Smuxi - Feature # 375: Better remote backend reconnect handling

Status:	Closed	**Priority:**	Normal
Author:	Michael Elsdörfer	**Category:**	Frontend GNOME
Created:	05/07/2010	**Assigned to:**	Mirco Bauer
Updated:	12/21/2011	**Due date:**	
Complexity:	Medium		
Subject:	Better remote backend reconnect handling		
Description:	I'm using Smuxi with a remote backend. If I loose connection to the server, which is at least one a day, but frequently happens more often than that of course, Smuxi will show a dialog in the current virtual workspace, asking me if I would like to reconnect. If I respond yes, it'll move the main window to the current workspace, and I need to move it back. I would argue that the dialog is wholly unnecessary - Upon losing the connection, Smuxi should automatically try to reconnect - the user doesn't even need to know that the connection was gone. If reconnection fails, further attempts could be delayed, with the delay slowly increasing etc. The interface would reflect the disconnected state, and all the user to request a manual reconnect at any time.		

Associated revisions

08/28/2011 10:01 AM - Mirco Bauer

[Frontend-GNOME] Auto-reconnect to smuxi-server if frontend has no focus (closes: #375)

08/28/2011 10:08 AM - Mirco Bauer

[Frontend-GNOME] Auto-reconnect to smuxi-server if frontend has no focus (closes: #375)

08/28/2011 11:22 AM - Mirco Bauer

[Frontend/Frontend-GNOME] Implemented background and concurrent chat sync (references: #375)

Instead of adding and syncing each chat at a time in the GUI thread and thus
block the GUI from refreshing, we are now using a thread pool with 4 worker
threads which add and sync the chats in the background and only populating the
retreived data is done in the GUI thread. This makes the frontend connect
faster, feels much more responsive and even allows to use the synced chats
already while it is still syncing the remaining ones.

Implemented the ChatViewSyncManager class which queues and processes the
AddChat and SyncChat calls using the ThreadPoolQueue.

Added Sync() to IChatView which the ChatViewSyncManager will call from a
thread pool worker thread and thus must be implemented thread-safely.

Added Populate() to IChatView which the ChatViewManager will call from it's
ChatViewSyncManager.ChatSynced handler to populate the data into the GUI.

Added ID and Position property to IChatView which the ChatViewManager will
call to populate the data before ChatViewSyncManager.ChatAdded will be raised
from a thread pool worker thread and thus must be implemented thread-safely.

Summary of the sync API workflow:

[Engine]

- FrontendManager.Sync() calls IFrontendUI.AddChat() and SyncChat() via
  .NET remoting.

[Frontend application]

- IFrontendUI.AddChat() calls ChatViewSyncManager.QueueAdd().
- IFrontendUI.SyncChat() calls ChatViewSyncManager.QueueSync().

[Frontend library]

- ChatViewSyncManager queues the Add and Sync requests to ThreadPoolQueue.
- ChatViewSyncManager.QueueAdd() create a sync lock to make sure a sync
  processor will not run before the chat was added to the frontend.
- ChatViewSyncManager's add worker will fetch ID, ChatType, Position and
  ProtocolManagerType from a thread pool worker thread and raises the
  ChatAdded event.

[Frontend application]

- ChatViewManager's ChatViewSyncManager.ChatAdded handler adds the chat to
  the GUI and calls ChatViewSyncManager.ReleaseSync() to signal the possibly
  waiting thread pool worker thread.

[Frontend library]

- ChatViewSyncManager's sync worker optionally honors a sync lock and then
  calls IChatView.Sync() from a thread pool worker thread and raises the
  ChatSynced event.

[Frontend application]

- IChatView.Sync() fetches all the data it needs from the ChatModel via
  .NET remoting.
- ChatViewManager's ChatViewSyncManager.ChatSynced handler calls
  IChatView.Populate() from the GUI thread.
- IChatView.Populate() pushes the fetched data into the GUI.

History

07/19/2010 05:27 PM - Mirco Bauer

- Category set to Frontend GNOME

- Complexity set to Medium

The problem with auto-reconnecting to the smuxi-server could make the user think the application is hanging. Around Smuxi 0.8 I plan to introduce a progress dialog when connecting to a smuxi-server which could solve this issue with auto-reconnecting. Does this sound acceptable for you? Auto-reconnect while showing that it lost the connection and is reconnecting? One other issue is that the reconnect might fail, say your network or internet is down, in that case the connect dialog could show the failed attempt though.

07/20/2010 09:51 PM - Michael Elsdörfer

Mirco Bauer wrote:

> The problem with auto-reconnecting to the smuxi-server could make the user think the application is hanging.

No, the main Smuxi Window would need to make the user aware of what is going on - by graying out the interface, or maybe showing a big "I am reconnecting right now"-overlay on top of everything. I just think that from a UI perspective, this is vastly superior to any dialog:

- As long as Smuxi is in the background, behind some other window, or on a different workspace, I don't see why the user would care about the connection being lost. He shouldn't be bothered. He is not interacting with Smuxi right now. There's nothing he gains from it; at best, he can press the "Yes, reconnect" button, something that could happen automatically.

- As soon as the user decides he wants to interact with Smuxi, the window will clearly indicate the current connection status and the reconnection attempt. The user will know about the connection loss right when he needs to, but not earlier.

> One other issue is that the reconnect might fail, say your network or internet is down, in that case the connect dialog could show the failed attempt though.

Again, I don't think it should. Not intrusively. Show it within the Smuxi window, of course. If you judge it important, I suggest that a notification could be displayed, or an indicator. When my connection fails, I don't want to be bombarded by all kinds of applications about their connection troubles. I think Smuxi should behave like most instant messenger apps do: If you lose the connection, try to reconnect as soon as possible, using a delayed reconnect loop, i.e. wait a little bit longer after each failed reconnect attempt. In the meantime, change the indicator icon to something that says "offline".

Also, with respect to the hanging, I notice that this happens from time to time in the UI when Smuxi is communicating with the server. For example, when I say "Yes" in the reconnection dialog, the UI is frozen for a couple of seconds until the reconnect is complete. It seems like the network communication is happening in the UI thread. A better architecture would keep the UI responsive at all times, and simply show a spinning "i am busy" icon.

08/24/2011 10:52 PM - Mirco Bauer

Auto-reconnect to smuxi-server when Smuxi is not focused is a very good idea. The user wouldn't notice anyhow and would not be faced with a reconnect nag dialog if he wants to chat... Of course when the reconnect is happening the user should be able to see that, else he would believe Smuxi is suddenly not working correctly.

08/28/2011 10:06 AM - Mirco Bauer

- Assigned to set to Mirco Bauer

- Target version set to 0.8.9

08/28/2011 10:06 AM - Mirco Bauer

- Status changed from New to Closed

- % Done changed from 0 to 100

Applied in changeset commit:"75c554063336e77efbf3e30440a5787ba2826154".

12/21/2011 11:31 AM - Mirco Bauer

This change can be tested with a development build which can be obtained from here: http://www.smuxi.org/documentation/running-from-git/