
Subject: Symlink/Shortcut support

Posted by [koldo](#) on Mon, 01 Feb 2010 07:02:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello all

Now symlinks-shortcuts are supported in Core only in Linux with:
FindFile::IsSymLink();

It would be good to have a wider support in Linux and Windows.

The proposal is to add IsSymlink() for both OS and a new function like bool
GetSymLinkPath(const char *linkPath, String &filePath); to get the real path of the symlink.

The implementation could be (some parts has been borrowed from U++):

```
bool IsSymLink(const char *path) {
#ifdef PLATFORM_WIN32
    return GetFileExt(path) == ".lnk";
#else
    struct stat stf;
    lstat(path, &stf);
    return S_ISLNK(stf.st_mode);
#endif
}
bool GetSymLinkPath(const char *linkPath, String &filePath)
{
#ifdef PLATFORM_WIN32
    HRESULT hres;
    IShellLink* psl;
    IPersistFile* ppf;
    CoInitialize(NULL);
    hres = CoCreateInstance(CLSID_ShellLink, NULL, CLSCTX_INPROC_SERVER, IID_IShellLink,
        (PVOID *) &psl);
    if(SUCCEEDED(hres)) {
        hres = psl->QueryInterface(IID_IPersistFile, (PVOID *) &ppf);
        if(SUCCEEDED(hres)) {
            hres = ppf->Load(ToSystemCharsetW(linkPath), STGM_READ);
            if(SUCCEEDED(hres)) {
                char fileW[_MAX_PATH] = {0};
                psl->GetPath(fileW, _MAX_PATH, NULL, 0);
                filePath = FromSystemCharset(fileW);
            } else
                return false;
            ppf->Release();
        } else
            return false;
        psl->Release();
    }
#endif
}
```

```
} else
    return false;
CoUninitialize();
return true;
#else
char buff[_MAX_PATH + 1];
bool ret;
int len = readlink(linkPath, buff, _MAX_PATH);
if (ret = (len > 0 && len < _MAX_PATH))
    buff[len] = '\0';
else
    *buff = '\0';
filePath = buff;
return ret;
#endif
}
```
