WAVE ファイルを出力します。

単位変換関連

VslibCent2Freq

```
double VslibCent2Freq(  
    int cent           // 1セント単位の音階  
)
```

パラメータ

int cent
C4=6000 とする 1セント単位の音階を指定します。

戻り値

周波数[Hz]

解説

C4=6000 とする 1セント単位の音階を Hz 単位の周波数に変換します。

VslibFreq2Cent

```
int VslibFreq2Cent(  
    double freq         // Hz 単位の周波数  
)
```

パラメータ

double freq
Hz 単位の周波数を指定します。

戻り値

C4=6000 とする 1セント単位の音階

解説

Hz 単位の周波数を C4=6000 とする 1セント単位の音階に変換します。

構造体

(R/-) のメンバ変数は、設定関数 (VslibSet～) で設定されません。取得のみ可能です。

プロジェクト情報

```
typedef struct{
    double masterVolume;           // (R/W) マスター音量[倍]
    int sampFreq;                 // (R/W) サンプリング周波数[Hz]
} VSPRJINFO;
```

トラック情報

```
typedef struct{
    double volume;                // (R/W) 音量[倍]
    double pan;                   // (R/W) パン(-1.0～1.0)
    int invPhaseFlg;             // (R/W) 逆相フラグ
    int soloFlg;                 // (R/W) ソロフラグ
    int muteFlg;                 // (R/W) ミュートフラグ
} VTRACKINFO;
```

トラック情報(EX)

```
typedef struct{
    char name[64];                // (R/W) トラック名
    double volume;                // (R/W) 音量[倍]
    double pan;                   // (R/W) パン(-1.0～1.0)
    double reserved1[2];          // 予約
    int invPhaseFlg;              // (R/W) 逆相フラグ
    int soloFlg;                 // (R/W) ソロフラグ
    int muteFlg;                 // (R/W) ミュートフラグ
    int color;                    // (R/W) トラックカラー(※1)
    int morphGroup;               // (R/W) モーフィンググループ(※2)
    unsigned int option;           // (R/W) トラックオプション(※3)
    int reserved2[10];            // 予約
} VTRACKINFOEX;
```

※1 0～7 がそれぞれ赤、橙、黄色、緑、水色、青、紫、ピンクに対応します。

※2 0 または 1 を指定します。0 は無効、1 はグループ 1 を表します。

※3 以下の値の論理和を指定します。

00000001H : モーフィング時にPITをミュート

00000002H : モーフィング時に音色をミュート

アイテム情報

```
typedef struct{
    char fileName[VSLIB_MAX_PATH]; // (R/-) ファイル名
    int sampFreq; // (R/-) サンプリング周波数[Hz]
    int channel; // (R/-) チャンネル数(1or2)
    int sampleOrg; // (R/-) オリジナルの音声ファイルのサンプル数
    int sampleEdit; // (R/-) 編集後のサンプル数
    int ctrlPntPs; // (R/-) 1秒あたりの制御点数
    int ctrlPntNum; // (R/-) 全制御点数
    int synthMode; // (R/W) 音声合成方式(M:0, MF:1, P:2)
    int trackNum; // (R/W) トラック番号(0～)
    int offset; // (R/W) オフセット[sample]
} VSITEMINFO;
```

アイテム制御点情報

```
typedef struct{
    double dynOrg; // (R/-) 編集前ダイナミクス[倍]
    double dynEdit; // (R/W) 編集後ダイナミクス[倍]
    double volume; // (R/W) ボリューム[倍]
    double pan; // (R/W) パン(-1.0～1.0)
    double spcDyn; // (R/-) スペクトルダイナミクス
    int pitAna; // (R/-) ピッチ解析値[cent, C4=6000]
    int pitOrg; // (R/W) 編集前ピッチ[cent, C4=6000]
    int pitEdit; // (R/W) 編集後ピッチ[cent, C4=6000]
    int formant; // (R/W) フォルマント[cent]
    int pitFlgOrg; // (R/-) 編集前ピッチ有無フラグ
    int pitFlgEdit; // (R/W) 編集後ピッチ有無フラグ
    int breathiness; // (R/W) ブレシネス[-10000～10000]
    int eq1; // (R/W) EQ1[-10000～10000]
    int eq2; // (R/W) EQ2[-10000～10000]
} VSCPINFOEX;
```

アイテム制御点情報 2

```
typedef struct{
    double dynOrg;           // (R/-) 編集前ダイナミクス[倍]
    double dynEdit;          // (R/W) 編集後ダイナミクス[倍]
    double volume;           // (R/W) ボリューム[倍]
    double pan;              // (R/W) パン(-1.0~1.0)
    double spcDyn;           // (R/-) スペクトルダイナミクス
    double reserved1;        // 予約
    int pitAna;              // (R/-) ピッチ解析値[cent, C4=6000]
    int pitOrg;              // (R/W) 編集前ピッチ[cent, C4=6000]
    int pitEdit;              // (R/W) 編集後ピッチ[cent, C4=6000]
    int formant;             // (R/W) フォルマント[cent]
    int pitFlgOrg;           // (R/-) 編集前ピッチ有無フラグ
    int pitFlgEdit;          // (R/W) 編集後ピッチ有無フラグ
    int breathiness;          // (R/W) ブレシネス[-10000~10000]
    int eq1;                 // (R/W) EQ1[-10000~10000]
    int eq2;                 // (R/W) EQ2[-10000~10000]
    int heq;                 // (R/W) HEQ[-10000~10000]
    int reserved2[6];         // 予約
} VSCPINFOEX2;
```

エラーコード

エラーコード一覧

エラーコード	内容
VSERR_NOERR	エラーなし
VSERR_PRM	パラメータ不正
VSERR_PRJOPEN	プロジェクトファイルオープンに失敗
VSERR_PRJFORMAT	プロジェクトファイルフォーマット不正
VSERR_WAVEOPEN	WAVE ファイルオープンに失敗
VSERR_WAVEFORMAT	WAVE ファイルフォーマット不正
VSERR_FREQ	サンプリング周波数不正
VSERR_MAX	最大数制限
VSERR_NOMEM	メモリ不足