

Ryzom - Feature # 123

Status:	Rejected	Priority:	Normal
Author:	kerval	Category:	
Created:	09/22/2008	Assignee:	
Updated:	10/16/2010	Due date:	
Subject:	Adapt NeL code to be fully compatible with Windows Vista/Linux/Mac OS		
Description			
<p>Windows Vista doesn't allow normal users to write in "Program Files" so we could add method for getting a profile folder where NeL could write/modify its files (logs, configs, etc...) and use it when writing logs/configs.</p> <p>We could also add a method for elevating user's privilege to admin when executing an external app.</p> <p>These new methods could also be adapted for Linux and use "~/.appname" folder for all these files.</p> <p>Same thing for Mac OS.</p>			

History

#1 - 09/22/2008 02:59 pm - lubos

kerval wrote:

These new methods could also be adapted for Linux and use "~/.appname" folder for all these files.

Exist new standard:

<http://standards.freedesktop.org/basedir-spec/basedir-spec-0.6.html>

#2 - 09/22/2008 03:39 pm - kerval

lubos wrote:

kerval wrote:

These new methods could also be adapted for Linux and use "~/.appname" folder for all these files.

Exist new standard:

<http://standards.freedesktop.org/basedir-spec/basedir-spec-0.6.html>

So it's related to Gnome and KDE, doesn't it ?

Doesn't exist a kernel function for doing that ?

#3 - 09/22/2008 04:18 pm - lubos

kervala wrote:

```
| Doesn't exist a kernel function for doing that ?
```

No, but glib (gutils.c) implement multiplatform version of `g_get_user_config_dir`, `g_get_user_data_dir` and `g_get_user_cache_dir`

<http://library.gnome.org/devel/glib/unstable/glib-Miscellaneous-Utility-Functions.html#g-get-user-config-dir>

#4 - 05/28/2009 03:07 pm - sfb

- Target version changed from Version 0.7.0 to 0.8.0

Since the importance of this is low (I misunderstood the nature of it based on the vague subject) I will be moving this to 0.8.0. It is possible to merge this with the issue to extract command line parsing into a usable class.

#5 - 08/05/2009 02:58 pm - sfb

Here's an example of how I do this in Werewolf. It may be worthwhile modifying CPath to include a variation of my function. It's a pretty common and accepted way to handle user data on both platforms:

```
// Retrieve the platform specific user data directory.
std::string getUserDataDirectory(std::string appName)
{
    std::string userAppDir;
    #if defined(NL_OS_WINDOWS)
        // User data directory for Windows looks something like:
        // C:\Documents and Settings\myuser\Application Data\ApplicationName
        TCHAR szAppData[MAX_PATH];
        SHGetFolderPath(NULL, CSIDL_PERSONAL, NULL, 0 szAppData);
        userAppDir.assign(szAppData);
        userAppDir += "\\";
        userAppDir += appName;
    #elif defined(NL_OS_MAC)
        // User data directory for Mac OSX looks something like:
        // ~/Library/Application Support/ApplicationName
        userAppDir = getenv("HOME");
        userAppDir += "/Library/Application Support/";
        userAppDir += appName;
    #else // NL_OS_WINDOWS
        // User data directory for Linux looks something like:
        // ~/.ApplicationName
        userAppDir = getenv("HOME");
        userAppDir += "/.";
        userAppDir += appName;
    #endif // NL_OS_WINDOWS
    return userAppDir;
}
```

Note: This is basically not tested, I only have used the Linux variation of this function. But it illustrates the essential idea.

My config path for Debian's Snowball package. Code is based on glib2.0(they have multithred and lazy load support).

```
1
2string CUserPaths::getHomeDir()
3{
4#ifdef NL_OS_WINDOWS
5  nerror ("TODO: Windows home dir");
6  return "";
7#else
8  char *home = getenv("HOME");
9  if (NULL == home || '\0' == home[0])
10 {
11  nerror ("No home dir");
12 }
13  return string(home) + "/";
14#endif
15}
16
17string CUserPaths::getConfigDir()
18{
19#ifdef NL_OS_WINDOWS
20  string configDir;
21  nerror ("TODO: Windows users's config dir");
22#else
23  char *xdgConfigHome = getenv ("XDG_CONFIG_HOME");
24
25  string configDir;
26  if (NULL == xdgConfigHome || '\0' == xdgConfigHome[0])
27  {
28  configDir = getHomeDir() + ".config/";
29  if (!isDir(configDir))
30  {
31    if (0 != mkdir (configDir.c_str(), S_IRUSR | S_IWUSR | S_IXUSR) )
32    {
33      //glib use temp here
34      nerror ("Can't create %s", configDir.c_str());
35    }
36  }
37  }
38  else
39  {
40  configDir = string(xdgConfigHome) + "/";
41  }
42#endif
43  return configDir;
44}
45
46bool CUserPaths::isDir(string path)
47{
48  struct stat stInfo;
```

```
49  if (0 != stat (path.c_str(), &slInfo))
50  return false;
51
52  return bool(S_ISDIR(slInfo.st_mode));
53}
```

#7 - 09/29/2010 09:12 pm - kerval

- *Project changed from NeL to Ryzom*
- *Category deleted (Misc)*
- *Target version deleted (0.8.0)*

#8 - 10/16/2010 10:00 am - kerval

- *Status changed from New to Rejected*

This is too generic, more detailed issues already have been resolved :)