

ChoPS

Quick Start Guide

Sean Lawlor
McGill University
Centre for Intelligent Machines

Patrick Diez
McGill University
Centre for Intelligent Machines

April 21, 2011

Contents

1	System Requirements	2
2	Server	2
2.1	Server Startup	2
2.2	Starting the server	3
2.3	Server Administration	3
3	Remote Administrative Frontend	4
3.1	Starting an Administrative Server Connection	4
3.2	Preparing the server	7
3.3	Starting a Project	7
3.4	Connecting to an existing Project as a client	12
4	Notes	15

Overview

This document outlines the basic commands and a brief walkthrough of how to utilize ChoPS for distributed data transmission.

NOTE: This document is a work in progress, and will be updated as new features and modifications are added/made.

1 System Requirements

The ChoPS system is built to be very lightweight and portable, so there is no need to complicated setups however there are some basic system requirements which are broken up by operating system.

- For all systems
 - At least a 500 MHz processor has been tested safely, however due to the fact that ChoPS takes very little resources this number may be reduced after some testing.
 - Preferably at least 512 MB of RAM, tested minimum: 128 MB
- Linux (x64 or x86)
 - Either the Sun Microsystems (Oracle) or OpenJDK v1.6 Java Runtime Environment. (If your system is updated this should not be a concern)
- Mac OSX
 - There has been a notification of a problem with Mac OSX, which is that the default JRE which the Mac System Updater gives is not of a new enough version. (It is version 1.5 and ChoPS requires v1.6 or greater) Therefore for Mac it is necessary to download the entire Java Development Kit available from Oracle. (Available from: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Windows XP/Vista/7
 - For Windows you simply need the most recent JRE installed on your system to run ChoPS.
- Android/iOS
 - Due to lack of testing in mobile applications, we are unable to provide validated support towards the Android/iOS platform at this time.

2 Server

The server is implemented in a simple Jarfile which contains the server, along with various modes and the GUI client which will be outlined in this section as well as Section 3. ChoPS can be started in the following manner:

2.1 Server Startup

```
cd <path to ChoPS.jar and resources folder>
java -jar ChoPS.jar
```

The following output is produced after running the previous command:

```
Usage for ChoPS is:
java -jar ChoPS.jar (System Type) <options>

Valid System Types are:
No System Type: Reprint this help menu
server <--loglevel=#> : Starts the ChoPS server ,
    with default log level unless specified
gui : Loads administrative gui frontend
hashtest <texttohash>: create a hash of specified test
debug <--loglevel=#>: start a debug instance of the server
    (with or without a specified loglevel)
```

```
LogLevels: (If no level specified, will default)
0: Critical
1: Error
2: Warning (Default value)
3: Info
4: Debug
```

```
The following options are for camserv testing only...
camserv-server host port serverPort - start the fake http server from host:port to ↔
serverPort
camserv-client host port serverPort - start the fake http client from host:port to ↔
serverPort
camserv-edges host port serverPort - start the edge-detector from chord ring output from ↔
host:port to serverPort
```

```
Written by: Sean Lawlor and Patrick Diez
Date Created: April 3, 2011
```

For the server, we now have two options on how to start the ChoPS daemon. The first option is in standard “server” mode, which forces the server to put all log output to a flat file located in *resources/log/chops.log*. The other option is to log to the standard out stream, which is the basic setup will simply put all logged output to the screen.

There are then a series of options for the loglevel to run, which will specify how much output to produce from the program. The default level is to log Warnings, Errors, and Critical Messages. However you are free to modify this with the *-loglevel=LEVEL* argument to either server or debug modes.

2.2 Starting the server

To start the server in “debug” mode:

```
java -jar ChoPS.jar debug
```

Or to start the server in “debug” mode with a specified, non-default loglevel

```
java -jar ChoPS.jar debug --loglevel=4
```

To start the server in standard “server” mode:

```
java -jar ChoPS.jar server
```

To start the server in standard “server” mode with a specified loglevel:

```
java -jar ChoPS.jar server --loglevel=4
```

2.3 Server Administration

The server will by default open port 10002 for the administrative connections necessary to administer itself, however this is modifiable by any user by simply either logging via the admin GUI or manually editing the configuration file located in *resources/config/chops_config.xml* and changing the value under the `admin_port` XML tag. However you cannot modify this file when the system is running, as updates will no be taken into account until an “update” command is sent to the server via the administration port. The server will open a standard TCP socket on that specified administrative port, and listen for incoming connections. It will then take plaintext commands over the socket and process them in order which will be all the commands needed for updating the system and making Chord rings activate/deactivate. You can administer the server via a tool such as *telnet* however the list of commands will be needed prior to attempting the connection.

The list of commands are outlined quickly in the following:

```
login <serveradmin, projectadmin, project> <username> <password> : login via the specified role
get <configuration, projects> : return the current configuration file, or the list of running ↔
  projects
set configuration <serialized configuration> : set a new configuration file
update : apply the new configuration file
<close, quit, exit> : close the connection
```

3 Remote Administrative Frontend

Finally the last piece of ChoPS is the administrative frontend graphical user interface. This GUI has been designed to try and minimize knowledge of how ChoPS works and to simply allow people to start/stop ChoPS projects as necessary without overhead.

The GUI can be started with the following command:

```
java -jar ChoPS.jar gui
```

When the system is started, the following splash screen is seen:

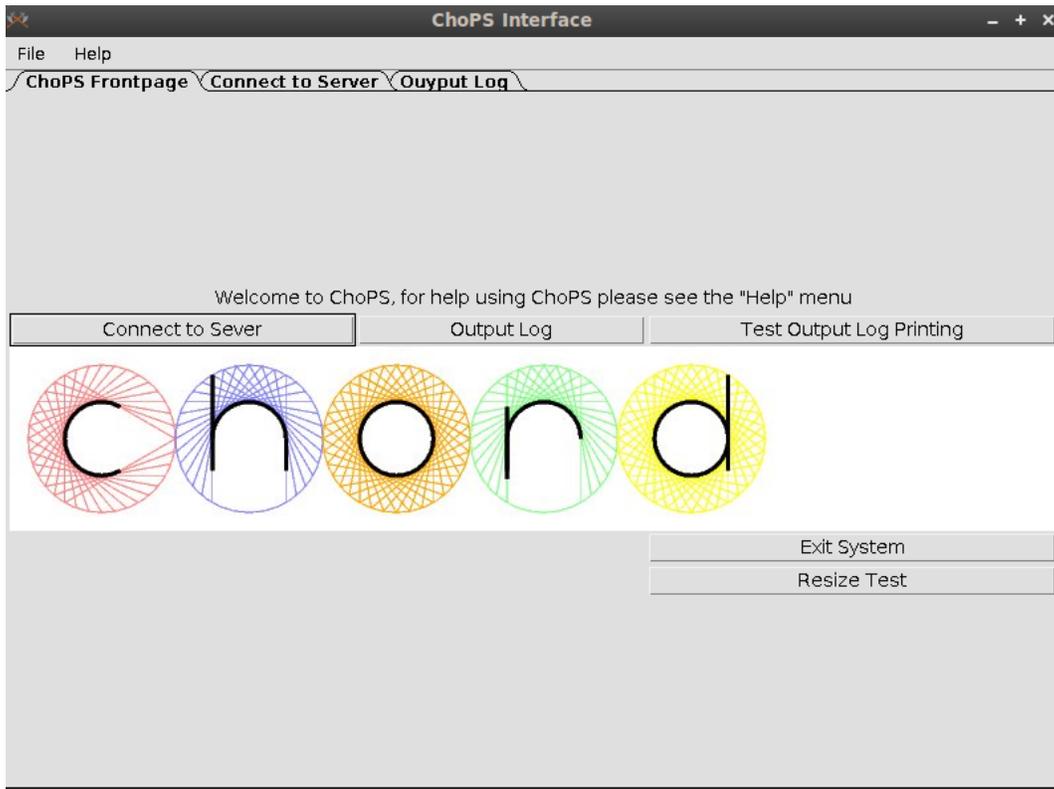


Figure 1: ChoPS Splash Screen

3.1 Starting an Administrative Server Connection

To connect to a server with administrative privileges, a connection to the server over the administrative port with the correct login is required. To do this with the GUI the following steps apply:

First from the frontpage/splash screen click on the *Connect to Server* button or tab.

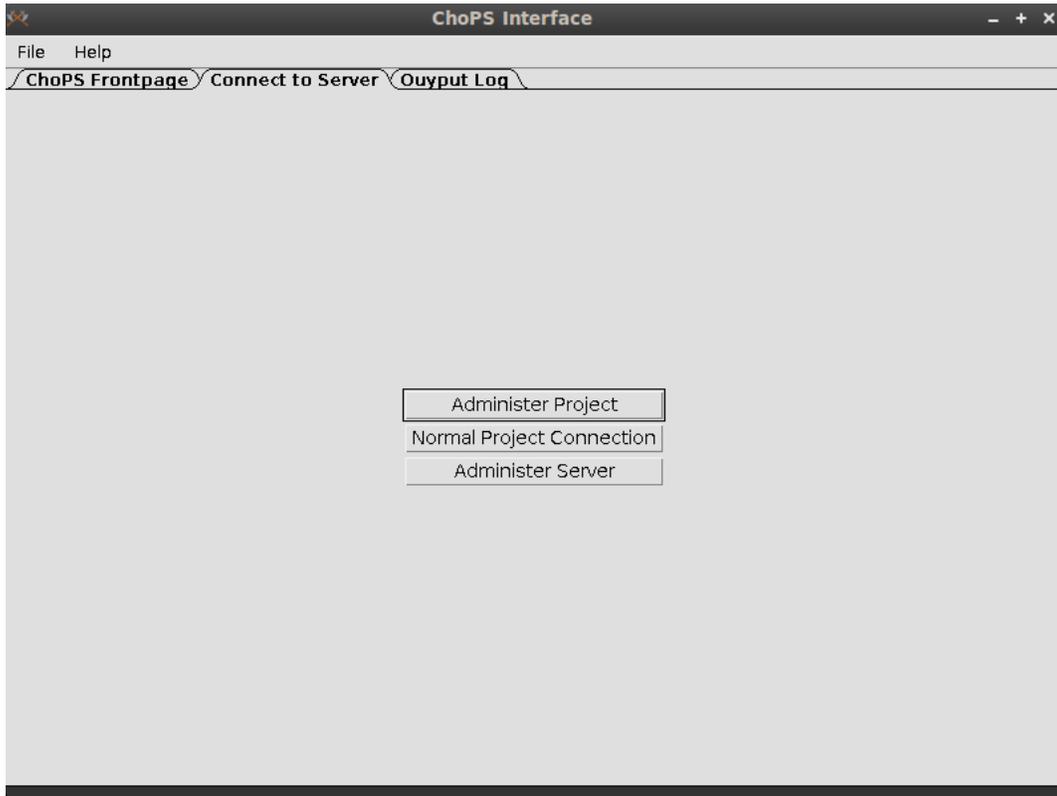


Figure 2: ChoPS Connect to Server Panel

At this time, the only active connection mode is the *Administer Server* phase, the other modes will be added as feature sets to project maintenance. Once one clicks on the *Administer Server* button, a new window will open which asks for the login information to a ChoPS Server.

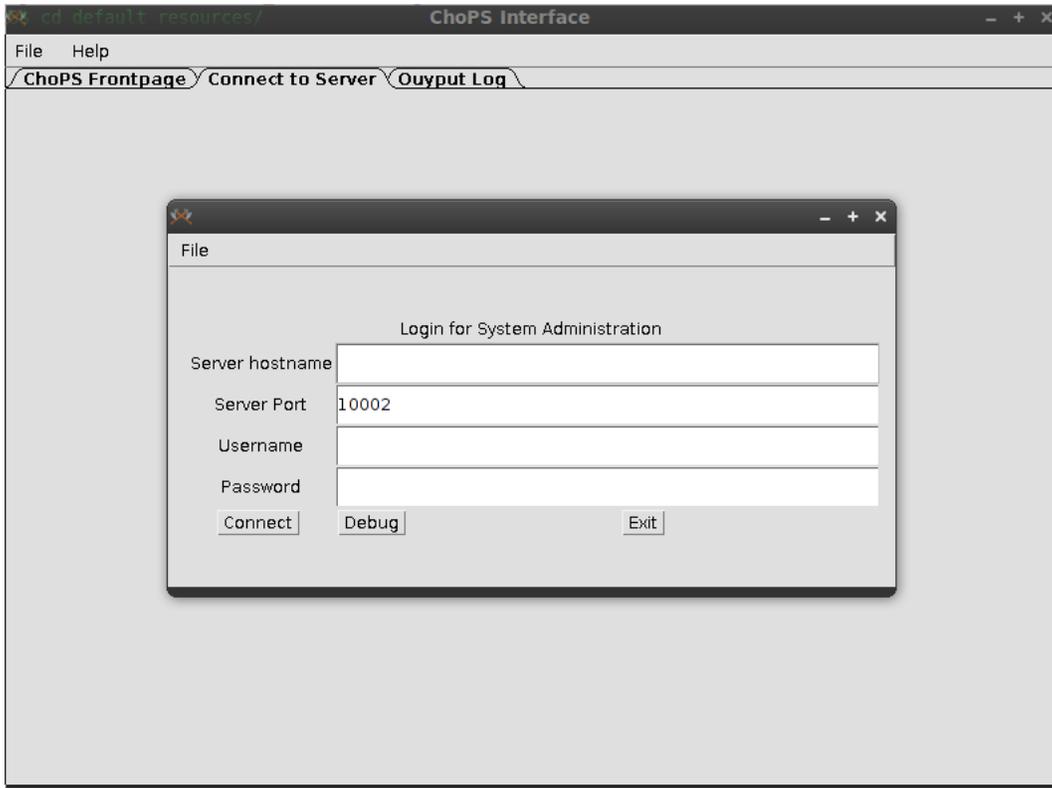


Figure 3: ChoPS Administer Server Login Window

The debug button here only exists for testing, and will preload the following information:
Server Hostname: localhost
Server Port: 10002
Username: root
Password: C0ldSn0w

**This information is the default login/password shipped with all ChoPS instances
For safety the admin password is encrypted with a SHA-512 encryption**

Once the information is entered on this screen, the connection can be started by clicking *Connect*, which will prompt the loading of a new tab on the main window, containing all the administrative features available to that specific server.

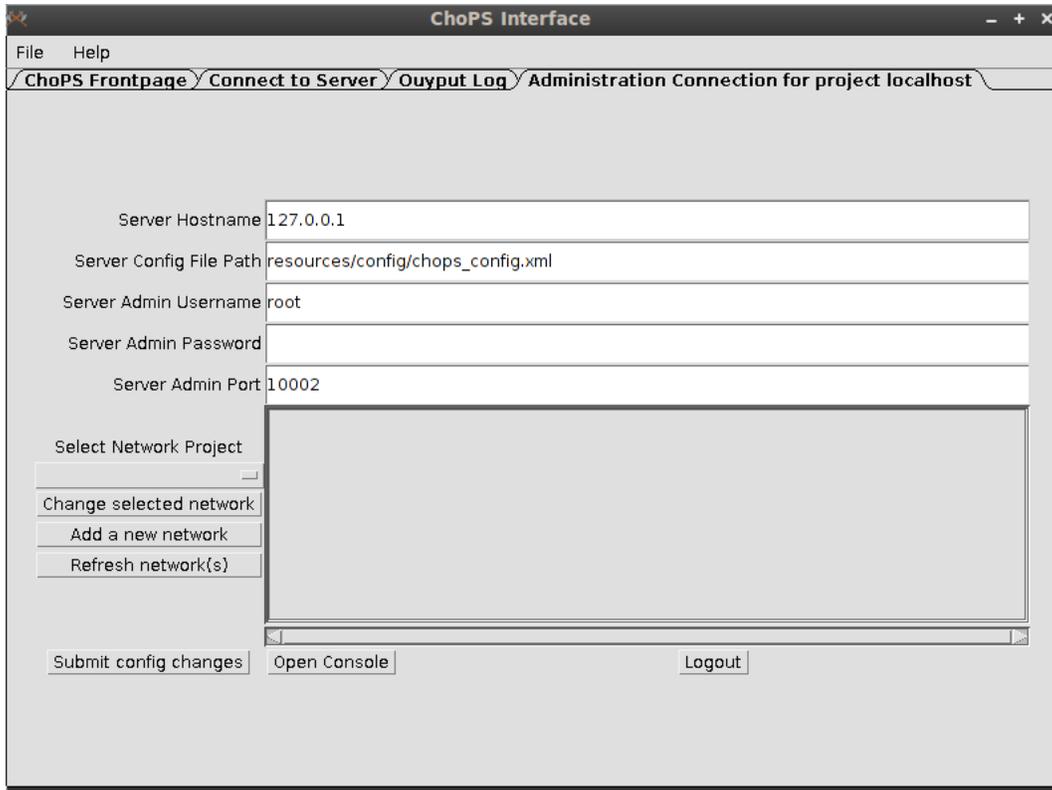


Figure 4: ChoPS Administer Server Tab

As one can see with this instance, there are no running projects because the *Select Network Project* dropdown list is empty.

3.2 Preparing the server

There is truly only one thing that needs to be done in order to prepare the server for starting/participating in ChoPS projects, and that is to set the "Server Hostname" as shown in Figure 4 to that of the *externally visible* IP address of the server. If you only have a ring running within a specified subnet, then the IP address within that subnet is fine. However if there is a single (1) host outside the subnet then all the IP addresses need to be routable.

3.3 Starting a Project

To start a ChoPS Project, there first needs to be a socket available for which the ChoPS server can read data off. For example, there needs to be data produced for which you need ChoPS to distribute to many hosts. This socket needs to be a server socket, since the ChoPS server only connects and then reads off the socket. Also ChoPS will not send ANY data across the data production stream, and therefore data needs to be fed once ChoPS connects to the data. Either that or some type of wrapper will need to be developed which does the initial "GET /" if HTTP and then just forwards all data to another socket which ChoPS can read off.

Finally within the GUI, to start a project click on "Add a new network" and a new window will open:

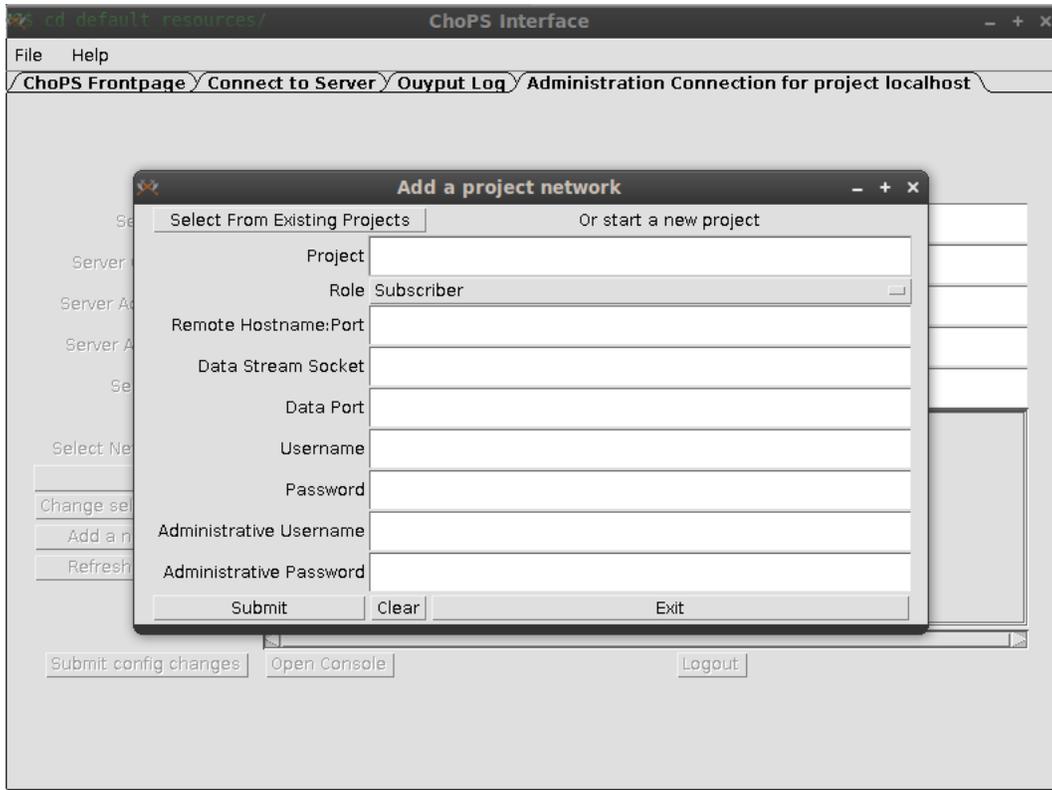


Figure 5: Adding a ChoPS Project/Network

You then need to fill in the following fields, and there is validation of the entered data. You can name the project any name you wish, as long as it is a unique name to that server. You then need to change the *Role* to Publisher through the drop down menu available. If you hover over some of the fields, there are tool-tips available to make sure you enter correct and accurate data. When you change your role, you'll notice that the fields get modified slightly:

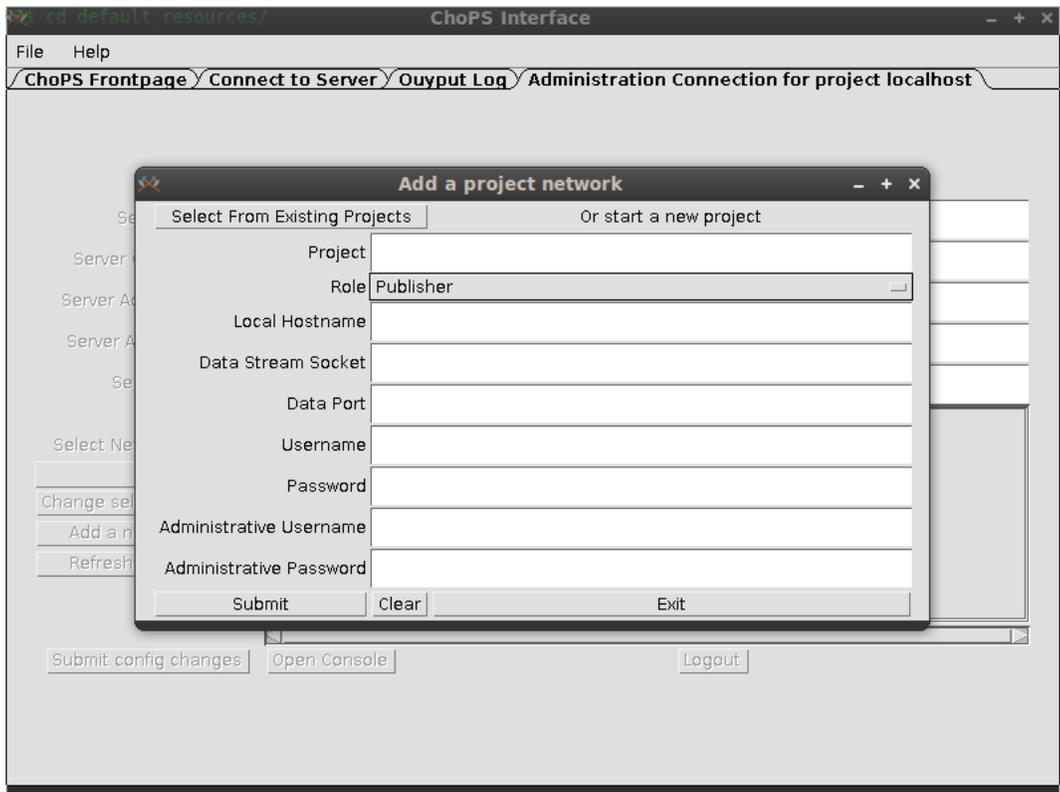


Figure 6: Creating a new ChoPS Project/Network

Now as a publisher, you are only asked to specify the local system hostname, no port along with it, because the Data Port will be added to the hostname as needed. So in Figure 7 you can see that I only specify my externally visible IP address.

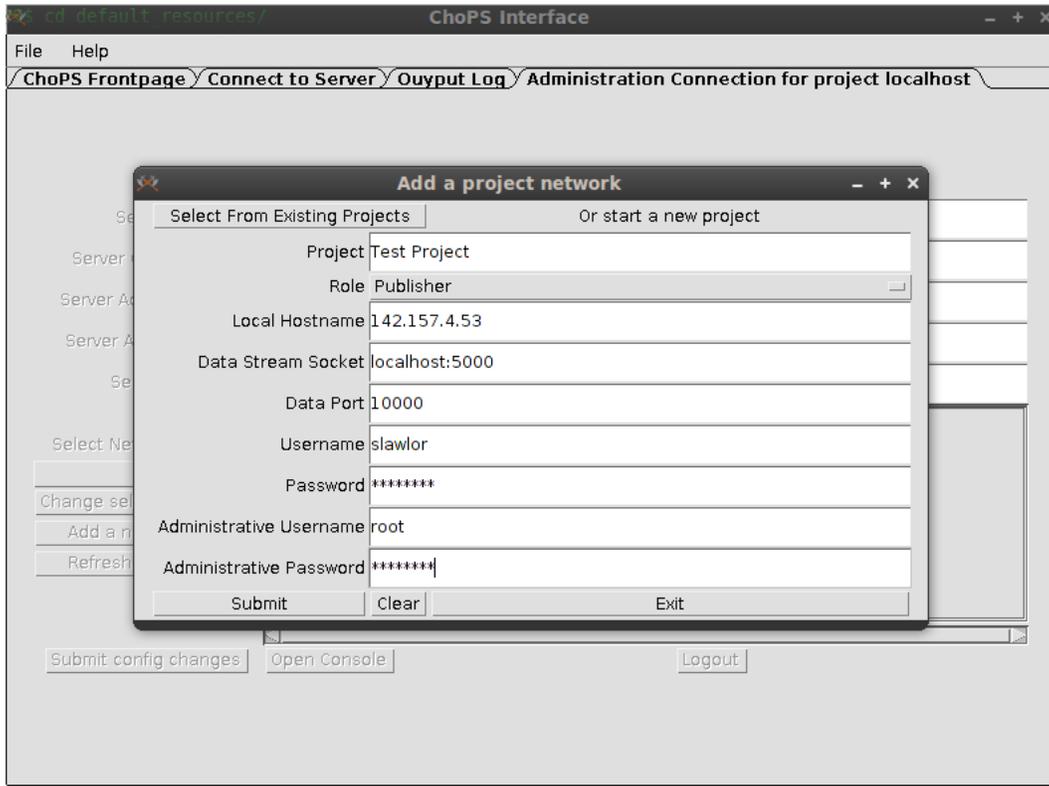


Figure 7: Creating a new ChoPS Project/Network with demonstrative data

Notes About Project Data:

- The Hostname should be externally visible to all hosts within a project
- The Data Stream Socket is the socket which ChoPS will bind to and read data out of, then replicate to all member hosts
- The Data Port is the port for all ChoPS communications between each other and is unique to a project on the specified host, meaning that you can only run a single ChoPS project on a single port.
- The Username/Password/Administrative Username/Administrative Password are, as of yet, unused so you may put whatever you wish in these fields, as long as they are not empty.

We can then click Submit, and we will be returned to the Administrate Server Tab : Figure 4. We finally need to click on “Refresh Network(s)” (This is a small bug which we are trying to fix, through some type of auto-refresh feature.) We should then see the drop down list update, with a single entry.

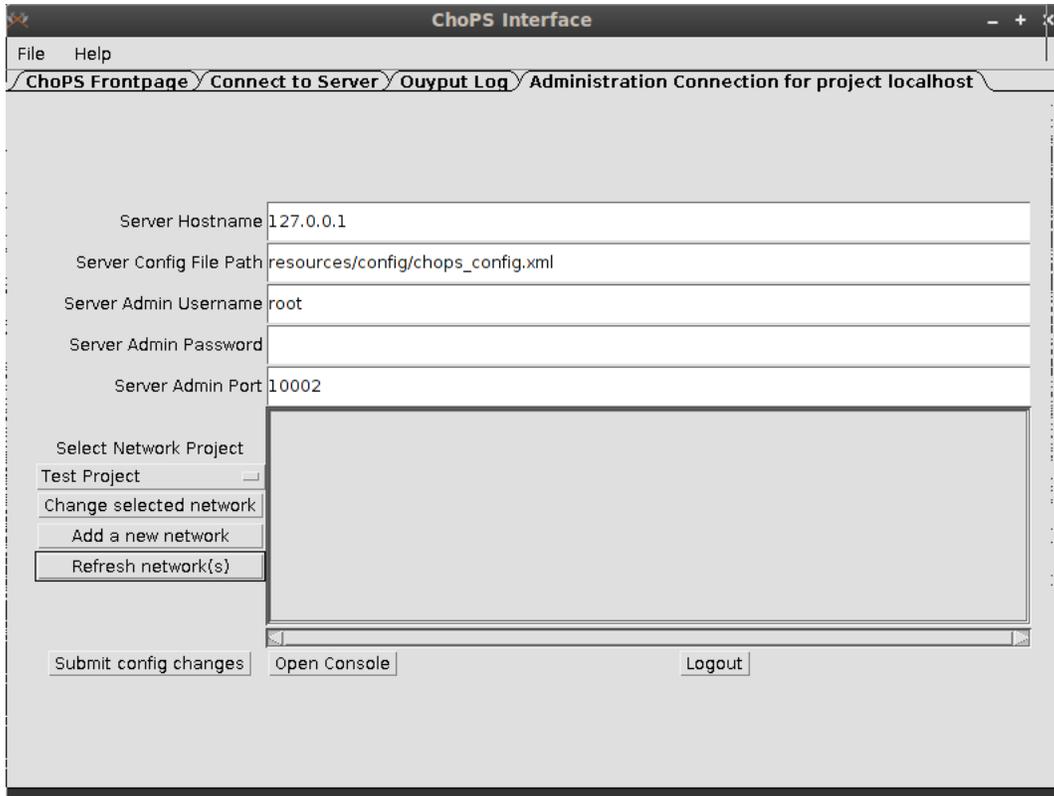


Figure 8: Administer Server Tab with a Single Publisher Network

And if we then click on the drop-down list and select “Test Project” we will see the pane fill with the project’s data which we can then verify for correctness.

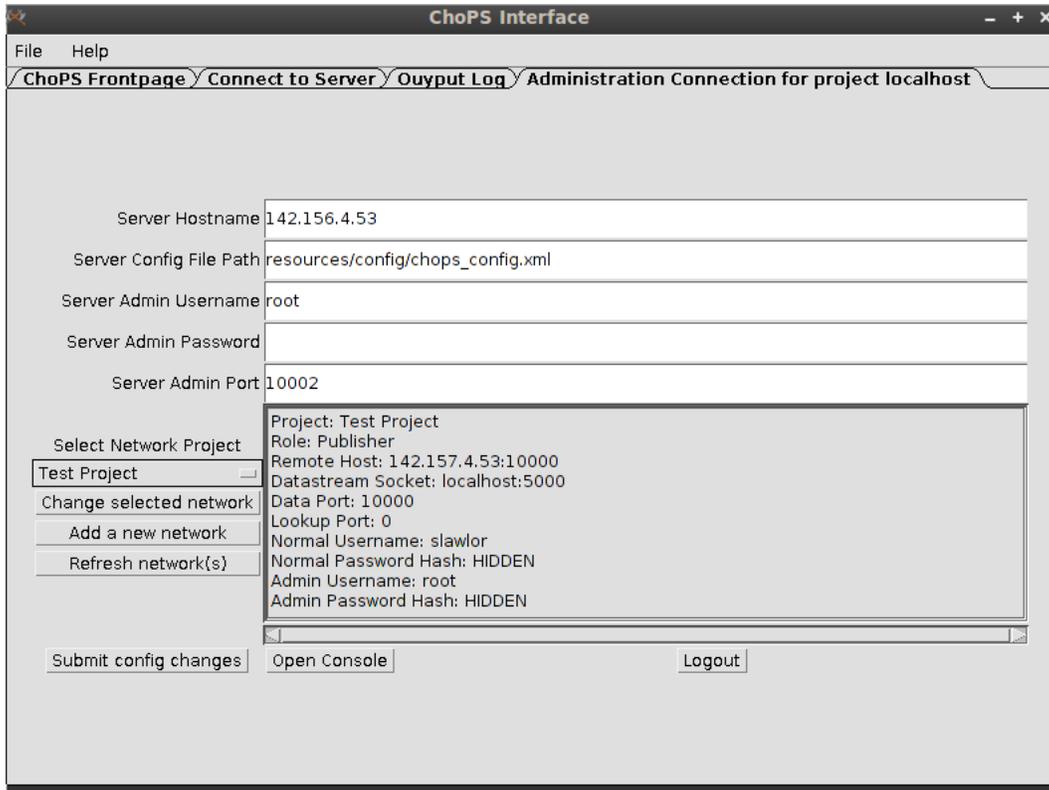


Figure 9: Administer Server Tab with a Single Publisher Network and Displayed Details

Then when the *Submit config changes* button is pressed on the main window, all the modified/added/deleted networks are sent to the server and all the changes activated.

If a field is left blank, and no popup occurs notifying you that the field needs to be filled it means that the previous set value will be used

3.4 Connecting to an existing Project as a client

There are two modes to ChoPS, *Publisher* and *Subscriber*. We previously covered the first, and here we will cover the latter. In order to receive data transmitted through ChoPS, one needs to subscribe to a data stream. To begin this, we first open a connection in administrative mode to the server, and we follow the beginning steps in Section 3.3 until we get to the *Add a Project/Network* window as shown in Figure 5.

We now want to leave the default setting in the drop-down list for the Role to *Subscriber*.

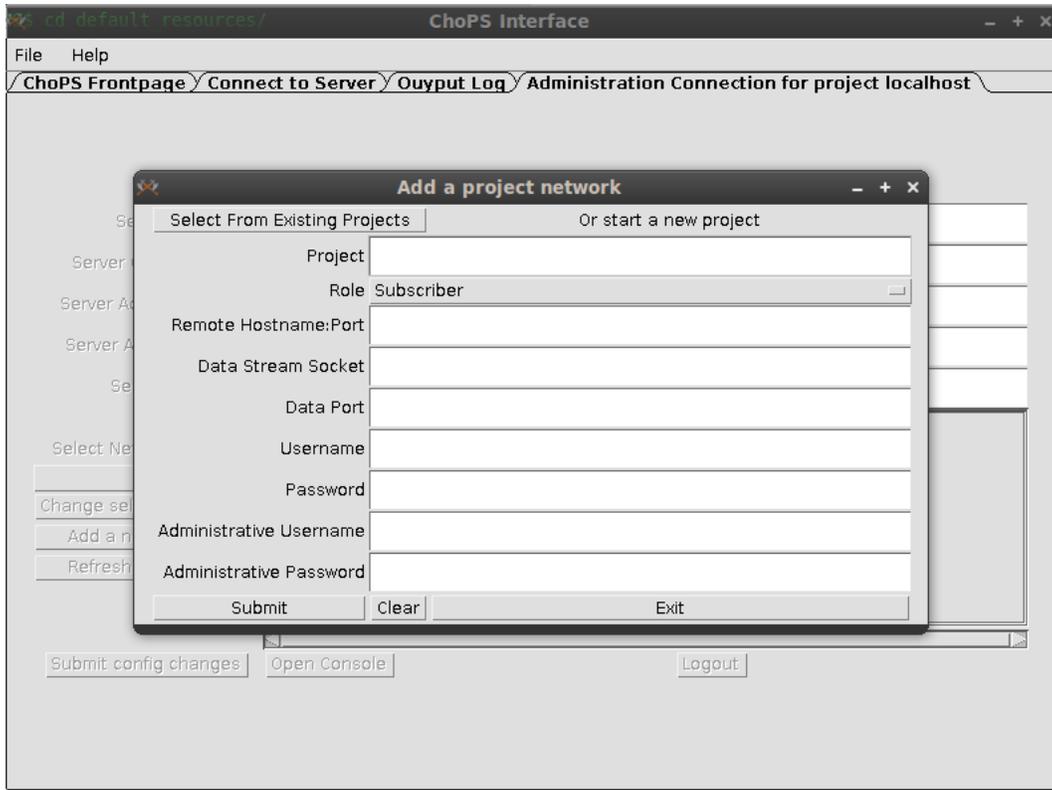


Figure 10: Adding a ChoPS Project/Network

We can then fill in some data to mimic connecting to a project.

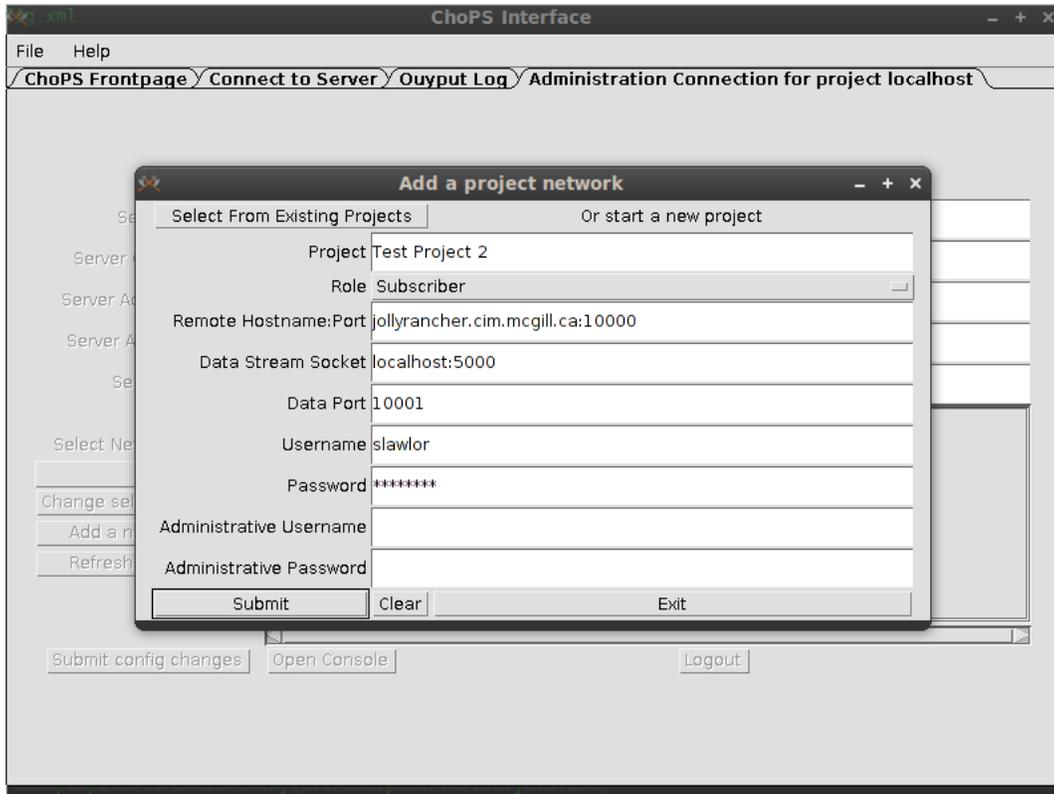


Figure 11: Adding a subscription to a ChoPS Project/Network

Notes About Subscriber Project Data:

- The Remote Hostname:Port should be in the format host/ip:port and be a **single** system already participating in the ChoPS project you are trying to connect to as well as that remote system's *Data Port*. The load-balancing and update to transmittable hosts will be done automatically.
- The Data Stream Socket is the socket which ChoPS will create a new Server Socket and replicate all data received to that port for local processing
- The Data Port is the port for all ChoPS communications between each other and is unique to a project on the specified host, meaning that you can only run a single ChoPS project on a single port. (Same as Publisher)
- The Username/Password are, as of yet, unused so you may put whatever you wish in these fields, as long as they are not empty.
- Administrative Username and Password are uneditable as a Subscriber, and this feature set will be explained when more features are added to the system.

We can then hit *Submit* and we will have the same behavior as with the Publisher creating a new project/network and we need to refresh the list of projects, and click on the new item in the drop-down list of projects for the data to validate.

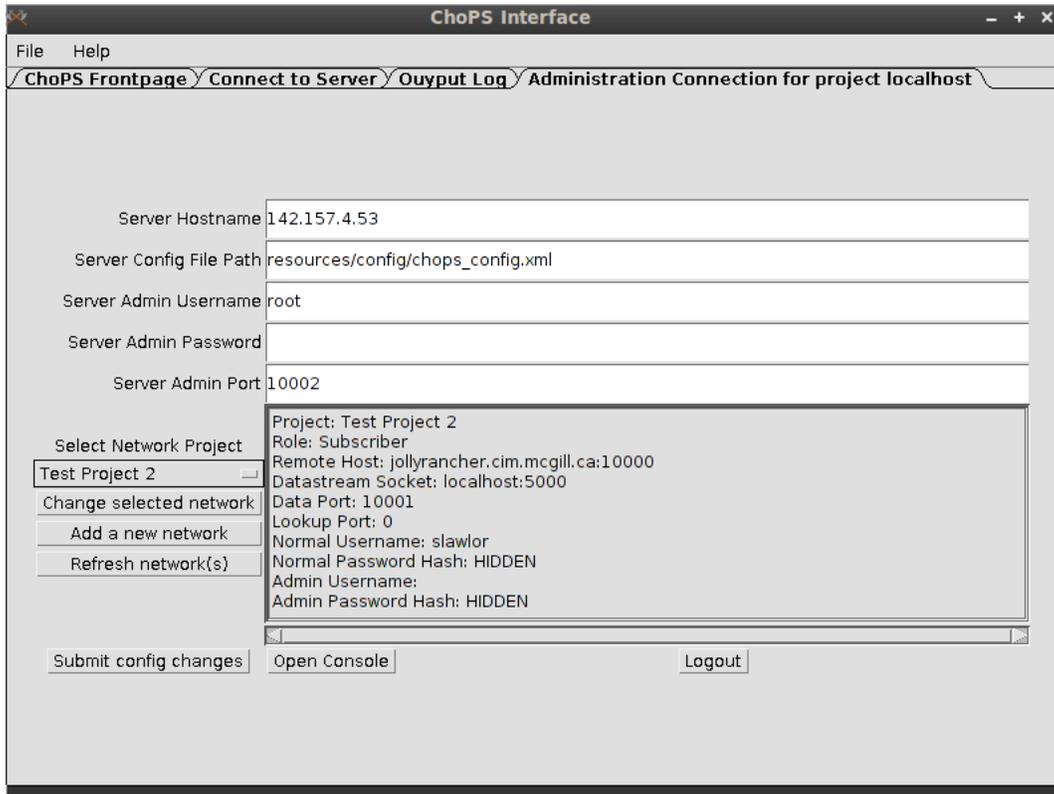


Figure 12: Validating a new subscription project

And again as with the Publisher, we simply submit the configuration changes and the system will go live.

4 Notes

1. There are still some error messages that need to be passed from Gui-to-client and vice-versa, so for now you'll need to check the logged output to make sure the project starts successfully when starting a project.