

引数:LnagM 主言語のコード 1:英語 0x41: 日本語  
 引数:LangS 補助言語のコード 1:英語 0x41: 日本語  
 引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定  
 引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定  
 引数:lpsIndexTable 出力:0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時はNULLを指定してください  
 呼び出しの後では、例えば

```
indexTable[0] 23 点字列のバイト数
indexTable[1] 0 点字列の第0バイト目は、入力文字列の第0バイト目に対応
indexTable[2] 0 点字列の第1バイト目は、入力文字列の第0バイト目に対応
indexTable[3] 2 点字列の第2バイト目は、入力文字列の第2バイト目に対応
indexTable[4] 2 点字列の第3バイト目は、入力文字列の第2バイト目に対応
indexTable[5] 4 点字列の第4バイト目は、入力文字列の第4バイト目に対応
indexTable[6] 6 点字列の第5バイト目は、入力文字列の第6バイト目に対応
indexTable[7] 6 点字列の第6バイト目は、入力文字列の第6バイト目に対応
indexTable[8] 8 点字列の第7バイト目は、入力文字列の第8バイト目に対応
```

となります。

なお、出力点字列の1バイト目は0バイト目のグループに属するものと考えます。

入力文字列: 漢字かな混じりの文章を半角のカナにしてから、点字にします。This is a pen.  
 出力文字列(点字): \*0"¥\*K Z"¥HS "X0@W39 U0\*%S \*KL ¥Q\*E; Q0"¥L ¥Z?4 8,? IS A P540

テーブルの内容 (table[1]から)

```
0 0 2 2 4 6 6 8
10 10 12 14 14 16 16 16
18 18 18 20 20 22 22 24
24 26 26 28 30 32 32 34
36 38 40 42 42 44 44 46
46 48 48 50 52 54 56 56
56 58 58 58 62 63 64 65
66 67 68 68 70 70
```

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-8 ZenToBrR 関数

全角を点字にします。逆方向インデックスをサポートします。

EXDLL\_EXPORT ZenToBrR(char\* InStr, char\* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short\* lpsIndexTable, short\* lpsRIndexTable);

引数:InStr 入力文字列  
 引数:OutStr 出力バッファ  
 引数:LnagM 主言語のコード 1:英語 0x41: 日本語  
 引数:LangS 補助言語のコード 1:英語 0x41: 日本語  
 引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定  
 引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定  
 引数:sOutLimit 出力バッファへの文字列の長さの制限を指定。  
 引数:lpsIndexTable 出力:0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時はNULLを指定してください

引数: `lpsRIndexTable` 逆方向インデックス 0番目に入力文字列のバイト数、1要素から点訳結果の文字が、元の文字のどれにあたるかという表を返す

引数: `lpsRIndexTable` `short` 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は `NULL` を与えてください

返り値: 成功の時に `TRUE`、失敗の時に `FALSE` を返します。

#### ◆ 10-9 ZenToBrHin 関数

全角を点字にします。逆方向インデックスをサポートします。

```
EXDLL_EXPORT ZenToBrHin(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable, short* lpsHinTable);
```

引数: `InStr` 入力文字列

引数: `OutStr` 出力バッファ

引数: `LnagM` 主言語のコード 1:英語 0x41: 日本語

引数: `LangS` 補助言語のコード 1:英語 0x41: 日本語

引数: `ForeignModeM` 後述の主要言語種別(`LangM`)に対する点訳の水準を指定

引数: `ForeignModeS` 後述の補助言語種別(`LangS`)に対する点訳の水準を指定

引数: `sOutLimit` 出力バッファへの文字列の長さの制限を指定。

引数: `lpsIndexTable` 出力: 0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時は `NULL` を指定してください

引数: `lpsRIndexTable` 逆方向インデックス 0番目に入力文字列のバイト数、1要素から点訳結果の文字が、元の文字のどれにあたるかという表を返す

引数: `lpsRIndexTable` `short` 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は `NULL` を与えてください

返り値: 成功の時に `TRUE`、失敗の時に `FALSE` を返します。

#### ◆ 10-10 RawZenToBrTabR 関数

全角を点字にします。

```
EXDLL_EXPORT RawZenToBrTabR(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable, short* lpsRIndexTable, short sTabs);
```

引数: `InStr` 入力文字列

引数: `OutStr` 出力バッファ

引数: `LnagM` 主言語のコード 1:英語 0x41: 日本語

引数: `LangS` 補助言語のコード 1:英語 0x41: 日本語

引数: `ForeignModeM` 後述の主要言語種別(`LangM`)に対する点訳の水準を指定

引数: `ForeignModeS` 後述の補助言語種別(`LangS`)に対する点訳の水準を指定

引数: `sOutLimit` 出力バッファへの文字列の長さの制限を指定

引数: `lpsIndexTable` 出力: 0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時は `NULL` を指定してください

引数: `lpsRIndexTable` `short` 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は `NULL` を与えてください

引数: `sTabs` 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

返り値: 成功の時に `TRUE`、失敗の時に `FALSE` を返します。

#### ◆ 10-11 RawZenToBrTab 関数

全角を点字にします。

```
EXDLL_EXPORT RawZenToBrTab(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable, short sTabs);
```

引数:InStr 入力文字列

引数:OutStr 出力バッファ

引数:LnagM 主言語のコード 1:英語 0x41: 日本語

引数:LangS 補助言語のコード 1:英語 0x41: 日本語

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定

引数:sOutLimit 出力バッファへの文字列の長さの制限を指定

引数:lpsIndexTable 出力:0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時は NULL を指定してください

引数:lpsRIndexTable short 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は NULL を与えてください

引数:sTabs 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

返り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-12 ZenToBr 関数

全角を点字にします。

```
EXDLL_EXPORT ZenToBr(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable);
```

引数:InStr 入力文字列

引数:OutStr 出力バッファ

引数:LnagM 主言語のコード 1:英語 0x41: 日本語

引数:LangS 補助言語のコード 1:英語 0x41: 日本語

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定

引数:sOutLimit 出力バッファへの文字列の長さの制限を指定

引数:lpsIndexTable 出力:0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。不要な時は NULL を指定してください

引数:sTabs 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

返り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-13 RawZenToBr 関数

全角を点字にします。

```
EXDLL_EXPORT RawZenToBr(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable);
```

引数:InStr 入力文字列

引数:OutStr 出力バッファ

引数:LnagM 主言語のコード 1:英語 0x41: 日本語

引数:LangS 補助言語のコード 1:英語 0x41: 日本語

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定

引数:sOutLimit 出力バッファへの文字列の長さの制限を指定

引数:lpsIndexTable 出力:0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です。

不要な時は NULL を指定してください

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-14 ZenToHan 関数

全角を半角にします。

EXDLL\_EXPORT ZenToHan(char\* InStr, char\* OutStr, short sOutLimit, short\* lpsIndexTable);

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-15 ZenToHanTabRHin 関数

全角を半角にします。

EXDLL\_EXPORT ZenToHanTabRHin(char\* InStr, char\* OutStr, short sOutLimit, short\* lpsIndexTable, short\* lpsRIndexTable, short\* lpsHinTable, short sTabs);

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です

引数:lpsRIndexTable short 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は NULL を与えてください

引数:sTabs 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-16 ZenToHanTabR 関数

全角を半角にします。

EXDLL\_EXPORT ZenToHanTabR(char\* InStr, char\* OutStr, short sOutLimit, short\* lpsIndexTable, short\* lpsRIndexTable, short sTabs);

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0番目に文字列のバイト長、1番以降は入力文字列0バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1が必要です

引数:lpsRIndexTable short 型配列を指すポインタ。第0要素に入力文字列のバイト長がセットされる。要素数は出力点字列のバイト長+1以上になるように充分余裕を持たせてください。作成しない場合は NULL を与えてください

引数:sTabs 入力全角文字列内のタブ・コードをスペース何個に展開するか指定  
戻り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-17 ZenToHanTabHin 関数

全角を半角にします。

```
EXDLL_EXPORT ZenToHanTabHin(char* InStr, char* OutStr, short sOutLimit, short* lpsIndexTable,  
short* lpsHinTable, short sTabs);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsHinTable 出力: 品詞テーブル

引数:sTabs 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

戻り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-18 ZenToHanTab 関数

全角を半角にします。(タブコードをスペースに変換する:スペースの数を設定可能)

```
EXDLL_EXPORT ZenToHanTab(char* InStr, char* OutStr, short sOutLimit, short* lpsIndexTable, short  
sTabs);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:sTab 入力全角文字列内のタブ・コードをスペース何個に展開するか指定

戻り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-19 ZenToHanHin 関数

全角を半角にします (品詞情報付)

```
EXDLL_EXPORT ZenToHanHin(char* InStr, char* OutStr, short sOutLimit, short* lpsIndexTable,  
short* lpsHinTable);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsHinTable 品詞情報

戻り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-20 ZenToHanRHin 関数

全角を半角に変換します。(品詞情報付)

```
EXDLL_EXPORT ZenToHanRHin(char* InStr, char* OutStr, short sOutLimit, short* lpsIndexTable,  
short* lpsRIndexTable, short* lpsHinTable);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsRIndexTable 0 番目出力文字列のバイト数、1 番以降は出力文字列に対する入力文字列のオフセット位置から変換されたことを表します。このテーブルは出力で出力文字列のバイト数+1 が必要です

引数:lpsHinTable 品詞情報

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-21 ZenToHanR 関数

全角を半角に変換します。

```
EXDLL_EXPORT ZenToHanR(char* InStr, char* OutStr, short sOutLimit, short* lpsIndexTable, short* lpsRIndexTable);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsRIndexTable 0 番目出力文字列のバイト数、1 番以降は出力文字列に対する入力文字列のオフセット位置から変換されたことを表します。このテーブルは出力で出力文字列のバイト数+1 が必要です

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-22 HanToBrR 関数

全角を半角に変換します。

```
EXDLL_EXPORT HanToBrR(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable, short* lpsRIndexTable);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsRIndexTable 0 番目に点字文字列のバイト数、1 番以降は点字文字列に対する入力文字列のオフセット位置から変換されたことを表します。このテーブルは出力で出力文字列のバイト数+1 が必要です

返回值:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-23 HanToBr 関数

半角を NABCC コードに変換します。

```
EXDLL_EXPORT HanToBr(char* InStr, char* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short* lpsIndexTable);
```

引数:InStr 入力文字列

引数:OutStr 出力文字列

引数:LangM short 型、主言語のコード。英語は 1、日本語は 41h です。

引数:LangS char 型、補助言語のコードです。

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:sOutLimit 出力バッファの長さ (byte 単位)

引数:lpsIndexTable 出力: 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です  
返り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-24 RawHanToBrR 関数

モードをセットし ex\_braille を呼び出すか、サポートチェックを行います。

EXDLL\_EXPORT RawHanToBrR(char\* InStr, char\* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short\* lpsIndexTable, short\* lpsRIndexTable);

引数:f\_lang\_m メイン言語

引数:LangM short 型、主言語のコード。英語は 1、日本語は 41h です。

引数:LangS char 型、補助言語のコードです。

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:lpsRIndexTable 0 番目に点字文字列のバイト数、1 番以降は点字文字列に対する入力文字列のオフセット位置から変換されたことを表します。このテーブルは出力で出力文字列のバイト数+1 が必要です

引数:raw\_mode 不明

返り値:成功の時に TRUE、失敗の時に FALSE を返します。

#### ◆ 10-25 RawHanToBr 関数

モードをセットし ex\_braille を呼び出すか、サポートチェックを行います。

EXDLL\_EXPORT RawHanToBr(char\* InStr, char\* OutStr, short LangM, short LangS, short ForeignModeM, short ForeignModeS, short sOutLimit, short\* lpsIndexTable);

引数:f\_lang\_m メイン言語

引数:LangM short 型、主言語のコード。英語は 1、日本語は 41h です。

引数:LangS char 型、補助言語のコードです。

引数:ForeignModeM 後述の主要言語種別(LangM)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:ForeignModeS 後述の補助言語種別(LangS)に対する点訳の水準を指定。日本語(0:通常文、11H:情報処理点字)、英語(0:1級、1:2級短縮、11H:北米情報)

引数:lpsIndexTable 0 番目に文字列のバイト長、1 番以降は入力文字列 0 バイト目の出力文字列のオフセット位置に変換されたことを表します。このテーブルは出力で入力文字列のバイト数+1 が必要です

引数:raw\_mode 不明

返り値:成功の時に TRUE、失敗の時に FALSE を返します。

---

---

---

---