

<日本語文字符号変換仕様書>

<基本仕様>

日本語文字符号変換の基本仕様について、記載する。対応する文字は、SJIS に、配置されている文字に限定する仕様とする。SJIS は1バイトの半角文字と、2バイトの全角文字から構成されているが、このように複数バイトからなる文字符号を含んだ符号体系を、以降、マルチバイトキャラクターと表現する。

- 実装する文字符号の変換機能は、基本的に SJIS と、他のマルチバイトキャラクターとの相互変換機能である。

他のマルチバイトキャラクターとは、EUC、JIS、UTF-8、Unicode(w_chjar)である。

- SJIS との相互変換を実装するので、文字符号変換に対応する文字は、SJIS に配置されている文字のみとする。別紙『SJIS コード表.xlsx』の一覧表に纏める。
- SJIS や、その他の文字符号に配置されていない文字は、? (&H3F)に変換する仕様とする。マルチバイトの文字符号については、そのバイト数分の ? (&H3F)に変換する。
- 文字符号の変換は、変換テーブル(配列)による変換処理とする。ソースの文字符号をキーに配列をインデックス参照し、符号変換するものである。この方式には、デメリットとして、無駄なリソースを確保する(使用されない配列が多く存在)点があげられるが、ソフトウェア構造のシンプルさと、インデックス参照による変換処理の高速さの利点を考慮し、この方式を採用する。但し、UTF-8 は、3バイト以上の文字符号が存在するため、配列が確保できない(サイズオーバー)ので、1バイト目をIF文により処理分岐し、以降のバイトデータをキーに配列参照する仕様とする。
- VBA のプログラム上では、文字列データは、String オブジェクト(Unicode)で管理されることがほとんどであるが、この String オブジェクトと SJIS の相互変換は、VBA のライブラリで提供されている StrConv 関数を仕様するものとする。また、String オブジェクトの保持している文字列バッファ(w_char)への参照は StrPtr を利用する。

<例> Dim str as string, Dim sjis_bufs() as Byte, Dim wp as LongPtr

sjis_bufs = StrConv(str, vbFromUnicode) 'SJIS のバッファに変換

str = StrConv(sjis_bufs, vbUnicode) 'String オブジェクトに変換

wp = StrPtr(str) 'String オブジェクトの w_char のバッファ

- SJIS と UTF-8 の相互変換における特記事項。
 - UTF-8 のバックスラッシュ \ (&H5C)に該当する文字は、SJIS には、存在しないので、不明符号 ? (&H3F)に変換する。
 - UTF-8 の半角 ¥ (&HC2A5)は、SJIS の半角 ¥(&H5C)に変換する。
 - SJIS の半角 ¥ (&H5C) は、UTF-8 の半角 ¥ (&HC2A5)に変換する仕様とする。
 - SJIS にバックスラッシュは存在しないので、バックスラッシュについては、特に何も言及しない。

<関数仕様>

• UTF-8

Public Function **UtfConvertSjisToUtf8**(_

ByRef inp() *As Byte*, *ByVal inp_num As Long*, _

ByRef out() *As Byte*, *ByRef out_num As Long*) *As Long*

概要 : SJIS 符号のバッファを、UTF-8 符号に変換し出力する。出力バッファのサイズについては、関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 *inp* : 入力データとなる SJIS 符号のバッファを指定する。

引数 *inp_num* : 入力データのサイズを指定する。

引数 *out* : 出力データ (UTF-8) を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 *out_num* : 出力データのサイズが格納される。

Public Function **UtfConvertUtf8ToSjis**(_

ByRef inp() *As Byte*, *ByVal inp_num As Long*, _

ByRef out() *As Byte*, *ByRef out_num As Long*) *As Long*

概要 : UTF-8 符号のバッファを、SJIS 符号に変換し出力する。出力バッファのサイズについては、関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 *inp* : 入力データとなる UTF-8 符号のバッファを指定する。

引数 *inp_num* : 入力データのサイズを指定する。

引数 *out* : 出力データ (SJIS) を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 *out_num* : 出力データのサイズが格納される。

• EUC

Public Function **EucConvertSjisToEuc**(_

ByRef inp() *As Byte*, *ByVal inp_num As Long*, _

ByRef out() *As Byte*, *ByRef out_num As Long*) *As Long*

概要 : SJIS 符号のバッファを、EUC 符号に変換し出力する。出力バッファのサイズについては、関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 *inp* : 入力データとなる SJIS 符号のバッファを指定する。

引数 *inp_num* : 入力データのサイズを指定する。

引数 *out* : 出力データ (EUC) を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 *out_num* : 出力データのサイズが格納される。

Public Function EucConvertEucToSjis(_

ByRef inp() As Byte, *ByVal inp_num* As Long, _

ByRef out() As Byte, *ByRef out_num* As Long) As Long

概要 : EUC 符号のバッファを、SJIS 符号に変換し出力する。出力バッファのサイズについては、関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 inp : 入力データとなる EUC 符号のバッファを指定する。

引数 inp_num : 入力データのサイズを指定する。

引数 out : 出力データ(SJIS)を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 out_num : 出力データのサイズが格納される。

• JIS

Public Function JisConvertSjisToJIS(_

ByRef inp() As Byte, *ByVal inp_num* As Long, _

ByRef out() As Byte, *ByRef out_num* As Long) As Long

概要 : SJIS 符号のバッファを、JIS 符号に変換し出力する。出力バッファのサイズについては、関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 inp : 入力データとなる SJIS 符号のバッファを指定する。

引数 inp_num : 入力データのサイズを指定する。

引数 out : 出力データ(JIS)を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 out_num : 出力データのサイズが格納される。

Public Function JisConvertJISToSjis(_

ByRef inp() As Byte, *ByVal inp_num* As Long, _

ByRef out() As Byte, *ByRef out_num* As Long) As Long

概要 : JIS 符号のバッファを、SJIS 符号に変換し出力する。関数内で再確保により、適切なサイズに変更する。

戻り値 : 出力バッファのサイズを返却する。

引数 inp : 入力データとなる JIS 符号のバッファを指定する。

引数 inp_num : 入力データのサイズを指定する。

引数 out : 出力データ(SJIS)を格納するバッファを指定する。関数呼び出し前に、必要サイズを確保。

引数 out_num : 出力データのサイズが格納される。

以上