GROMACS - Feature #1635

Proper Unicode support

11/04/2014 08:08 PM - Roland Schulz

Status:	New
Priority:	Normal
Assignee:	
Category:	
Target version:	
Difficulty:	uncategorized

Description

We currently don't have any special treatment for character encoding and we currently don't use Unicode on Windows. Where we use the Windows API we use the narrow version (without W) and it uses the system locale encoding not UTF8. Thus if build, install path or arguments contain characters not representable in the system locale than it doesn't work correctly. I think the correct solution is to always use UTF8. The reasoning is given at http://utf8everywhere.org/ and is implmeneted in http://cppcms.com/files/nowide/ (only susbset is needed). For Unix we don't have to change anything. For Windows we need to always use the wide API and convert all arguments from UTF8 to UTF16. We should also avoid iterating over bytes in string as this fails with UTF8.

History

#1 - 11/05/2014 08:46 AM - Roland Schulz

It is surprising non-obvious how to print UTF8 using printf on Windows. printf with/without _setmode and with/without converting argv to UTF8 doesn't work.
Given argv converted to UTF8 using

```
nowide::args a(argc,argv)
```

One can use wprintf:

```
_setmode(fileno(stdout), _O_U8TEXT); //doesn't matter whether U8 or U16
wprintf(L"%ls\n", nowide::widen(argv[0]).c_str());
```

But that cannot easily be wrapped into some gmx_printf because it requires the special format string (type wchar_t and with the "%ls" format).

Thus probably the best approach is to use:

```
int gmx_printf(const char *fmt, ...)
{
    va_list ap;
    va_start(ap, fmt);
    std::wstring buf = nowide::widen(vformatString(fmt, ap)); //vformatString is the exisiting formatString with va_start replaced with va_copy
    va_end(ap);
    DWORD written = 0, len;
    while(written<buf.size())
    {
        WriteConsoleW(GetStdHandle(STD_OUTPUT_HANDLE), buf.c_str()+written, buf.size()-written, &len, NULL);
        written+=len;
    }
}
```

I think nicer would be

```
nowide::cout << boost::format("%s %s") %argv[0] %argv[1] << std::endl;
```

but that wouldn't be suitable for wrapping into a gmx_printf function. It would be nicer because it is type-safe and doesn't require the allocation loop for vsnprintf (streaming instead).
We could use the nicer option if we reconsidered using iostream for C++ (at least for stdin/stdout).

#2 - 11/06/2014 03:19 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1635.

Uploader: Roland Schulz (roland@rschulz.eu)
Change-Id: Iac984d5a59e25b9d67eb6dad70a8cd73de01b2f0
Gerrit URL: https://gerrit.gromacs.org/4206

#3 - 06/18/2015 08:34 PM - Erik Lindahl

- Tracker changed from Bug to Feature

Changed to feature, since we have never claimed to support anything but ASCII.