

# DLL StringHandler マニュアル



## DLL StringHandler の機能

文字列、文字配列を加工・編集・検索するVisual C#のプログラムで汎用に利用できるライブラリ群を収納する。人名、文献名を編集・加工・検索するライブラリも含む。

## DLL StringHandler の構造

namespace StringHandler

```
{  
    public class pickTop  
    {  
        この中にライブラリ群を定義する。  
        public string module(string[] strA, int intB,...)  
        {  
        }  
    }  
}
```

# DLL StringHandler の使い方



1. StringHandler.dllのファイルを任意のフォルダーにコピーする。
2. 利用するプログラムの編集画面で「プロジェクト」のプルダウン中「参照の追加」をクリック。
3. 現れた画面で「参照」ボタンを押し、stringHandler.dllをクリックし、「追加」ボタンを押し。
4. 利用するプログラムの編集画面で「表示」のプルダウンリスト中、「ソリューションエクスプローラ」をクリック。
5. ソリューションエクスプローラのウインドウで「参照」をクリック。
6. 参照のリスト内にStringHandler.dllが入っていれば導入成功。
7. プログラムの最初にusing StringHandler;を宣言する。
8. プログラムの初めの諸変数を定義する部分に  
`Internal (publicでも良い) pickTop alpha (任意の名前) = new pickTop();`  
を宣言する。
9. プログラム中でプログラムの出力形式にあわせて  
`alpha.モジュール名 (引数) ; //public void型`  
`int intA=alpha.モジュール (引数) ; //整数を返すモジュール`  
`string[] strB= alpha.モジュール (引数) ; //文字配列を返すモジュール`  
のように利用できる。

# モジュール目次



モジュール名	機能	ページ
textOutput	ファイル fileName にテキスト strText をフォーマット strFormat で書き込む。	6
textInput:	ファイル fileName 内のテキストをライン毎に配列にして全て読み込む	7
fetchTop	strTemp から numSeq 番目の strKey より前の文字列 strTop を出力する	8
pickBack	strBase の最後尾から intNum 文字を取り出す。intNum が多すぎたら、文字数が合うように後方に "X" を加える。	9
GetCharaPos	文字列 strTotal から intCount 番目の strChara の先頭位置 intChara を返す 見付からない時は intChara=-1	10
getExtention	ファイルのパス名 fileName からファイルの拡張子とそれを除いた文字列を取得する。	11
countKey	文字列 strBase の中に含まれる文字列 strKey の数を返す。	12
trimText	strAll 中の "" (ダブルクォーテーション) と " " (空文字) を除去した後の文字列の前 intBefore 語数、後 intAfter 語数分除去	13

# モジュール目次



モジュール名	機能	ページ
cutInclusive	strRangeからintQue 番目のstrBeforeと直後のstrAfterを含めた部分を取り出す	14
cutExclusive	strRangeからintQue 番目のstrBeforeと直後のstrAfterno 間の文字列からstrBeforeとstrAfterを除いた部分を取り出す	15
pickString	文字列strRangeから文字数intQue 番目からのnumString 語数を取り出す	16
ChangeText	strBaseに含まれるintSeq番目のstrOutをstrInに変える intSeq=-1 : strBaseに含まれるstrOutを全てstrInに変える	17
SelectUnique	文字配列strAll[]から、重複の無い文字配列strSelect[]を選び出す。	18
fileFound	パス名fileNameのファイルが存在する場合 1 を、存在しない場合 - 1 を返す	19
DecompText	文字列strTextから文字列keyCharaのある前後で文字列を文字配列に分解する。 keyCharaは含めない (引数intBefore, intAfterは不使用)	20
splitString	strBase中のintSplit番目のstrKey前後の文字列をstrOut[0],strOut[1]に返す。	21

## モジュール目次(続2)



モジュール名	機能	ページ
<b>strConnect</b>	文字配列strPartsをタブ ( <code>¥¥t</code> )で区切って単一の文字strOutとする。	22
<b>lstConnect</b>	文字配列strPartsを単一のList<string> Genericとする。	23
<b>findLastPath</b>	パス名strPathの最後のフォルダまたはファイル名strOutを返す	24
<b>setName</b>	英名の名前strNameを指定の表記法strCodeに変換する。 表記法： N (Normal) :E, lbe R (Reverse) :lbe, E. strNameがNかRを判定し、strCode(NまたはR) に変換する。	25

<b>プログラム名</b>	<b>public void textOutput(string fileName, string strText, string strFormat)</b>	
<b>機能</b>	ファイルfileNameにテキストstrTextをフォーマットstrFormatで書き込む。	
<b>引数</b>	fileName	書きこみパス(string)
	strText	文字列(string)
	strFormat	フォーマット(string、UTF-8など)
<b>戻り変数</b>	なし	void
<b>使用例</b>		

<b>プログラム名</b>	<b>public string[] textInput(string fileName)</b>	
<b>機能</b>	ファイルfileName内のテキストをライン毎に配列にして全て読み込む。	
<b>引数</b>	fileName	ファイル名(string)
<b>戻り変数</b>	行毎の文字列	string[]
<b>使用例</b>		

プログラム名	public string fetchTop(string strTemp, string strKey, int numSeq)	
機能	strTempからnumSeq番目のstrKeyより前の文字列strTopを出力する	
引数	strTemp	加工対象文字列(string)
	strKey	検索文字列(string)
	numSeq	検索文字列の先頭からの個数(int)
戻り変数	先頭からの文字列	string
使用例	<p style="text-align: right;">▼たとえばnumSeq=3</p> <p>strTemp : aaabbbb...dddd<b>strkey</b>eee<b>strkey</b>ffff<b>strkey</b>xxxx.....</p> <p style="text-align: center;">└──────────────────────────────────┘</p> <p style="text-align: center;">戻りのstring</p>	



プログラム名	public string pickBack(string strBase,int intNum)	
機能	strBaseの最後尾からintNum文字を取り出す。intNumが多すぎたら、文字数が合うように後方に” X “を加える。	
引数	strBase	対象文字列(string)
	intNum	取り出す文字数(int)
戻り変数	取り出した文字列	string
使用例	<p>strBase : aaabbbbdddd ⇒</p> <p style="text-align: center;">└──────────┘</p> <p style="text-align: center;">11文字</p> <p>intNum=8のとき戻り文字列はaaabbbbd</p> <p>intNum=15のとき戻り文字列はaaabbbbddddXXXX</p>	

プログラム名	public int GetCharaPos(string strTotal, string strChara, int intCount)	
機能	文字列strTotalからintCount番目のstrCharaの先頭位置intCharaを返す 見付からない時はintChara=-1	
引数	strTotal	対象文字列(string)
	strChara	検索文字列
	intCount	strCharaの抽出順番（最初が1）
戻り変数	strCharaの先頭の文字数 最初は0	int
使用例	<p>strTotal : aaakeybbbkeybddd  strChara: key  intCount=2⇒</p> <p style="text-align: center;">↓</p> <p style="text-align: right;">戻り値は9</p>	

プログラム名	public string[] getExtention(string fileName)	
機能	ファイルのパス名fileNameからファイルの拡張子とそれを除いた文字列を取得する。	
引数	fileName	
戻り変数	string[0]	ファイルの拡張子
	String[1]	ファイルのパス名から".拡張子"を除いた文字列
使用例	<pre>filename= 'l://¥homepage¥document¥test.txt'</pre> <pre>=&gt; string[0]= 'txt'</pre> <pre>string[1]== 'l://¥homepage¥document¥test'</pre>	

プログラム名	public int countKey(string strBase,string strKey)	
機能	文字列strBaseの中に含まれる文字列strKeyの数を返す。	
引数	strBase	検索対象の文字列
	strKey	カウントする文字列
戻り変数	int	strKeyが含まれていた個数
使用例	<b>strBase= 'aaa<b>key</b>kkk<b>key</b>vvv<b>key</b>nn'</b> <b>strKey= 'key'</b>  <b>=&gt;int=3</b>	

プログラム名	public string trimText(string strAll, int intBefore, int intAfter)	
機能	strAll中の“”（ダブルクォーテーション）と” “（空文字）を除去した後の文字列の前intBefore語数、後intAfter語数分除去	
引数	strAll	対象文字列
	intBefore	先頭から削除する語数
	intAfter	末尾まで削除する語数
戻り変数	整形後の文字列	string
使用例	<b>strAll=' abc d"ffyg'</b> <b>intBefore=2</b> =>整形後の文字列='cdff'  <b>intAfter=3</b>	

プログラム名	public string cutInclusive(string strRange, string strBefore, string strAfter, int intQue)	
機能	strRangeからintQue 番目のstrBeforeと直後のstrAfterを含めた部分を取り出す	
引数	strRange	対象文字列
	strBefore	取り出す先頭の文字列
	strAfter	取り出す末尾の文字列
	intQue	抽出するstrBeforeの順番数
戻り変数	整形後の文字列	string
使用例	<pre> strange=xyz<b>before</b>hhhfg<b>after</b>lllk<b>before</b>nnn strBefore= 'before' strAfter= 'after' intQue=1 =&gt;整形後の文字列='beforehhhfgafter' </pre>	

プログラム名	<code>public string cutExclusive(string strRange, string strBefore, string strAfter, int intQue)</code>	
機能	strRangeからintQue 番目のstrBeforeと直後のstrAfter間の文字列からstrBeforeとstrAfterを除いた部分を取り出す	
引数	strRange	対象文字列
	strBefore	取り出す先頭の文字列
	strAfter	取り出す末尾の文字列
	intQue	抽出するstrBeforeの順番数
戻り変数	整形後の文字列	string
使用例	<pre> strange=xyz<b>before</b>hhhfg<b>after</b>lllk<b>before</b>nnn strBefore= 'before' strAfter= 'after' intQue=1 =&gt;整形後の文字列='hhhfg' </pre>	

プログラム名	<code>public string pickString(string strRange, int intQue, int numString)</code>	
機能	文字列strRangeから文字数intQue 番目からのnumString 語数を取り出す	
引数	strRange	対象文字列
	intQue	取り出す最初からの文字数
	numString	取り出す語数
戻り変数	整形後の文字列	string
使用例	<pre> <b>strRange= 'sdfgrghunjki'</b> <b>intQue=5</b> <b>numString=4</b>           =&gt; <b>'ghun'</b> </pre>	



プログラム名	public string ChangeText(string strBase, string strOut, string strIn, int intSeq)	
機能	strBaseに含まれるintSeq番目のstrOutをstrInに変える intSeq=-1 : strBaseに含まれるstrOutを全てstrInに変える	
引数	strBase	加工前の文字列
	strOut	変更対象の文字列
	strIn	変更する文字列
	intSeq	変更対象のstrOutの順番。 - 1 の時すべて変更
戻り変数	加工後の文字列	string
使用例	<b>strBase= 'ddffkkeycfbhkeykhh'</b> <b>strOut= 'key'</b> <b>strIn= 'in'</b>  => intSeq=1 の時: 'ddffkincfbhkeykhh' intSeq=-1の時: 'ddffkincfbhinkhh'	

プログラム名	public string[] SelectUnique(string[] strAll)	
機能	文字配列strAll[]から、重複の無い文字配列strSelect[]を選び出す。	
引数	strAll	加工対象の文字配列
戻り変数	(strSelect)	加工後の文字配列。 配列数は重複の無い文字列数。
使用例	<pre> strAll[0] = 'abc' strAll[1] = 'klm' strAll[2] = 'abc'  =&gt; strSelect[0] = 'abc'     strSelect[1] = 'klm' </pre>	

プログラム名	<code>public int fileFound(string filename)</code>	
機能	パス名 <code>fileName</code> のファイルが存在する場合1を、存在しない場合-1を返す	
引数	<code>fileName</code>	ドライブ名を含むファイルのパス名
戻り変数	在非を示す整数	存在する場合1を、存在しない場合-1を返す
使用例		

プログラム名	<code>public string[] DecompText(string strText, string keyChara, int intBefore, int intAfter)</code>	
機能	文字列strTextから文字列keyCharaのある前後で文字列を文字配列に分解する。keyCharaは含めない（引数intBefore, intAfterは不使用）	
引数	strText	対象文字列
	keyChara	前後で分解する文字列
戻り変数	string[]	分解後の文字配列
使用例	<pre> strText = 'aaakeybbkeycccckeyd' keyChara = 'key'  =&gt; string[0] = 'aaa'     string[1] = 'bb'     string[2] = 'cccc'     string[3] = 'd' </pre>	

プログラム名	<code>public string[] splitString(string strBase,string strKey, int intSplit)</code>	
機能	strBase中のintSplit番目のstrKey前後の文字列をstrOut[0],strOut[1]に返す。	
引数	strBase	分解対象の文字列
	strKey	その前後を分解する文字列
	intSplit	分解するstrKeyの順番数
戻り変数	strOut[0]	分解した前半の文字列
	strOut[1]	分解した後半の文字列
使用例	<pre> <b>strBase = 'happykeyviewkeyend'</b> <b>strKey = 'key'</b> <b>intSplit = 2</b>  =&gt; <b>strOut[0] = 'happykeyview'</b> <b>strOut[1] = 'end'</b> </pre>	

プログラム名	public string strConnect(string[] strParts)	
機能	文字配列strPartsをタブ ( <code>¥¥t</code> )で区切って単一の文字strOutとする。	
引数	strParts	結合する文字配列
戻り変数	(strOut)	結合後の文字列
使用例	<pre> <b>strParts[0] = 'start'</b> <b>strParts[1]= 'middle'</b> <b>srPart[2]= 'end'</b>  =&gt; <b>strOut=start¥¥tmiddle¥¥tend'</b> </pre>	

プログラム名	public List<string> lstConnect(string[] strParts)	
機能	文字配列strPartsを単一のList<string> Genericとする。	
引数	strParts	結合対象の文字配列
戻り変数	List<string>	結合後のList<string>
使用例	<pre> <b>strParts[0] = 'start'</b> <b>strParts[1]= 'middle'</b> <b>srPart[2]= 'end'</b>  =&gt; strOut= 'start;middle;tend'   (;は結合子を概念的に示す) </pre>	

プログラム名	public string findLastPath(string strPath)	
機能	パス名strPathの最後のフォルダまたはファイル名strOutを返す	
引数	strPath	パス名
戻り変数	(strOut)	最後のファイルまたはフォルダ名
使用例	<pre>strPath= 'l://¥homepage¥document¥test.txt'</pre> <pre>=&gt; strOut= 'test.txt'</pre>	



プログラム名	public string setName(string strName, string strCode)	
機能	<p>英名の名前strNameを指定の表記法strCodeに変換する。  表記法：  N (Normal) :E, lbe  R (Reverse) :lbe, E.  strNameがNかRを判定し、strCode(NまたはR)に変換する。</p>	
引数	strName	変換前の名前
	strCode	変換コード
戻り変数	string	変換後の名前
使用例	<pre>for (int i = 0; i &lt; intCoauthor; i++) {     strTemp = strCoauthor[i];     allCoauthor[i] = alpha.setName(strTemp, "R"); }</pre>	