Contents

1

strutilsxx

This library is a collection of useful additional C++ string processing functions like trimming, replacing, formating, etc.

Author:	Klaus Reimer <k@ailis.de>
Version:	0.7.0

2

string **trimLeft** (const string &sString)

Left trim a string.

Left trim a string. This function removes all spaces and tabs on the left side of the string **sString** and returns the result.

3

string **trim** (const string &sString)

Trim a string.

Trim a string. This function removes all spaces and tabs on the left side and on the right side of the string **sString** and returns the result.

4

string **stripQuotes** (const string &sString)

Remove the enclosing quote characters from a string.

Remove the enclosing quote characters from a string. This function removes the enclosing quote characters from the string **sString** and returns the result.

5

string **replace** (const string &sString, const string &sOld, const string &sNew)

Replace parts in a string.

Replace parts in a string. This function can be used to replace all occurences of the sub string **sOld** in the string **sString** with the new sub string **sNew**. The function returns the resulting string.

6

string **addSlashes** (const string &sSource, const bool bEscape8Bit=false, const string &sEscape="")

Add Slashes to special characters.

Add Slashes to special characters. This function adds backslashes to all special characters in the string **sSource** that needs escaping. If **bEscape8Bit** is set to *true* then all characters greater than character number 127 are escaped by backslashes, too. You can define more characters to be escaped in the string **sEscape**. The function returns the resulting string.

7

string **stripSlashes** (const string &sSource)

Parse string with escaped special characters.

Parse string with escaped special characters. This function converts all escaped characters in the string **sSource** and returns the converted string.

8

string **stripNewLines** (const string &sSource)

Remove carriage return and line feed characters.

Remove carriage return and line feed characters. This function removes all carriage return and line feed characters from the string **sSource** and returns the resulting string.

9

string **upperstr** (const string &sString)

Convert string to upper-case.

Convert string to upper-case. This function converts all characters of the string **sString** to upper-case and returns the resulting string.

10

string **lowerstr** (const string &sString)

Convert string to lower-case.

Convert string to lower-case. This function converts all characters of the string **sString** to lower-case and returns the resulting string.

11

string **intstr** (const long iValue, const int iRadix=10, const int iLen=1)

Convert interger to string.

Convert interger to string. This function converts the integer **iValue** to a string and returns it. Optionally you can specify a different numeric base than 10 with the parameter **iRadix** and you can use the **iLen** Parameter to specify the length of the resulting string (filled with zeroes). For example: If you convert the value *254* with base *16* and length *4* you get a string value of *00fe*.

12

string **uintstr** (const unsigned long iValue, const int iRadix=10, const int iLen=1)

Convert unsigned interger to string.

Convert unsigned interger to string. This function converts the unsigned integer **iValue** to a string and returns it. Optionally you can specify a different numeric base than 10 with the parameter **iRadix** and you can use the **iLen** Parameter to specify the length of the resulting string (filled with zeroes). For example: If you convert the value *254* with base *16* and length *4* you get a string value of *00fe*.

13

string **format** (const char* sString, ...)

Format a string.

Format a string. This function is similiar to the printf() command in C. The first parameter is a format string and all other parameters are values to be included in the format string. The formated string is returned.

14

vector< string > **explode** (const string &sSeparator, const string &sString)

Converts a custom separated string into a string vector.

Converts a custom separated string into a string vector. This function separates all items in **sString** which are separated by **sSeparator** and stores these items in a string vector which is returned. You can use this function to convert a comma-separated string into a list of strings.

15

vector< string > **explodeQuoted** (const string &sSeparator, const string &sString)

Converts a custom separated string into a string vector.

Converts a custom separated string into a string vector. This function is doing the same as the function *explode* but this function honors quotes and escape characters. Separators in quotes or escaped separator-characters are not identified as separators.

16

string **implode** (const string &sGlue, const vector< string > &slPieces)

Converts a string vector into a custom separated string.

Converts a string vector into a custom separated string. This function takes all items from the string vector **slPieces** and glues them together with the string **sGlue**. The resulting string is returned.

17

bool **matchRegex** (const string &sString, const string &sPattern)

Check string against regular expression.

Check string against regular expression. This function checks if the string **sString** matches the regular expression **sPattern** and returns the boolean result.

18

istream& **rgetline** (istream &IFile, string &sLine, char cDelimiter='\n')

Read text file backward.

Read text file backward. This function reads a line from the file **IFile** and stores the line in **sLine**. **cDelimiter** defaults to a carriage return character and can be changed to use an other line delimiter. The different to the STL-function getline() is that this function is reading backward. If you open a file and seek to the end of the file you can use this function to read line by line from the end to the beginning.