QFJ-Specific JMX Management Console

> May latest thinking on this is to initially provide tutorials on how to use JManage and MC4J consoles rather than creating a special-purpose console.

Unknown macro: {scrollbar}

The UI below is a rough first cut at a console. However, it does have some real functionality even now. It is connecting to the QFJServer MBean server and retrieving session information. I added sessions for each FIX version to have a few more rows. The UI is using the Java Desktop Network Components(JDNC) library so the table is sortable and filters/highlighting can easily be added. The toolbar can be undocked. All columns are real data from the MBeanServer.

Could not generate thumbnail: Image file format not supported

(click for larger image)

I'd like to have a QFJ-specific GUI for managing sessions, their supporting services and application objects. One starting point would be to define the desired functionality for this console, possibly in terms of use cases.

Some potential features:

- Display of basic session information
  - Connection state
  - Messaging statistics
- Session control
  - Logoff, disconnect, logon, and reset sessions.
  - Modify sequence numbers
- Automated monitoring and alerting (could be separate feature)
- Application monitoring
  - Statistics
  - Status and control

The application status monitoring would require a standard data format for reporting application status. Each application will be different and the JMX console should not be dependent on application-specific classes. Using an XML description like JBoss's XMBean (MX4J has something similar) descriptions would allow us to use dynamic mbeans to do simple monitoring and control applications.

For monitoring I'm thinking of just a table and charting widget that would work with any standard MBean. One challenge is how to provide decent user interfaces for application control that requires nontrivial user input. While I'm brainstorming, a possibility would be for the application to provide an XML definition of a simple Swing UI for entering nontrivial commands. The definition would tell how to map the collection information to JMX operations. This approach would be limited to standard Swing widgets to avoid any remote classloading issues.