

Ryzom - Feature # 1153

Status:	Assigned	Priority:	Normal
Author:	rti	Category:	OS: Mac
Created:	11/02/2010	Assignee:	rti
Updated:	02/06/2011	Due date:	
Subject:	Make ryzom_client run completely in an .app bundle		
Description			
Related issues:			
related to Ryzom - Feature # 1217: Per user config/logs for OVQT and GEQT app...	New	12/12/2010	

History

#1 - 11/02/2010 11:18 pm - rti

- % Done changed from 0 to 50

See [[MacBinaries]] for a first test.

#2 - 11/03/2010 07:09 pm - rti

- Status changed from New to Resolved

- % Done changed from 50 to 100

Applied in changeset r1090.

#3 - 11/03/2010 07:14 pm - rti

- Assignee set to rti

- Target version set to Version 0.8.0

#4 - 11/05/2010 12:15 am - rti

- % Done changed from 100 to 80

After starting the .app from one location, that path to client_default.cfg is written into client.cfg.

So if now the .app is moved, it will not start up anymore because the path to client_default.cfg is not valid anymore...

This should be fixed!

#5 - 11/05/2010 12:16 am - rti

- Status changed from Resolved to Assigned

#6 - 11/05/2010 09:08 am - kervala

This night I thought about something :)

- Use the client_default.cfg.in to use a token such as <BundlePath> (that's just a proposition, but you can change this name) in PreDataPath and DataPath
- Copy the generated client_default.cfg to Bundle
- In CClientConfig::init, when creating client.cfg, put the real path to client_default.cfg (in Bundle) for RootConfigFilename, if the client.cfg is already

created, update the RootConfigFilename variable to client_default.cfg (in Bundle), so when you launch Ryzom, it will always use the right client_default.cfg

- When adding PreDataPath and DataPath to search paths, replace all <BundlePath> by the Bundle real path, so absolute, relative and relative from Bundle paths should be working :)

#7 - 11/09/2010 10:50 am - vl

Seems quite complex.

Another solution:

By default, the code move the working directory to the bundle/Contents/Resources/ where we have the client_default.cfg data/ directory and so on.

So all the code is working without any modification.

Be sure the client.cfg RootConfigFile contains "client_default.cfg" without any path.

And voila, no need of special case like "preDataPath = getAppBundlePath() + "/Contents/Resources/" + preDataPath;" or "defaultConfigFileName = getAppBundlePath() + "/Contents/Resources/" + defaultConfigFileName;"

#8 - 11/09/2010 10:58 am - rti

The goal was to split client.cfg, screenshot etc stuff from data and client_default.cfg. This way, every user on the machine has it's own config and the ryzom_client.app bundle does not need to be writable (in case of non-admin users).

#9 - 11/09/2010 10:59 am - rti

So the working directory is set to ~/Application Support/Ryzom atm.

#10 - 11/09/2010 12:06 pm - vl

It's not a problem.

You can set the working dir to bundle/Contents/Resources/ and write client.cfg, screenshot and other user files in your ~/Application Support/Ryzom atm.

#11 - 11/09/2010 12:36 pm - sfb

Just a quick question out of curiosity... CPack generates .app, .framework, etc. Are you considering letting it build the .app for you? I understand the config file work has to happen outside of it. But it would be nice if we could harness that.

#12 - 11/09/2010 12:47 pm - rti

@ace: How could that be done? As I understood the code, lots of stuff is created in / written to working directory. E.g.

init.cpp:742

```
// create the save dir.  
if (!CFile::isExists("save")) CFile::createDirectory("save");
```

```
// create the user dir.  
if (!CFile::isExists("user")) CFile::createDirectory("user");
```

client_cfg.cpp:251

```
const string ConfigFileName = "client.cfg";
```

client_cfg.cpp:1892

```
void CClientConfig::init(const string &configFileName)  
{  
    // if the users client config does not exist  
    if(!CFile::fileExists(configFileName))  
    {  
        // create the basic .cfg  
        FILE *fp = fopen(configFileName.c_str(), "w");
```

All those parts would need adjustments then, don't they?

@sfb: Even ADD_EXECUTABLE can create .app bundles. To be honest, I never looked into CPack. Should have a look into it to see what is possible using that.

#13 - 11/09/2010 03:52 pm - vl

Indeed but the rework is just to append a userDirectory() function that get the ~/.ryzom on linux ~/Application Support/Ryzom on mac.

#14 - 11/09/2010 04:01 pm - rti

Ok, so you'd suggest to have the working path where data is (not necessarily writable), and the writable user path somewhere else (platform dependent), so the code handles the prefixes for this. Hm. Well, yes, thats the other way around and makes the custom client_default.cfg unnecessary... Might be a cleaner solution...

#15 - 11/09/2010 04:09 pm - kerval

I already did that for all debug/log/temporary stuff, it's using getLogDirectory(), which is the one passed to createDebug(logpath); ;)

#16 - 11/09/2010 04:49 pm - rti

ok, finally we agreed on:

- working path is writable
- working path is per user and platform specific (e.g. ~/.ryzom or ~/Application Support/Ryzom)

- data and client_default.cfg may be read only
- on mac, they are in ryzom_client.app/Contents/Resources/
- client.cfg's path to client_default.cfg must only be "client_default.cfg", even it is not in the working dir but in the app bundle
- client_default.cfg's data path must be only "data" if ryzom_client.app/Contents/Resources/data is meant

#17 - 11/09/2010 04:51 pm - vl

Perfect!

#18 - 11/14/2010 11:31 pm - rti

I am thinking about moving this functionality to NLMISC::CPath, because if we want to make CConfigFile able to load RootConfigFilename (even if it is just the plain file name) from the app bundle (or in general, the "(write only) resource path"), then NLMISC needs to know about app bundles. So it is not enough to have the app_bundle_utils in ryzom client anymore.

NLMISC::CPath already provides a similar functionality; it is possible to get:

- a temp path using NLMISC::CPath::getTemporaryDirectory()
- the user specific writable path using NLMISC::CPath::getApplicationDirectory() (the name is badly chosen imho)
- the windows system path using NLMISC::CPath::getWindowsDirectory() (which is not useful in this context)

So I thought about adding the possibility to get:

- the resource path using NLMISC::CPath::getResourceDirectory() which is
 - SOME_NAME.app/Contents/Resources/ on Mac OS X
 - the path the executable is in on Windows, (e.g. C:/Programs Files/Ryzom/)
 - RYZOM_SHARE_PREFIX on e.g. linux (e.g. /usr/local/share/ryzom/)

And I thought about renaming getApplicationDirectory() to something like getUserDirectory() since getApplicationDirectory() sounds like it is the path containing the executable in my opinion.

Another thing is getLogDirectory() which is defined in nel/misc/debug.h. This functionality could be provided by CPath as well to keep the structure.

I would really like to see the path handling work the same way on all platforms. It does not feel right to just add another #ifdef NL_OS_MAC in the client to make it run. If CPath would return the paths to all necessary directories, this could be achieved quite easily.

And, if this would be implemented across all platforms equally, the windows implementation would gain per user configurations. Isn't that nice? :) But: There should be some way then to automatically migrate the current system wide configuration to the per user directories though.

What do you guys think?

#19 - 01/08/2011 06:48 pm - rti

- Target version changed from Version 0.8.0 to Version 0.9.0

#20 - 02/05/2011 03:52 pm - Jeru

- File atys.icns added

For those who wants, I made a low resolution icon for Mac by converting the ryzom.png file. Just copy it in the .../ryzom_client/Contents/Resources folder.

If someone knows hwo to edit the make install procedure to copy automatically the file, it should be great.

PS : If you have a higher resolution of the ryzom.png file, i'll make a better icon. Thx

#21 - 02/06/2011 09:37 am - vl

- File *atys.png* added

Jeru you can find the big version (1000x1000 pixels) of the ryzom logo on the attachement.

#22 - 02/06/2011 11:13 am - Jeru

- File *atys.icns* added

Here's the high res icon file for mac :) (From 512*512 to 8*8)

Files

atys.icns	12.7 kB	02/05/2011	Jeru
atys.png	1.6 MB	02/06/2011	vl
atys.icns	627.8 kB	02/06/2011	Jeru