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 sa 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.

Subject: Re: Updater package
Posted by mdelfede on Sat, 15 Jan 2011 21:28:54 GMT
View Forum Message <> Reply to Message

Updater now works also in windows.
Test files are on server, so you can test it in both environment (LINUX64 and WINDOWS32).

Ciao

Max

TEST INSTRUCTIONS :

1) Build and run UpdaterTest application
2) It will ask for install, answer Yes
3) App will download version 1.5 from server (notice that in test app the maximum version is limited to 1.6)
4) Launch the installed app (NOT the one you built), it'll notice that a version 2.0 is available and ask for update
5) Answer yes and you'll get the 2.0 version installed
6) Launch again the application, it'll just run and show 2.0 version, as it's the last one on server
7) run 'UpdaterTest --UNINSTALL' and the app will be completely removed, besides user configuration folder that could contain user data.

if you rebuild the test app changing the SetMaxVersion() from 1.6 to something greater than 2.0, you'll notice that it'll download the 2.0 directly.
SetMaxVersion() has the purpose to work together with a license manager (Protect stuff, here) when the user isn't allowed to pass some version number.

Still no docs by now, and still no shell embedding... I'd appreciate some help to add this one for windows platform

Ciao

Max

Subject: Re: Updater package
Posted by mdelfede on Sun, 16 Jan 2011 20:57:12 GMT
View Forum Message <> Reply to Message

Shell embedding for Linux is ready

It was quite long because of ill-brained theming, but now, on install, the app is embedded inside menu, there's a mimetype icon for files and an application icon, independently of installed theme.
On uninstall, all is rolled back.

Now it's the windows part....

Ciao

Max

Subject: Re: Updater package
Posted by mdelfede on Mon, 17 Jan 2011 00:42:29 GMT
View Forum Message <> Reply to Message

And now for Windows is also almost all working.
Menu and icons are setup correctly, it still miss icon for file associations.

I'd appreciate if somebody could test it on Windows7, I did only on XP.

Ciao

Max

Subject: Re: Updater package
Posted by koldo on Mon, 17 Jan 2011 11:26:13 GMT
View Forum Message <> Reply to Message

Thank you Massimo.

I think it will be useful for me in the next future.

Subject: Re: Updater package
Posted by mdelfede on Mon, 17 Jan 2011 11:41:08 GMT
View Forum Message <> Reply to Message

You're wellcome... but I guess that it'll need more work to run on Vista+ os... They changed much on protection side.

Ciao

Max

Subject: Re: Updater package

Well, finally the Updater is running ok on both Linux and Windows.
It should work from Xp to Windows7 and in most Linux flavours.
Here a sample code :

```
GUI_APP_MAIN
{
 Updater updater;
 updater
  // accept 1.6 as maximum downloadable version
  .SetMaxVersion(1.6)

  // setup web root for program downloads
  .SetWebRoot("www.timberstruct.com/webupdater/updatertest")

  // ask for updates on launch
  .UpdateManual()

  // program menu goes in "Office" cathegory
  // (Linux only by now, no harm on windows)
  .SetCathegory("Office")

  // helper comment for application
  // (shows in associated files properties)
  .SetComment("A sample test program")

  // associates .xxx and .yyy extensions to this application
  // and sets application icons for this files
  .SetExtensions(Vector<String>() << "*.xxx" << "*.yyy")

  // setup the icon for application
  // (here Updater default icon)
  .SetIcon(updater.DefaultIcon())

  // we want a desktop icon too
  .DesktopIcon();

 // run updater -- DON'T change this !
 if(!updater.Run())
  return;

 // run the default prompts -- see Updater source code
 // if you need some fine-grained control
 if(!updater.DefaultPrompts())
  return;
```

```
 // here the rest of your application.....
 PromptOK("App version is 1.0");

}
```

As you can see, the Installer/Updater is completely transparent to your code; it's enough to add above lines just after GUI_APP_MAIN and setup a web repository (instruction on first post)...
That's all. Once run your application will install itself and on each run will check the webserver for updates, installing them on requests.
It handles file associations, icons and menu items, uninstaller (windows) and does a complete system cleanup on uninstall.

It would be trivial to add installation of data files if requested, but for now it's enough

Ciao

Max

Subject: Re: Updater package
Posted by mdelfede on Thu, 27 Jan 2011 08:45:43 GMT
View Forum Message <> Reply to Message

A small change, now there's a ProgramVersion class holding version numbers. It allows Major.Minor.Devel version schemas.
It causes a small change in demo app :

```
// accept 1.6 as maximum downloadable version
updater.SetMaxVersion("1.6.1234");
```

instead of

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

Also server layout changes a bit, with version subfolders in form of

/MMM.mm.ddd/

where M are major version numbers, m minor ones and d developmend.
Major and minor always present, devel only on request.
So :

..../1.0/
..../1.5/
..../1.67/
..../1.5.432/

are correct version numbers/folders.
Notice that the number is NOT a floating point number as before, so :
ProgramVersion("1.6.2") < ProgramVersion("1.6.15")
ProgramVersion("1.5") < ProgramVersion("1.22")

Ciao

Max

Subject: Re: Updater package
Posted by koldo on Wed, 12 Oct 2011 04:37:54 GMT
View Forum Message <> Reply to Message

Hello Massimo

I wanted to use your Updater with your Protect packages.

Some questions:

- Now Bazaar/Updater has the same name that Updater package, so I cannot get the Bazaar one from a package kept outside Bazaar. How can I overcome it?

- It seems that Updater is also an installer package as it:
-- Download the files
-- Install the program
-- Creates Program Files folder and Application Data folder

Would it be possible to choose the Program Files folder location and name, and the Application Data folder name?

- Could it work from LAN folders too? (this is a must for me now)

- The uninstaller is run from a program with a name from GetTempFileName() that appears in window caption. Could you use just the program name or with something like " - Uninstaller" added?

- In my case the uninstaller do not remove the Application Data folder. May this be switched on or off? (to ask the user if s/he wants to delete data folder)

These are just the first set of questions . Thank you. Updater seems VERY complete.

Subject: Re: Updater package
Posted by mdelfede on Wed, 12 Oct 2011 08:22:51 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 12 October 2011 06:37Hello Massimo

I wanted to use your Updater with your Protect packages.

Some questions:

- Now Bazaar/Updater has the same name that Updater package, so I cannot get the Bazaar one from a package kept outside Bazaar. How can I overcome it?

I guess you can, just add Bazaar nest to your working assembly.
I do it always in my app... I've a TimberStruct assembly with TimberStruct, Bazaar and UppSrc nests inside and it works.

Quote:
- It seems that Updater is also an installer package as it:
-- Download the files
-- Install the program
-- Creates Program Files folder and Application Data folder

Yep, it is. you can (as I did...) deploy a small programm that contains just updater and setup to fetch your real app.
When user launches it first time, he gets the true application downloaded and installed, in its latest version, so you can leave the installer forever on your web site without worrying updating it.

Quote:
Would it be possible to choose the Program Files folder location and name, and the Application Data folder name?

It would need some changes.... you can do it, but it should maintain the actual defaults to not break the apps using it now.
You could add a "SetProgramFilesFolder()" and "SetAppDataFolder()" member functions that could change the paths from the default.

Quote:
- Could it work from LAN folders too? (this is a must for me now)

Uhmmmm.... never tried. You can try to install the Server app onto a PC in your lan and see.... it's a quick test.

Quote:
- The uninstaller is run from a program with a name from GetTempFileName() that appears in window caption. Could you use just the program name or with something like " - Uninstaller" added?

ops, yes... I guess I forgot to set the title.

Quote:
- In my case the uninstaller do not remove the Application Data folder. May this be switched on or off? (to ask the user if s/he wants to delete data folder)

that could also be added as an extension, not too difficult.
I'm sorry that I can't work on it on these days... but you can do it, just be sure to not break actual behaviour, or my customers will kill me

Quote:
These are just the first set of questions . Thank you. Updater seems VERY complete.

Thank you... I guess it could be extended, indeed, The most important missing thing is the ability to update other files than the executable itself; it should be a trivial addition too, indeed.

Ciao

Max

Subject: Re: Updater package
Posted by koldo on Wed, 12 Oct 2011 11:42:16 GMT
View Forum Message <> Reply to Message

Hello Massimo

Quote:I guess you can, just add Bazaar nest to your working assembly.I think that to do it I would have to put first Bazaar and after that uppsrc, but I prefer the opposite: to have uppsrc first. This way it always put uppsrc Updater instead of Bazaar Updater. I tried to put Bazaar/Updater but it does not work.

Quote:Yep, it is. you can (as I did...) deploy a small programm that contains just updater and setup to fetch your real app.
When user launches it first time, he gets the true application downloaded and installed, in its latest version, so you can leave the installer forever on your web site without worrying updating it.So:
- You deploy an installer with Updater inside.
- Updater downloads the application, that also has Updater inside

Quote:It would need some changes.... you can do it, but it should maintain the actual defaults to

not break the apps using it now.It is an honor . Of course compatibility is a must.

Quote:You could add a "SetProgramFilesFolder()" and "SetAppDataFolder()" member functions that could change the paths from the default.Great. SetProgramFilesFolder() will be filled by the installer (asking the user). I may do an installer soon with a look and feel similar to InstaShield or Nullsoft.

Quote:Uhmmmm.... never tried. You can try to install the Server app onto a PC in your lan and see.... it's a quick test.A focus could be to create a local Licenses Server running in a LAN handling all customer licenses.

Other focus could be to update directly from LAN folders like uppsrc/Updater does, but following your versions standard about folder names.

Quote:The most important missing thing is the ability to update other files than the executable itselfI will check it, following an interface as similar as possible to uppsrc/Updater but without breaking compatibility (or asking you before if a have a good idea).

In addition to this, and recognizing that the security options to save a key are in order:
1. A Web server
2. An USB dongle
3. A local encrypted file including key and data to identify computer hardware

... could it be added the option 3 to Protect? . It could be done this way (you may think something better):
- First time the installer is run, it gives a code and gives the option to send an email with it. This code includes the key you put with ProtectEncrypt and local computer info in a hash code or encrypted (if encrypted you can get that data, for example to monitor deep hardware changes that void the key but that can be permitted so you will send a new key).
- You return the user a key including a hash of the code and additional data as the license end date.

This is because in my case the application is expected to work without dongle and without internet connexion.

In addition, other thing to add to Updater and client access to get a key in Web, is the local Proxy handling, very usual in company environments.

Subject: Re: Updater package
Posted by mdelfede on Wed, 12 Oct 2011 12:21:47 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 12 October 2011 13:42
.....
3. A local encrypted file including key and data to identify computer hardware

... could it be added the option 3 to Protect? . It could be done this way (you may think something

better):
- First time the installer is run, it gives a code and gives the option to send an email with it. This code includes the key you put with ProtectEncrypt and local computer info in a hash code or encrypted (if encrypted you can get that data, for example to monitor deep hardware changes that void the key but that can be permitted so you will send a new key).
- You return the user a key including a hash of the code and additional data as the license end date.

This is because in my case the application is expected to work without dongle and without internet connexion.

In addition, other thing to add to Updater and client access to get a key in Web, is the local Proxy handling, very usual in company environments.

Well, I was planning a similar protection schema on the beginning, but I didn't find any *reliable* way to get a key from user's machine. For "reliable" I mean that :
1) It don't change if user adds/replaces an hard disk or and usb disk
2) It don't allows easily to user to fake it, for example, running app into vmware or similar....

The other remaining option, the dongle, should be not too difficult to implement, besides the fact that it must be multiplatform....

Max

Subject: Re: Updater package
Posted by koldo on Wed, 12 Oct 2011 13:35:37 GMT
View Forum Message <> Reply to Message

Hello Massimo

I can send you a proposal based in data from SysInfo. It has been tested in about thirty computers with XP and 7.

Subject: Re: Updater package
Posted by koldo on Thu, 13 Oct 2011 11:06:40 GMT
View Forum Message <> Reply to Message

Hello Massimo

To identify an equipment and user I use these functions:

GetSystemInfo(manufacturer, productName, version, numberOfProcessors, mbSerial);
GetBiosInfo(biosVersion, biosReleaseDate, biosSerial);

Array <NetAdapter> netAdapter = GetAdapterInfo(); // Includes MAC addresses for all adapters (WAN, LAN, WLAN, ...)
GetHDSerial();
GetUserName();
From them I use this data. These are the results:

- manufacturer: Good although sometimes empty in PC clones
- productName: Good although sometimes empty in PC clones
- mbSerial: Good although sometimes empty in PC clones
- biosVersion: Good although sometimes empty in PC clones
- biosSerial: Good although sometimes empty in PC clones
- mac: Good using "ETHERNET" adapter. Other adapters can change unexpectedly
- hdserial: Good although code messed up in some Windows 7 PCs 64 bits
- userName: Good although easily changed for home users

As you imagined the identification is not perfect, but it is good in professional environments.

Subject: Re: Updater package
Posted by mdelfede on Thu, 13 Oct 2011 13:10:34 GMT
View Forum Message <> Reply to Message

mhhhh.... I don't trust it too much, and it's usually defeated using some emulator like vmware and also in wine....

Btw, the problem of changing an hard disk / network card is there, and you can't do anything about it.
If customer is "smart" he can tell you he changed something and ask for a new auth code, and you can't say simply no....

Max

Subject: Re: Updater package
Posted by koldo on Thu, 13 Oct 2011 13:50:53 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Thu, 13 October 2011 15:10mhhhh.... I don't trust it too much, and it's usually defeated using some emulator like vmware and also in wine....

Btw, the problem of changing an hard disk / network card is there, and you can't do anything about it.
If customer is "smart" he can tell you he changed something and ask for a new auth code, and you can't say simply no....

Max

Hello Massimo

It is true. Anyway if when asking for the key the user sends you all that info (encrypted):
- you know if it is run on a virtual system. You can deny it
- If they say they changed the HD, you will see everything unchanged but the HD code
- Nowadays all PCs have an embedded network hardware so you have a MAC. It is of course possible to change the MAC, but computers with the same MAC cannot live in the same network .

They can purchase PC clones with the same MAC unconnected to LAN or Internet... well but different HD so ... . It is very difficult to handle.

Because of it I think it is worthwhile to have the key in a file. Many people does it alone or combining it with a web authentication just when Internet is available.

Subject: Re: Updater package
Posted by koldo on Thu, 13 Oct 2011 20:02:23 GMT
View Forum Message <> Reply to Message

Hello Massimo

SetAppDataFolder(), SetApplicationFolder(), SetSystemConfigFolder() are added, and improved compatibility with MinGW.

Probably tomorrow file repository will be added. (all of it very simple)

Subject: Re: Updater package
Posted by mdelfede on Thu, 13 Oct 2011 20:20:01 GMT
View Forum Message <> Reply to Message

koldo wrote on Thu, 13 October 2011 22:02Hello Massimo

SetAppDataFolder(), SetApplicationFolder(), SetSystemConfigFolder() are added, and improved compatibility with MinGW.

Probably tomorrow file repository will be added. (all of it very simple)

Hi again, Koldo !

I'll test it on weekend, rebuilding my app with it.
I can't risk doing it on working days, my customer would kill me if something breaks !

Ciao

Max

Subject: Re: Updater package
Posted by koldo on Fri, 14 Oct 2011 08:41:28 GMT
View Forum Message <> Reply to Message

Thank you Massimo

I will update it this afternoon early.

A thing I would appreciate is changing the package name.

It is not good in U++ to have two packages and classes with the same name.

Subject: Re: Updater package
Posted by mdelfede on Fri, 14 Oct 2011 12:59:30 GMT
View Forum Message <> Reply to Message

Well, I don't see a big problem keeping both with same name, usually you don't use them together, but anyways... if you have a good suggestion for name, it's wellcome

p.s.: did you try also my Uniq package ?

Max

Subject: Re: Updater package
Posted by koldo on Sun, 16 Oct 2011 14:24:43 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 14 October 2011 14:59Well, I don't see a big problem keeping both with same name, usually you don't use them together, but anyways... if you have a good suggestion for name, it's wellcome

p.s.: did you try also my Uniq package ?

Max

Hello Massimo

Perhaps AppUpdater?.

About Uniq, now I understand it. However in my case I do not know if it is better to have only one instance or as many instances as the user wants.

Please do not forget to check Protect packages changes .

Subject: Re: Updater package
Posted by mdelfede on Fri, 21 Oct 2011 13:07:49 GMT
View Forum Message <> Reply to Message

:
zlib

Version:
2.22.0 (November 6th, 2011)

Description:
The AngelCode Scripting Library, or AngelScript as it is also known, is an extremely flexible cross-platform scripting library designed to allow applications to extend their functionality through external scripts. It has been designed from the beginning to be an easy to use component, both for the application programmer and the script writer.

Efforts have been made to let it call standard C functions

Subject: Re: Updater package
Posted by mdelfede on Fri, 21 Oct 2011 19:38:47 GMT
View Forum Message <> Reply to Message

and C++ methods with no need for proxy functions. The application simply registers the functions, objects, and methods that the scripts should be able to work with and nothing more has to be done with your code. The same functions used by the application internally can also be used by the scripting engine, which eliminates the need to duplicate functionality.

For the scrip