Team Members:

Monica Sangam

Matthew Belanger

Overall Goals:

1. We want to create a Knime node that reads any qvx file and makes the data available in Knime. The qvx files will be located on a local QlikView server.
2. We want to create a Knime node that takes tabular data (in Knime) as input and converts it into qvx format. The qvx file will be saved on a local QlikView server.

Stage 1 - Reading and writing qvx files:

1. QvxReader Java class:
   1. Input: qvx file
   2. Output: 2D Object Array (Object[][])
   3. **We have a working prototype for this class**. There are some things that need to be improved. Most importantly:
      1. qvx file currently is assumed to only have unsigned integer fields
      2. Documentation/comments
      3. Deal with large qvx files
2. QvxWriter Java class:
   1. Input: Either csv or 2D String Array (String[][])
   2. Output: qvx file
   3. **We have a working prototype for this class.** There are some things that need to be improved (the sames things that need improvement in the QvxReader class):
      1. qvx file currently is assumed to only have unsigned integer fields
      2. Documentation/comments
      3. Deal with large datasets

Project Structure:

We used Jaxb in order to bind Java Classes to an XML representation, which made it fairly easy to deal with the XML files. We created an XML file based on the QVX documentation, and Jaxb automatically created several classes based on this file. The most important classes that we wrote are QvxReader and QvxWriter (located in “edu.njit.knime.adapter.qvx” package). Our code can be tested using the Java files found in the “edu.njit.test” package.

To set up in Eclipse: Download “eclipse-workspace.zip”, open project from file system and select eclipse-workspace/eclipse-workspace/QvxAdapter.