
Subject: Updater package

Posted by [mdelfede](#) on Fri, 14 Jan 2011 22:49:51 GMT

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

Well... the name could also be WebUpdater.

It's a package for installing/updating applications automagically.

No need for installers, but you can have them

It's enough to include the package just after GUI_APP_MAIN :

```
GUI_APP_MAIN
```

```
{  
  Updater updater;  
  updater  
    .SetMaxVersion(GetMaxVersion())  
    .SetWebRoot("www.timberstruct.com/webupdater/updatertest")  
    .UpdateManual();  
  
  if(!updater.Run())  
    return;
```

```
-----FOLLOWING APPLICATION CODE FROM HERE -----
```

```
}
```

The web server layout is :

```
[WebRoot]/[AppName]/LINUX32/versions  
[WebRoot]/[AppName]/LINUX32/[version1]/[AppName]  
[WebRoot]/[AppName]/LINUX32/[version2]/[AppName]  
.....  
[WebRoot]/[AppName]/LINUX32/[versionN]/[AppName]
```

```
[WebRoot]/[AppName]/LINUX64/versions  
[WebRoot]/[AppName]/LINUX64/[version1]/[AppName]  
[WebRoot]/[AppName]/LINUX64/[version2]/[AppName]  
.....  
[WebRoot]/[AppName]/LINUX64/[versionN]/[AppName]
```

```
[WebRoot]/[AppName]/WINDOWS32/versions  
[WebRoot]/[AppName]/WINDOWS32/[version1]/[AppName]  
[WebRoot]/[AppName]/WINDOWS32/[version2]/[AppName]  
.....  
[WebRoot]/[AppName]/WINDOWS32/[versionN]/[AppName]
```

```
[WebRoot]/[AppName]/WINDOWS64/versions  
[WebRoot]/[AppName]/WINDOWS64/[version1]/[AppName]  
[WebRoot]/[AppName]/WINDOWS64/[version2]/[AppName]  
.....  
[WebRoot]/[AppName]/WINDOWS64/[versionN]/[AppName]
```

'versions' file contains a list of available file versions in double format xxxx.xx (major + 2 digits minor version number) one for each line.

On launch, the application will (if enabled) look for an updated version on web server, ask user (or not, if autoupdate is enabled) if he wants to update, copy itself on a temporary folder, respawn in superuser mode (POSIX), fetch the application code and replace the older one. It will then store the current app version in a system-wide config file AND re-run the updated application as a previous normal user.

It supports version limiting, i.e., if there's 2.00 version available but user is allowed up to 1.6, it's enough to do

```
updater.SetMaxVersion(1.6)
```

and updater will limit to this one at most.

Code SHOULD (and WILL !) be cross platform, by now I tested it on Linux64 bit only. It's still missing icon and menu embedding; if somebody knows how to do it in Windows, I'll appreciate his help !

Ciao

Max

p.s: the only limitation is that updater gets his paths from application name. So, if you want your app to be installed as "TimberStruct" the executable MUST be named TimberStruct[.exe]. The executable can also be a simple stub with just the updater code; on launch it will download from web the "true" application, on newest version; it can so be deployed as a very small installer app.

If run with --UNINSTALL as first option, it will uninstall the app, along with his system config folder.

Feel free to test it, but beware... Windows is completely untested by now!

p.p.s : if you want to test, beware that I've got testfiles on my server jsut for LINUX64 by now... on next days I'll add WINDOWS versions too, hopefully.