



[Home](#) [Contents](#)

## wxWidgets helper classes

wxWidgets consists of a large group of helper classes that help programmers to do their job. These include classes for working with strings, files, XML files, streams, database, or network. Here we will show only a tiny drop of the whole lake.



wxWidgets library can be used to create console and GUI applications. In this chapter, we will illustrate some of the helper classes in console based applications.

### Console

This is a simple console application. The application puts some text into the console window.

console.cpp

```
#include <wx/string.h>
```

```
int main(int argc, char **argv)
{
    wxPuts(wxT("A wxWidgets console application"));
}
```

A wxWidgets console application

This is the output.

## wxString

wxString is a class representing a character string.

In the following example, we define three wxStrings. We create one string of these strings using addition operation.

addition.cpp

```
#include <wx/string.h>

int main(int argc, char **argv)
{
    wxString str1 = wxT("Linux");
    wxString str2 = wxT("Operating");
    wxString str3 = wxT("System");

    wxString str = str1 + wxT(" ") + str2 + wxT(" ") + str3;

    wxPuts(str);
}
```

Linux Operating System

This is the output.

The Printf() method is used to format strings.

## formatted.cpp

```
#include <wx/string.h>

int main(int argc, char **argv)
{

    int flowers = 21;

    wxString str;
    str.Printf(wxT("There are %d red roses."), flowers);

    wxPuts(str);
}
```

There are 21 red roses.

This is the output.

The following example checks, whether a string contains another string. For this we have a `Contains()` method.

## contains.cpp

```
#include <wx/string.h>

int main(int argc, char **argv)
{

    wxString str = wxT("The history of my life");

    if (str.Contains(wxT("history"))) {
        wxPuts(wxT("Contains!"));
    }

    if (!str.Contains(wxT("plain"))) {
        wxPuts(wxT("Does not contain!"));
    }
}
```

```
}
```

Contains!

Does not contain!

This is the output.

The `Len()` method returns the number of characters in the string.

length.cpp

```
#include <wx/string.h>

int main(int argc, char **argv)
{
    wxString str = wxT("The history of my life");
    wxPrintf(wxT("The string has %d characters\n"), str.Len());
}
```

The string has 22 characters

This is the output.

The `MakeLower()` and `MakeUpper()` methods make characters lower case and upper case.

cases.cpp

```
#include <wx/string.h>

int main(int argc, char **argv)
{
    wxString str = wxT("The history of my life");

    wxPuts(str.MakeLower());
    wxPuts(str.MakeUpper());
}
```

the history of my life

THE HISTORY OF MY LIFE

This is the output.

## Utility functions

wxWidgets has several handy utility functions for executing a process, getting a home user directory or getting the OS name.

In the following example, we execute the `ls` command. For this, we have the `wxShell()` function (Unix only).

shell.cpp

```
#include <wx/string.h>
#include <wx/utils.h>

int main(int argc, char **argv)
{
    wxShell(wxT("ls -l"));
}
```

total 40

```
-rwxr-xr-x 1 vronskij vronskij 9028 2007-09-06 22:10 basic
-rw-r--r-- 1 vronskij vronskij  95 2007-09-06 22:09 basic.cpp
-rw-r--r-- 1 vronskij vronskij  430 2007-09-06 00:07 basic.cpp~
-rwxr-xr-x 1 vronskij vronskij 11080 2007-09-05 23:17 console
-rw-r--r-- 1 vronskij vronskij  500 2007-09-05 23:17 console.cpp
-rw-r--r-- 1 vronskij vronskij  485 2007-09-05 23:16 console.cpp~
```

This is the output.

Next we will we will get the home user directory, os name, user name, host name, and total free memory.

system.cpp

```
#include <wx/string.h>
#include <wx/utils.h>

int main(int argc, char **argv)
{
    wxPuts(wxGetHomeDir());
    wxPuts(wxGetOsDescription());
    wxPuts(wxGetUserName());
    wxPuts(wxGetFullHostName());

    long mem = wxGetFreeMemory().ToLong();

    wxPrintf(wxT("Memory: %ld\n"), mem);
}
```

```
/home/vronskij
Linux 2.6.20-16-generic i686
jan bodnar
spartan
Memory: 741244928
```

This is the output.

## Time & date

In wxWidgets, we have several classes for working with date & time.

The example shows current date or time in various formats.

datetime.cpp

```
#include <wx/datetime.h>
```

```
int main(int argc, char **argv)
{
    wxDateTime now = wxDateTime::Now();

    wxString date1 = now.Format();
    wxString date2 = now.Format(wxT("%X"));
    wxString date3 = now.Format(wxT("%x"));

    wxPuts(date1);
    wxPuts(date2);
    wxPuts(date3);
}
```

Fri Sep 7 21:28:38 2007

21:28:38

09/07/07

This is the output.

Next we will show current time in different cities.

datetime2.cpp

```
#include <wx/datetime.h>

int main(int argc, char **argv)
{
    wxDateTime now = wxDateTime::Now();

    wxPrintf(wxT(" Tokyo: %s\n"), now.Format(wxT("%a %T"),
        wxDateTime::GMT9).c_str());
    wxPrintf(wxT(" Moscow: %s\n"), now.Format(wxT("%a %T"),
        wxDateTime::MSD).c_str());
    wxPrintf(wxT("Budapest: %s\n"), now.Format(wxT("%a %T"),
        wxDateTime::CEST).c_str());
    wxPrintf(wxT(" London: %s\n"), now.Format(wxT("%a %T"),
        wxDateTime::WEST).c_str());
    wxPrintf(wxT("New York: %s\n"), now.Format(wxT("%a %T"),
        wxDateTime::EDT).c_str());
}
```

```
}
```

Tokyo: Sat 05:42:24

Moscow: Sat 00:42:24

Budapest: Fri 22:42:24

London: Fri 22:42:24

New York: Fri 16:42:24

This is the output.

The following example shows, how we can add date spans to our date/time. We add one month to the current time.

datespan.cpp

```
#include <wx/datetime.h>

int main(int argc, char **argv)
{
    wxDateTime now = wxDateTime::Now();
    wxString date1 = now.Format(wxT("%B %d %Y"));
    wxPuts(date1);

    wxDateSpan span(0, 1);
    wxDateTime then = now.Add(span);

    wxString date2 = then.Format(wxT("%B %d %Y"));
    wxPuts(date2);
}
```

September 07 2007

October 07 2007

This is the output.



## Files

wxWidgets has several classes to facilitate working with files. This is low level access to files, as opposed to working with streams.

In the following example, we use the `wxFile` class to create a new file and write data to it. We also test, whether the file is opened. Note that when we create a file, it automatically stays as opened.

createfile.cpp

```
#include <wx/file.h>

int main(int argc, char **argv)
{

    wxString str = wxT("You make me want to be a better man.\n");

    wxFile file;
    file.Create(wxT("quote"), true);

    if (file.IsOpened())
        wxPuts(wxT("the file is opened"));

    file.Write(str);
    file.Close();

    if (!file.IsOpened())
        wxPuts(wxT("the file is not opened"));
}
```

```
$ ls quote
```

```
ls: quote: No such file or directory
```

```
$ ./createfile
```

```
the file is opened
```

```
the file is not opened
```

```
$ cat quote
```

```
You make me want to be a better man.
```

This is the output.

The `wxTextFile` is a simple class which allows to work with text files on line by line basis. It is easier to work with this class than with `wxFile` class.

In the next example, we will print the number of lines in a file, first and last lines and finally we will read and show the contents of the file.

readfile.cpp

```
#include <wx/textfile.h>

int main(int argc, char **argv)
{
    wxTextFile file(wxT("test.c"));

    file.Open();

    wxPrintf(wxT("Number of lines: %d\n"), file.GetLineCount());
    wxPrintf(wxT("First line: %s\n"), file.GetFirstLine().c_str());
    wxPrintf(wxT("Last line: %s\n"), file.GetLastLine().c_str());

    wxPuts(wxT("-----"));

    wxString s;

    for ( s = file.GetFirstLine(); !file.Eof();
          s = file.GetNextLine() )
    {
        wxPuts(s);
    }

    file.Close();
}
```

Number of lines: 8

First line: #include <glib.h>

Last line: }

-----

```
#include <glib.h>
```

```
#include <glib/gstdio.h>
```

```
int main() {
```

```
    g_mkdir("/home/vronskij/test", S_IRWXU);
```

```
}
```

This is the output.

The `wxDir` class allows us to enumerate files and directories.

In the following example, we will print all files and directories available in the current working directory.

dir.cpp

```
#include <wx/dir.h>
#include <wx/filefn.h>

int main(int argc, char **argv)
{
    wxDir dir(wxGetCwd());

    wxString file;

    bool cont = dir.GetFirst(&file, wxEmptyString,
        wxDIR_FILES | wxDIR_DIRS);

    while (cont) {
```

```
wxPuts(file);  
cont = dir.GetNext(&file);  
}  
}
```

```
$ ./dir  
dir  
temp  
console  
basic.cpp  
basic  
quote  
createfile  
console.cpp  
basic.cpp~  
test.c  
console.cpp~
```

This is the output.

In this chapter, we have covered some wxWidgets helper classes.



[Home](#) ⇄ [Contents](#) ⇄ [Top of Page](#)

[ZetCode](#) last modified October 13, 2014 © 2007 - 2014 Jan Bodnar