

Morpho - Bug #202

new config management/profile feature for morpho framework

04/09/2001 02:39 PM - Matt Jones

Status:	Resolved	Start date:	04/09/2001
Priority:	Normal	Due date:	
Assignee:	Dan Higgins	% Done:	0%
Category:	morpho - general	Estimated time:	0.00 hour
Target version:	1.1	Spent time:	0.00 hour
Bugzilla-Id:	202		
Description			
Need to add the ability to have multiple user profiles on morpho that can differentiate the files owned by various users, and their preferences.			
Related issues:			
Blocked by Morpho - Bug #232: [RFE] saved queries		Resolved	05/08/2001

History

#1 - 04/09/2001 02:45 PM - Matt Jones

This includes the ability to allow all plugins to store their preferences/configuration in a common config file managed by the top-level framework.

#2 - 04/09/2001 02:48 PM - Matt Jones

Also includes finishing the XMLConfig format for storing all preferences.

#3 - 04/09/2001 02:50 PM - Matt Jones

Also includes building a graphical options editor to change options in the config file. Note that the configuration module should be a standalone class that we can reuse in other client projects and have a simple interface for getting and setting preferences, including repeating preferences and hierarchical preferences.

#4 - 06/06/2001 01:27 PM - Matt Jones

Reassigned to Jones because this feature is needed for implementing saved queries.

#5 - 06/10/2001 08:25 PM - Matt Jones

Further work on the new profiles feature of morpho. Now, upon morpho startup, we check to see if there was a profile in effect from a previous session, and if so, we load that profile and ask the user to log in. If not, we prompt the user to create a new profile, which initializes the profiles directory and creates a default profile. The "profiles/username" directory contains the profile file (username.xml) and the "data", "cache", and "temp" directories for that user. Any user-specific data should now be stored in their profile directory.

Please note that only global configuration information is now stored in "lib/config.xml" (although I haven't excised it all yet), and that user specific configuration information is now stored in "profiles/username/username.xml". The easy (and correct) way to programmatically access this information is to get a handle to the current profile object from the framework and use it, like this:

```
ConfigXML profile = framework.getProfile();
String datadir = profile.get("datadir", 0);
```

I have modified DataStore to use these new profile directories, but have not yet modified LocalQuery to reflect this change. Will do so tomorrow. Also found several places in the code

where the "lib/config.xml" is hardcoded and should not be, so I will fix this as well (by getting the config object from the framework properly).

TODO

- 1) Still need to copy sample data to the profile dir when creating the profile.
- 2) Possibly create a metacat account if it doesn't exist
- 3) Change behaviour to deal properly with the LDAP baseDN
- 4) Change behaviour to properly deal with unique scope for accession #
- 5) Decide what info we need for the profile and finish collecting and storing it in the profile (e.g., contact info)

#6 - 06/10/2001 08:28 PM - Matt Jones

Bug on linux: new profile feature on Morpho startup always results in Morpho exiting because, for some reason, it thinks the user has chosen not to create a profile, even though the profile is created properly. Thus, it is likely that somehow ClientFramework.profile is not getting set properly, and so ClientFramework.getProfile() is returning null, and causing the error.

#7 - 06/13/2001 01:31 AM - Matt Jones

Fixed the linux bug (order of operations problem).
Fixed LocalQuery, so it now queries the data in the users profile.
Removed hardcoded refs to the config.xml file (except in a couple main() methods for testing).
Copied sample data to profiles/username/data when a new profile is created.

TODO

- 2) Possibly create a metacat account if it doesn't exist
- 3) Change behaviour to deal properly with the LDAP baseDN
- 4) Change behaviour to properly deal with unique scope for accession # (in other words, if the username+baseDN is globally unique, but we use username as a scope, we're gonna have a bunch of accession# conflicts).
- 5) Decide what info we need for the profile and finish collecting and storing it in the profile (e.g., contact info)

#8 - 10/16/2001 07:55 PM - Matt Jones

New bugs created for several items. Onky feature here that remains TODO is:

- 5) Decide what info we need for the profile and finish collecting and storing it in the profile (e.g., contact info)

#9 - 11/06/2002 01:08 PM - Matt Jones

DONE. Final task (5) regarding complete profile information added in new bug 669 where responsible party information is dealt with more comprehensively.

#10 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 202