CUD (Chinook User Document)
Running Chinook
Author: Stephen Montgomery Date: May 21st, 2004

This document supercedes the CUD entitled "How-to Create a Chinook Development Environment in Jbuilder 8.0". It describes the methods involved in running a Chinook P2P Node, Client, or Server using the new node architecture.

Background: The node architecture was designed to make Chinook true peer-to-peer. (One network entity for a user regardless of how many instances of servers or clients they are running on their computer.)

Clients and servers connect to the P2PNode using TCP sockets to ask for the list of discovered/valid services or to recommend services for advertisement over the network. By removing these function we are able to create a large number of clients and servers for various platforms without concern for the underlying P2P engine. This does require though that a P2PNode daemon is started before a user runs a client or server that requires the P2P functionality.

Description:
All the main class have been moved to packages called exec.

Running the P2P Node:

The main class for the P2P node is
ca.bcgsc.chinook.p2p.exec.ChinookP2PNode
all you need to do is run it.

Note: Ensure all the dependency jars in the chinook/lib/ folder are added to the CLASSPATH before you start any chinook code. Furthermore, it is **required** that the chinook/resources/ folder is added to the CLASSPATH.

Running the Client Node:

The main class for the Chinook gui'd client is
ca.bcgsc.chinook.client.exec.ChinookClient
all you need to du is run it.

Note: Ensure all the dependency jars in the chinook/lib/ folder are added to the CLASSPATH before you start any chinook code. Furthermore, it is **required** that the chinook/resources/ folder is added to the CLASSPATH.

Running the Server Node:

The main class for the Chinook server is
ca.bcgsc.chinook.server.exec.ChinookServer
there are a few parameters to customize for proper execution. This basically

is setting the RMI security manager and specifying the right codebase. This can be done by adding to your VM parameters (note that the codebase parameters are separated by a space):

-Djava.rmi.server.codebase="file:///home/smontgom/jbproject/chinook/classes/ file:///home/smontgom/jbproject/chinook/lib/filewire.jar"

-Djava.security.policy=/home/smontgom/jbproject/chinook/resources/chinookRMI.policy

Customize these to your own chinook installation!

The default mode from CVS is rmi, remember to start the rmiregistry before starting a server.

Cheers!