Subject: What improvements for Upp on Linux to comply with freedesktop?
Posted by fudadmin on Fri, 27 Jul 2007 12:36:51 GMT
View Forum Message <> Reply to Message

You might also noticed that some (or many?) upp forum users have their different config functions and improvements and some offering them scattered in different forum places.
And because Mirek's just asked...

this thread should be the answer to:
http://www.ultimatepp.org/forum/index.php?t=msg&&th= 2511&goto=10738#msg_10738

So, who disagrees that freedesktop.org is doing a great job to make some order on Linux and other *X's ? Shell Upp try, at least, not to lag behind the most important and useful freedesktop recommendations?
Have you got any wishes in this area?

P.S. some links:
http://www.freedesktop.org/wiki/Specifications?action=show&a mp;redirect=Standards
http://www.freedesktop.org/wiki/Specifications/XSettingsRegi stry

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by fudadmin on Fri, 27 Jul 2007 15:59:47 GMT
View Forum Message <> Reply to Message

Well, first of all, I think, it's worse than a joke when Upp makefile does some kind of "installation" and copies files onto Linux systems leaving users with a manual tracing of installed files in case they want to uninstall...

Also, if a user tries compiling all U++ examples plus some of his own, it becomes easier to find a needle in a rubbish truck than to clean his home directory without damaging some of his useful configuration.

So, some of my proposals (regarding clean order on Linux) are:
1. First and the most important - keep all (to one Upp version related) files under ONE Upp configs directory with subdirectories for each app and NOT scattered all over users HOMEDIR!
2. According to some freedesktop recommendations, e.g:
==> /home/username/.config/upp/appname/*
and/or, because some of us have several versions of upp:
==>/home/username/.config/upp_version_name/appname/*
(open /home/username/.config/ to check which open source projects have already switched)

3. This is achieved easily with a change I've for my version (Core/App.cpp line ~123):
```
String  ConfigFile(const char *file) {
#if defined(PLATFORM_WIN32)
        return GetExeDirFile(file);
#elif defined(PLATFORM_POSIX)
```

```
	String p = GetHomeDirFile(".config/upp/" + GetExeTitle());
```
Ok, that is not ideal. The subdirs should be programmable, too. At least, upp_version_name. In fact, I really want that Upp users had an easy possibility to have their own versions to experiment and easy switching but that can wait a few days...

4. Similar functions to Zardos:
http://www.ultimatepp.org/forum/index.php?t=msg&&th= 2511&goto=10729#msg_10729
should be discussed in the context of above.
I also have:
(they work for me on my Linux but must be corrected...)

```
String LoadAppConfigDirFile(const char *filename);
String LoadUPPConfigDirFile(const char *filename);

String GetAppConfigDirFile(const char *filename);
String GetUPPConfigDirFile(const char *filename);

bool SaveAppConfigDirFile(const char *filename, const String& data);
bool SaveUPPConfigDirFile(const char *filename, const String& data);
//====
String LoadAppConfigDirFile(const char *filename)
{
 return LoadFile(GetAppConfigDirFile(filename));
}

String LoadUPPConfigDirFile(const char *filename)
{
 return LoadFile(GetUPPConfigDirFile(filename));
}

String GetAppConfigDirFile(const char *filename)
{
 return Environment().Get("UPP_ARIS_CONFIG")+"/"+ GetExeTitle()+"/"+filename;
}

String GetUPPConfigDirFile(const char *filename)
{
 return Environment().Get("UPP_ARIS_CONFIG")+"/"+filename;
}

bool SaveAppConfigDirFile(const char *filename, const String& data)
{
 return SaveFile(GetAppConfigDirFile(filename), data);
}

bool SaveUPPConfigDirFile(const char *filename, const String& data)
{
 return SaveFile(GetUPPConfigDirFile(filename), data);
}
```

5. Maybe a separate logs subdir in each Upp app subdir?
6. Maybe more programmable configs flexibility in general?
7. A separate Utils-Configs-Settings package to keep with ever changing freedesktop and distros spirit?

I'm not saying that everything must be accepted. (Personally, I can keep them in my own libs and be happy.
But if many Upp users waste time on the same mundane things then maybe it's better to change something globally?

Just some thoughts. Perhaps, others opinions would be more help.

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by ebojd on Fri, 27 Jul 2007 17:04:37 GMT
View Forum Message <> Reply to Message

It would also be productive to try to conform to something like the Linux Standard Base (LSB) for installation locations, guidelines, etc.

The single .config path for applications is really nice. I would prefer to see the versioning modeled after KDE:

/home/${USERNAME}/.config/upp/${VERSION}/appname/*

That way all UPP files regardless of version can be maintained. I simply skimmed the URL listed and may read them in detail later. Another thing I would like to interject here (which may or may not be covered in the suggested standards) is to have a single directory tree for each config which are *overloadable* by the user configs. Example, the location of the uppsrc tree:

/usr/share/upp/2007.1/.config/uppsrc.var

```
UPP = "/usr/share/upp/2007.1/uppsrc";
OUTPUT = "/old_root/upp/devout";
COMMON = "/usr/share/upp/2007.1/Common";
```

which is then oveloaded with any settings in:

/home/username/.config/upp/2007.1/uppsrc.var

```
OUTPUT = "/home/${USERNAME}/upp/devout";
```

I would also like to see the addition of BIN_DIR, LIB_DIR, DATA_DIR, and CONFIG_DIR to help

manage installation of binary, library, data and config files.

When I started on of my current projects for which I choose U++ I had started writing all my code using GNU programming standards. Unfortunately, U++ and GNU standards do not mix well (like naming conventions). There are, however, some guidelines which would be good to glean from them such as *all* makefiles which include an install directive *must* also include an uninstall.

I have also mentioned wishing an extension (for package organizer) that allows developers to allow pkg-config to be used. Something similar to:

  LINUX: `pkg-config --libs x11`

More later I have to go...

  EBo --

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by Zardos on Fri, 27 Jul 2007 17:24:44 GMT
View Forum Message <> Reply to Message

Regarding config files in general:

Comming mainly from windows I seperate storing a config file and loading a config file.

There are three main locations for a config file:

1.) The user profile app directory (home/<user>/.<app> under unix)
2.) An all user profile app directory (etc/<app> under unix)
3.) The application directory <install_path>/config (same on unix)

... when loading a config file the three locations above are searched for a config file in the order 1, 2, 3:

If a config file is not found in the user profile (home), all user profile is checked for a config file (etc). If not found there the app directory is checked for a config file.

This way you can:

1.) Deliver some default settings in the app directory
2.) A corporation can place a coporate config file in the all user profile (etc) - without fear it is overwritten with a new version of the app.
3.) Users can overwrite the default application settings and the coporate settings

=> A config file is always saved in the user profile (home).

While this of course does not match all possible scenarios it is ok for most common cases.

... All this does not solve the problem that a new version may have an incompatible new config file format. So this have to be handled seperately by the app. For examples serializing maps instead of member vars directly.

Finally for windows it is sometimes important to make use of "user local settings" (for example temp is located at this place). Under windows in a corporate network the user profile is often replicated via network when the user logs on to different workstations. And usually there is a quota on the user profile (always to small 10MB at our place). So for "cache data" like a browser cache it is important to store the data NOT in the normal user profile. Instead the local user profile should be used.

Does this map to /var/local/app/<user> under unix?

Never store config file in the app directory (defaults are ok). Even under windows it is possible that more users share one computer...

This is the reason why I have four functions for config management.

Last Edit: I have a special "usbstick / pendrive mode". All above is ignored and everything is stored in the app directory -> Keep the app with all you settings on the pendrive.

- Ralf

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by fudadmin on Fri, 27 Jul 2007 23:23:21 GMT
View Forum Message <> Reply to Message

Quote:The single .config path for applications is really nice. I would prefer to see the versioning modeled after KDE:

/home/${USERNAME}/.config/upp/${VERSION}/appname/*
Thanks, Ebo! I didn't dare alone to make an accent on this... Shell we vote? Who against?

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by mr_ped on Sat, 28 Jul 2007 09:07:33 GMT
View Forum Message <> Reply to Message

$version of Upp?
It may be handy for development, but I can't see how the ordinary user would profit from this, if he would lose his settings with every new version of Upp software.

And does this apply also on final applications?
Or will there be some easy mechanism how to import older settings into new version?

I'm not sure if I do get you right and fully understand how the Upp should work and how applications build upon Upp should work to take advantage of this.

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by ebojd on Sat, 28 Jul 2007 19:36:52 GMT
View Forum Message <> Reply to Message

> Or will there be some easy mechanism how to import older
> settings into new version?

The programs which have provided this level of control typically have a mechanism which will copy the last versions config info to the new version the first time the new version is run -- signified by no config files for that version. So, this is a question of "bootstrapping". Another option is to ask the use if they want the defaults or to copy over the old config info. There can be annoyances/issues with doing either (such as should we override the UPPSRC and COMMON variables or keep the old ones... I'll write more on that if people are interested in that discussion which is likely to be rather long and tedious.

> I'm not sure if I do get you right and fully understand
> how the Upp should work and how applications build upon
> Upp should work to take advantage of this.

I can see it going either way. Should the application keep it's own version configuration tree,

/home/${USERNAME}/.config/upp/${VERSION}/appname/${APPVER}/*

or will it suffice to have the application build relative to a distribution of U++:

/home/${USERNAME}/.config/upp/${VERSION}/appname/*

For most cases I think that the latter will suffice because it allow the developer to (re)build the application against a specific version of the U++ source tree. From a users perspective they simply rebuild the applications when they update U++.

The real advantage of this for the developer is that they can build/test/maintain applications across multiple versions and recreate a clients configuration. Have you ever updated U++ to discover that one of the API's had changed and your code will no longer compile? I have... From the users perspective the program hast to be rebuilt if they want to use the new code anyway, so there should be no requirements beyond that from the user.

If we are going to pick this one apart, can anyone suggest a specific application so we can come up with some concrete examples?

EBo --

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by mirek on Wed, 01 Aug 2007 08:46:51 GMT
View Forum Message <> Reply to Message

Just a note about version...

As long as config is managed using Serialize(Stream&), I usually tend to version this Serialize, which means it is usually upward compatible.

In the same time, there are compatibility checks in the code, which means wrong config file is refused.

So far, I little cared about this in my Win32 applications, using single config file place for years (literally - some of .cfg files in IDIS were indeed created first in 1999 . Sometimes configuration is lost when uploading a new version of software, but I guess this is something that is to be expected anyway.

If I understand issues well, I see the advantage of $VERSION only if you consider downgrading software, or perhaps running several versions at once, correct?

(Note: This is just a description of things as they are, not an argument against $VERSION).

Mirek

Subject: Re: What improvements for Upp on Linux to comply with freedesktop?
Posted by ebojd on Wed, 01 Aug 2007 12:14:56 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 01 August 2007 03:46
If I understand issues well, I see the advantage of $VERSION only if you consider downgrading software, or perhaps running several versions at once, correct?

(Note: This is just a description of things as they are, not an argument against $VERSION).

Mirek

I'll have to think about it a little, but I see it mainly for multiple version (especially creating development sandboxes) and downgrading. There are some nice tricks that can be done with overwriting system configurations at runtime, but this is a subtle thing.

Hopefully more later,

EBo --