# Tests

| **Test #** | **Test procedure / description** | **Expected Outcome** | **Actual Outcome & Remarks** | **Pass / Fail** |
| --- | --- | --- | --- | --- |
| 1 | 1. Start the application 2. Left click folder list items on the local files view | The application navigates to the selected directory, and shows the contents of the directory. | As expected, with some exceptions | OK; bug 1 |
| 2 | 1. Start the application 2. Left click a file list item on the local files view | The file is uploaded to the remote host | An error message was displayed, indicating that there was no remote host to connect to...this is a valid outcome. | Pass |
| 3 | 1. Start the application 2. Start the server on localhost 3. Connect to the server on localhost | The client connects successfully to the server | As expected | Pass |
| 9 | Repeat test 7 | Same as test 7 | As expected | Pass |

# Bugs

| **Bug #** | **Steps to reproduce & effect** | **Cause** | **How it was resolved** | **Solved** |
| --- | --- | --- | --- | --- |
| 1 | Navigate through some directories, and try to navigate into a directory with elevated privileges.  The application fails to navigate into the directory, and does nothing. | The application does not have the permissions to read the directory. | Run the application in administrator mode | Kind of |
| 2 | Upload a text file to the remote host.  Open the uploaded file on the remote host, and compare the last bit of this file with the last bit of the original file. They do not match. | The protocol for transferring files always writes the whole buffer into the file, even if it was only partially filled with new data from read operations, causing old data to be written to the new file. | Only write as much data to the file as was read from the socket. | Yes; test 8 & 9 |

# Screenshots