
NAME

TextUtil

SYNOPSIS

```
use TextUtil;

use TextUtil qw(:all);
```

DESCRIPTION

TextUtil module provides the following functions:

AddNumberSuffix, ContainsWhiteSpaces, GetTextLine, HashCode, IsEmpty, IsFloat, IsInteger, IsNotEmpty, IsNumberPowerOfNumber, IsNumerical, IsPositiveInteger, JoinWords, QuoteAWord, RemoveLeadingAndTrailingWhiteSpaces, RemoveLeadingWhiteSpaces, RemoveTrailingWhiteSpaces, WrapText

FUNCTIONS**AddNumberSuffix**

```
$NumberWithSuffix = AddNumberSuffix($IntegerValue);
```

Returns number with appropriate suffix: 0, 1st, 2nd, 3rd, 4th, and so on.

ContainsWhiteSpaces

```
$Status = ContainsWhiteSpaces($TheString);
```

Returns 1 or 0 based on whether the string contains any white spaces.

GetTextLine

```
$Line = GetTextLine(\*TEXTFILE);
```

Reads next line from an already opened text file, takes out any carriage return, and returns it as a string. NULL is returned for EOF.

HashCode

```
$HashCode = HashCode($TheString);
```

Returns a 32 bit integer hash code using One-at-a-time algorithm By Bob Jenkins [Ref 38]. It's also implemented in Perl for internal hash keys in hv.h include file.

IsEmpty

```
$Status = IsEmpty($TheString);
```

Returns 1 or 0 based on whether the string is empty.

IsInteger

```
$Status = IsInteger($TheString);
```

Returns 1 or 0 based on whether the string is a positive integer.

IsPositiveInteger

```
$Status = IsPositiveInteger($TheString);
```

Returns 1 or 0 based on whether the string is an integer.

IsFloat

```
$Status = IsFloat($TheString);
```

Returns 1 or 0 based on whether the string is a float.

IsNotEmpty

```
$Status = IsNotEmpty($TheString);
```

Returns 0 or 1 based on whether the string is empty.

IsNumerical

```
$Status = IsNumerical($TheString);
```

Returns 1 or 0 based on whether the string is a number.

IsNumberPowerOfNumber

```
$Status = IsNumberPowerOfNumber($FirstNum, $SecondNum);
```

Returns 1 or 0 based on whether the first number is a power of second number.

JoinWords

```
$JoinedWords = JoinWords($Words, $Delim, $Quote);
```

Joins different words using delimiter and quote parameters, and returns it as a string.

QuoteAWord

```
$QuotedWord = QuoteAWord($Word, $Quote);
```

Returns a quoted string based on *Quote* value.

RemoveLeadingWhiteSpaces

```
$OutString = RemoveLeadingWhiteSpaces($InString);
```

Returns a string without any leading and trailing white spaces.

RemoveTrailingWhiteSpaces

```
$OutString = RemoveTrailingWhiteSpaces($InString);
```

Returns a string without any trailing white spaces.

RemoveLeadingAndTrailingWhiteSpaces

```
$OutString = RemoveLeadingAndTrailingWhiteSpaces($InString);
```

Returns a string without any leading and trailing white spaces.

SplitWords

```
@Words = SplitWords($Line, $Delimiter);
```

Returns an array *Words* containing unquoted words generated after splitting string value *Line* containing quoted or unquoted words.

This function is used to split strings generated by `JoinWords` as replacement for Perl's core module function `Text::ParseWords::quotewords()` which dumps core on very long strings.

WrapText

```
$OutString = WrapText($InString, [$WrapLength, $WrapDelimiter]);
```

Returns a wrapped string. By default, *WrapLength* is 40 and *WrapDelimiter* is Unix new line character.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

FileUtil.pm

COPYRIGHT

Copyright (C) 2004-2012 Manish Sud. All rights reserved.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.