Branston User Guide

By

Group A

* Kwan Sin Chan
* Daniel Davis
* Micah Duchesne
* Jonathan Darrell Jackson
* Miguel Aires Ribeiro Valadas

Submitted to

The University of Liverpool

MASTER-OF-SCIENCE-COMPUTER-SCIENCE

CSCK541-Software Development in Practice

# Table of Contents

[Table of Contents 1](#_Toc671940226)

[Requirements 2](#_Toc1367863008)

[Server Wizard 3](#_Toc206147223)

[Client Wizard 4](#_Toc45207832)

[Dictionary 5](#_Toc1007952547)

[Textfile 5](#_Toc1532372106)

[Security 6](#_Toc1384756710)

[Actions once Client Wizard is completed 6](#_Toc1920515726)

# Requirements

User guides should be created to provide a walkthrough on how to:

* Use the Dictionary function
* Encrypt a file
* Decrypt a dictionary
* Send a non-encrypted file
* Send an encrypted file
* Disable the connection between the client and the server

# Server Wizard

The Server Wizard is a console application which gathers the settings required to instantiate the Server.

Prompt: Please configure the Branston Server for clients on a single PC or multiple PCs

User Input: s (for server and client on a single PC), or m (for server and client on multiple PCs)

Prompt: Please choose the Server Port

User Input: A number from 1024 to 65535 (to avoid reserved ports 1-1023)

Prompt: Please choose to output received data to console or textfile

User Input: c (for console print), or t (for textfile)

Prompt: Please choose where to save textfiles

User Input: Directory

Once these setting have been gathered, the server is initialized, and listens for connections from clients. The server hostname and port number as instantiated, must be supplied to a clients who wish to connect to it.

# Client Wizard

Prompt: Please type the hostname of the Server

User Input: IP address of the Server

Prompt: Please choose the Server Port

User Input: Port Number (in the range 1024 to 65536)

Prompt: Do you want to send a dictionary or a textfile?

User Input: d (for dictionary), or t (for textfile)

Depending on the User’s choice, they will now be guided in alternative paths to create a dictionary on one hand, or to select or build a textfile on the other.

## Dictionary

If the user chose to create a dictionary, a simple dictionary builder is invoked.

Prompt: Choose a pickling format: Binary, JSON, or XML

User Input: b (for binary), j (for JSON), or x (for XML)

Prompt: Now we will build the dictionary

Prompt: Item 1 key

User Input: the first key

Prompt: Item 1 value

User Input: the first value

Prompt: type x if that was your last item

Depending on the user input, the dictionary builder will either stop or ask for the key and value for the next item.

A summary of all the user input is displayed.

## Textfile

If the user chose the textfile option then they are asked the following questions.

Prompt: Do you want to upload an existing textfile?

User Input: y (yes), or n (no)

If the user chose to upload an existing textfile they are asked:

Prompt: Please input the filepath to the textfile

User Input: the filepath to the text file

If the user chose not to upload an existing textfile then a new file must be created.

Prompt: Please choose where to save textfiles

User Input: Directory

then a simple text file builder is invoked:

Prompt: Now we will build the textfile

Prompt: Line 1 text

User Input: The first line of text

Prompt: type 'x' if that was your last line:

Depending on the user input, the textfile builder will either stop or ask for the next line of text.

A summary of all the user input is displayed.

## Security

The user is then prompted for the required level of security.

Prompt: Do you want to encrypt the file?

User Input: y (yes), or n (no)

# Actions once Client Wizard is completed

Once these setting have been gathered, the client is initialized, and requests connection to the server. Dictionaries are serialized ackording the to the chosen serialisation format. Textfiles are loaded into a text stream.

If the security level chosen was “encrypted” then the client requests the public key from the server.

The stream is then encrypted.

The client handles all further messaging and delivers the payload to the server.

The server accepts connections from clients.

* Receives the client settings
* Sends the public key if requested.
* Receives the payload.
* Decrypts the stream if required.
* Deserialises the data according to the chosen format.
* Outputs either the dictionary to the server console, or saves the textfile.