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();
```

```
 } 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
}
```

Subject: Re: Symlink/Shortcut support
Posted by mirek on Mon, 01 Feb 2010 13:42:29 GMT
View Forum Message <> Reply to Message

This sounds quite good. Before applying patch, however, I would like to hear more from others about the idea

```
 return GetFileExt(path) == ".lnk";
```

I am not quite convinced that file extension is equivalent to file flag.... (Maybe just give me a couple of hours to think about it

Also, maybe GetSymLinkPath should simply return String and Null in the case of error?

Mirek

Subject: Re: Symlink/Shortcut support
Posted by koldo on Mon, 01 Feb 2010 14:49:03 GMT
View Forum Message <> Reply to Message

I am not quite convinced that file extension is equivalent to file flag.... (Maybe just give me a couple of hours to think about it Smile

Also, maybe GetSymLinkPath should simply return String and Null in the case of error?
Fully agree

Subject: Re: Symlink/Shortcut support
Posted by cbpporter on Mon, 01 Feb 2010 14:54:35 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 01 February 2010 16:49I am not quite convinced that file extension is equivalent to file flag.... (Maybe just give me a couple of hours to think about it Smile

Also, maybe GetSymLinkPath should simply return String and Null in the case of error?
Fully agree
Solution would be to load the .lnk file and see if has the correct format, like the UNIX "file" utility does for files, but I don't think we want to do that.

Subject: Re: Symlink/Shortcut support
Posted by koldo on Mon, 01 Feb 2010 15:24:04 GMT
View Forum Message <> Reply to Message

Quote:Solution would be to load the .lnk file and see if has the correct format, like the UNIX "file" utility does for files, but I don't think we want to do that.In fact this is done in function GetSymLinkPath.

However there is nothing for free... it would require a little time.

Subject: Re: Symlink/Shortcut support
Posted by fudadmin on Mon, 01 Feb 2010 20:40:20 GMT
View Forum Message <> Reply to Message

Hi guys, sorry for big breaks but life sometimes make crude corrections... But I've never stopped working on u++. Hopefully I'll to start posting my contributions if... ok, we will see. And, sorry, I've had symlink in my u++ version for windows for some time.
Some of my code if that helps:

```
bool FindFile::IsShellLinkDir() const //aris002 wrong name? tmp
{
 if (!GetName().EndsWith(".lnk")) //aris002 shell we check ext somewhere outside?
  return false;
// char linkpath[MAX_PATH];
 LPCSTR linkpath=~path;
  LLOG("IsShellLinkDir getname="<<GetName());
 char szGotPath[MAX_PATH];
 char szDescription[MAX_PATH];
```

```
 //   WIN32_FIND_DATA wfd;
 HRESULT hres;
 IShellLink* psl;
 CoInitialize(NULL);
 hres = CoCreateInstance(CLSID_ShellLink, NULL, CLSCTX_INPROC_SERVER,
                         IID_IShellLink, (LPVOID *) &psl);
 if(SUCCEEDED(hres)) {
  IPersistFile* ppf;
  psl->QueryInterface(IID_IPersistFile, (void**)&ppf);
  if(SUCCEEDED(hres)) {

   WCHAR szPath[_MAX_PATH] = { 0 };
   MultiByteToWideChar(CP_ACP, 0, linkpath, (int)strlen(linkpath), szPath, _MAX_PATH);
  //not correct? - might be links with diffrerent extensions
  //get shortcuts extension from registry? how?

   hres = ppf->Load(szPath, STGM_READ); //Load the shortcut.
   if (SUCCEEDED(hres)) {
   // LLOG("ppf->Load ok!! "<<linkpath);
    hres = psl->GetPath((LPSTR)szGotPath, MAX_PATH,
            NULL,  SLGP_UNCPRIORITY );
     LLOG("name "<<GetName()<<" target "<<szGotPath);
//    if(SUCCEEDED(hres))
//     LLOG("SUCCEEDED link ");
    ppf->Release();
   }
  psl->Release();
  }
 }

 CoUninitialize();
 if(SUCCEEDED(hres)) {
  LLOG("SUCCEEDED2 link ");
  return true;
 }
 else
  return false;
 //return SUCCEEDED(hres);
}




bool ResolveShellLink(const String& path, String& linked) //aris002
{
// char linkpath[MAX_PATH];
 LPCSTR linkpath=path;
                LLOG("IsShellLinkDir getname="<<path);
```

```
 char szGotPath[MAX_PATH];
 HRESULT hres;
 IShellLink* psl;
 CoInitialize(NULL);
 hres = CoCreateInstance(CLSID_ShellLink, NULL, CLSCTX_INPROC_SERVER,
                         IID_IShellLink, (LPVOID *) &psl);
 if(SUCCEEDED(hres)) {
  IPersistFile* ppf;
  psl->QueryInterface(IID_IPersistFile, (void**)&ppf);
  if(SUCCEEDED(hres)) {

   WCHAR szPath[_MAX_PATH] = { 0 };
   MultiByteToWideChar(CP_ACP, 0, linkpath, (int)strlen(linkpath), szPath, _MAX_PATH);
  //not correct? - might be links with diffrerent extensions
  //get shortcuts extension from registry? how?

   hres = ppf->Load(szPath, STGM_READ); //Load the shortcut.
   if (SUCCEEDED(hres)) {
   // LLOG("ppf->Load ok!! "<<linkpath);
    hres = psl->GetPath((LPSTR)szGotPath, MAX_PATH,
            NULL,  SLGP_UNCPRIORITY );
     LLOG("name "<<linkpath<<" target "<<szGotPath);
     linked=szGotPath;
//    if(SUCCEEDED(hres))
//     LLOG("SUCCEEDED link ");
    ppf->Release();
   }
  psl->Release();
  }
 }

 CoUninitialize();
 if(SUCCEEDED(hres)) {
  LLOG("SUCCEEDED2 RESOLVED link ");
  return true;
 }
 else
  return false;
 //return SUCCEEDED(hres);
}
```

they are not optimal and might need above or somewhere else switching off the check of target existance if I can remember.
Below the whole files attached.
P.S I've got also adapted for older version of FileSel for those with icon overlays. And, sorry i don't use svn because because that's too dificult for me, esp, merging.
Huge thanks to all of you!

File Attachments
1) Path.h, downloaded 561 times
2) Path.cpp, downloaded 425 times

Subject: Re: Symlink/Shortcut support
Posted by mirek on Mon, 01 Feb 2010 21:29:01 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Mon, 01 February 2010 15:40Hi guys, sorry for big breaks but life sometimes make crude corrections...

Oh, welcome back!

How long was that? Only 3 years?

Mirek

Subject: Re: Symlink/Shortcut support
Posted by mirek on Fri, 05 Feb 2010 09:22:43 GMT
View Forum Message <> Reply to Message

```
 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;
```

Is not a memory leak there? (no release for ppf upon return false).

Mirek

Subject: Re: Symlink/Shortcut support
Posted by mirek on Fri, 05 Feb 2010 09:25:36 GMT
View Forum Message <> Reply to Message

Please check my altered version:

```
String GetSymLinkPath(const char *linkpath)
{
#ifdef PLATFORM_WIN32
 String path;
 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);
    path = FromSystemCharset(fileW);
   }
   ppf->Release();
  }
  psl->Release();
 }
 CoUninitialize();
 return path;
#else
 char buff[_MAX_PATH + 1];
 bool ret;
 int len = readlink(linkPath, buff, _MAX_PATH);
 if(len > 0 && len < _MAX_PATH)
  return String(buff, len);
 return Null;
#endif
}
```

(now in U++, thanks).

Subject: Re: Symlink/Shortcut support
Posted by koldo on Fri, 05 Feb 2010 10:42:51 GMT
View Forum Message <> Reply to Message

Hello Mirek

Thanks. It works OK for me

Please do not forget IsSymlink(). Fudamin way seems to be the most complete.

Subject: Re: Symlink/Shortcut support
Posted by mirek on Sat, 06 Feb 2010 11:49:29 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 05 February 2010 05:42Hello Mirek

Thanks. It works OK for me

Please do not forget IsSymlink(). Fudamin way seems to be the most complete.

What about to recycle GetSymLinkPath there?

Mirek

Subject: Re: Symlink/Shortcut support
Posted by mirek on Sat, 06 Feb 2010 12:20:12 GMT
View Forum Message <> Reply to Message

Like this:

```
bool FindFile::IsSymLink() const
{
 String name = GetName();
 if(GetFileExt(name) != ".lnk")
  return false;
 return !IsNull(GetSymLinkPath(AppendFileName(path, name)));
}
```

Mirek

Subject: Re: Symlink/Shortcut support
Posted by koldo on Sat, 06 Feb 2010 13:30:45 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 06 February 2010 13:20Like this:

bool FindFile::IsSymLink() const

```
{
 String name = GetName();
 if(GetFileExt(name) != ".lnk")
  return false;
 return !IsNull(GetSymLinkPath(AppendFileName(path, name)));
}
```

Mirek
Well. It is not the most efficient possible but it is very valid.

Subject: Re: Symlink/Shortcut support
Posted by mirek on Sat, 06 Feb 2010 18:40:37 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 06 February 2010 08:30luzr wrote on Sat, 06 February 2010 13:20Like this:

```
bool FindFile::IsSymLink() const
{
 String name = GetName();
 if(GetFileExt(name) != ".lnk")
  return false;
 return !IsNull(GetSymLinkPath(AppendFileName(path, name)));
}
```

Mirek
Well. It is not the most efficient possible but it is very valid.

Why? System calls appear to be equivalent for the test and for obtaining the path.

Mirek

Subject: Re: Symlink/Shortcut support
Posted by koldo on Sat, 06 Feb 2010 22:13:24 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 06 February 2010 19:40koldo wrote on Sat, 06 February 2010 08:30luzr wrote on Sat, 06 February 2010 13:20Like this:

```
bool FindFile::IsSymLink() const
{
 String name = GetName();
```

```
 if(GetFileExt(name) != ".lnk")
  return false;
 return !IsNull(GetSymLinkPath(AppendFileName(path, name)));
}
```

Mirek
Well. It is not the most efficient possible but it is very valid.

Why? System calls appear to be equivalent for the test and for obtaining the path.

Mirek
Hello Mirek

Being punctilious these is not necessary:

```
char fileW[_MAX_PATH] = {0};
psl->GetPath(fileW, _MAX_PATH, NULL, 0);
path = FromSystemCharset(fileW);
```

Subject: Re: Symlink/Shortcut support
Posted by mirek on Sat, 06 Feb 2010 22:19:29 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 06 February 2010 17:13luzr wrote on Sat, 06 February 2010 19:40koldo wrote on Sat, 06 February 2010 08:30luzr wrote on Sat, 06 February 2010 13:20Like this:

```
bool FindFile::IsSymLink() const
{
 String name = GetName();
 if(GetFileExt(name) != ".lnk")
  return false;
 return !IsNull(GetSymLinkPath(AppendFileName(path, name)));
}
```

Mirek
Well. It is not the most efficient possible but it is very valid.

Why? System calls appear to be equivalent for the test and for obtaining the path.

Mirek
Hello Mirek

Being punctilious these is not necessary:

```
char fileW[_MAX_PATH] = {0};
psl->GetPath(fileW, _MAX_PATH, NULL, 0);
path = FromSystemCharset(fileW);
```

I have to admit I have only considered fudadmin code.

OK, I will change it...

Mirek

Subject: Re: Symlink/Shortcut support
Posted by mirek on Sat, 06 Feb 2010 22:26:58 GMT
View Forum Message <> Reply to Message

Optimized.

Btw, uppdev/SymLinkTest used for developing this..
