```
Index: C:/Users/chris/workspaces/CSCI602/jEdit/jars/commons-net-3.3.jar
Cannot display: file marked as a binary type.

Index: C:/Users/chris/workspaces/CSCI602/jEdit/.classpath
@@ -5,5 +5,6 @@
    <classpathentry kind="src" path="jars/MacOS"/>
    <classpathentry kind="src" path="jars/QuickNotepad"/>
    <classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/jre6"/>
+   <classpathentry kind="lib" path="/java_ftp_sample/commons-net-ftp-2.0.jar"/>
    <classpathentry kind="output" path="bin"/>
 </classpath>

Index: C:/Users/chris/workspaces/CSCI602/jEdit/org/gjt/sp/jedit/browser/FTPFileChooserDialog.java
@@ -0,0 +1,253 @@
+package org.gjt.sp.jedit.browser;
+
+import java.awt.event.ActionEvent;
+import java.awt.event.ActionListener;
+import java.io.File;
+import java.io.FileInputStream;
+import java.io.FileOutputStream;
+import java.io.FileWriter;
+import java.io.InputStream;
+
+import javax.swing.GroupLayout;
+import javax.swing.JButton;
+import javax.swing.JFrame;
+import javax.swing.JLabel;
+import javax.swing.JPanel;
+import javax.swing.JTextField;
+import javax.swing.WindowConstants;
+
+import org.apache.commons.net.ftp.FTP;
+import org.apache.commons.net.ftp.FTPClient;
+import org.apache.commons.net.ftp.FTPFile;
+import org.gjt.sp.jedit.Buffer;
+import org.gjt.sp.jedit.View;
+import org.gjt.sp.jedit.jEdit;
+import org.gjt.sp.util.Log;
+/**
+ * A UI component for requesting FTP login/filename data from the user to enable
+ * remote file editing via Apache's FTPClient.
+ * @author Chris Cargile
+ * @since 11/22/13
+ */
+@SuppressWarnings("serial")
+public class FTPFileChooserDialog extends JFrame implements ActionListener {
+
+   static String userHome=System.getProperty("user.home")+"\\FTPtemp\\";
+   static File file;
+   static File ftpTemp = new File(userHome);
+   static Buffer b;// <- current,active buffer
+   JTextField textField1; // FTP login data fields
+   JTextField textField2;
+   JTextField textField3;
+   JTextField textField4;
+   JTextField textField5;
+   static JPanel panel = new JPanel();
+   public JButton saveButton;
+   public JButton getButton;
+   static final JFrame frame = new JFrame("FTP Client Action-Settings");
+   static FTPFileChooserDialog instance;
+   /**
+    * help on java streaming-IO came from:
+    * http://docs.oracle.com/javase/tutorial/essential/io/
+    */
+   public FTPFileChooserDialog() {
+       instance=this;
+       GroupLayout layout = new GroupLayout(getContentPane());
+       getContentPane().setLayout(layout);
+       saveButton = new JButton("PUT(STOR) file");
+       saveButton.addActionListener(this);
+
+       getButton = new JButton("GET file");
+       getButton.addActionListener(this);
+
+       textField1 = new JTextField("ftp.charlestonwebapps.com");
+       textField2 = new JTextField("introswe@charlestonwebapps.com");
+       textField3 = new JTextField("fall2013");
+       textField4 = new JTextField("/");
+       textField5 = new JTextField("");
+
+       // labels added to JFrame
+       JLabel server = new JLabel("FTP Server:");
+       JLabel fuser = new JLabel("FTP User:");
+       JLabel fpass = new JLabel("FTP Pass:");
+       JLabel fdir = new JLabel("FTP Dir:");
+       JLabel fFilename = new JLabel("<html>FTP File:<BR>"+
+       "(leave blank to remotely save-as buffers name<BR>"
+           + " or to GET remote file with buffers name (if it exists))");
+
+       // in these(3) columns, we'll list the items' vertical placement in order
+       layout.setHorizontalGroup(layout
+           .createSequentialGroup()
+           .addGroup(
+                   layout.createParallelGroup().addComponent(server)
+                           .addComponent(fuser).addComponent(fpass)
+                           .addComponent(fdir).addComponent(fFilename))
+           .addGroup(
+                   layout.createParallelGroup().addComponent(textField1)
+                           .addComponent(textField2)
+                           .addComponent(textField3)
+                           .addComponent(textField4)
+                           .addComponent(textField5))
+           .addGroup(
+                   layout.createParallelGroup().addComponent(getButton)
+                           .addComponent(saveButton)));
+
+       // in these rows, we'll list the items' horizontal placement in order
+       layout.setVerticalGroup(layout
+           .createSequentialGroup()
+           .addGroup(
+                   layout.createParallelGroup().addComponent(server)
+                       .addComponent(textField1)
+                       .addComponent(getButton))
+           .addGroup(
+                   layout.createParallelGroup().addComponent(fuser)
+                           .addComponent(textField2)
+                           .addComponent(saveButton))
+           .addGroup(
+                   layout.createParallelGroup().addComponent(fpass)
+                           .addComponent(textField3))
+           .addGroup(layout.createParallelGroup().addComponent(fdir).
+                   addComponent(textField4))
+           .addGroup(layout.createParallelGroup().addComponent(fFilename)
+                   .addComponent(textField5))
+       );
+       setTitle("FTP Upload");
+       pack();
+       //setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
+   }
+
+   public void actionPerformed(ActionEvent e) {
+       View v=jEdit.getActiveView();
+       b=v.getBuffer();
+       String path = b.getPath();
+       //we assume(/assure) buffer/file-names won't contain '\' or '/'
+       int backSlash = path.lastIndexOf("\\")+1;
+       int forwardSlash = path.lastIndexOf("/");
+       // index @ where filename begins (after last directory "/")
+       int index = (backSlash > forwardSlash ? backSlash : forwardSlash);
+       path=path.substring(index, path.length()); // (String)path<-filename less path
+       String website = textField1.getText();
+       String username = textField2.getText();
+       String password = textField3.getText();
+       String directory = textField4.getText();
+       //set filename<-buffer's base-filename or....'Untitled-xyz.txt' as a fallback
+       String filename= ( (b.getPath()==null || b.getPath().equals("") ) ? "Untitled-xyz.txt":path);
+       if(textField5.getText()!=null && textField5.getText().length()>0)
+           filename=textField5.getText();
+       ftpTemp=new File(userHome);
+       ftpTemp.mkdir();
+       ftpTemp=new File(userHome);
+       // helper library credit due: http://commons.apache.org/proper/commons-net/apidocs/org/apache/commons/net/ftp/FTPClient.html
+       FTPClient client = new FTPClient();
+       try {
```

```
+     if (e.getSource() == getButton) {
+         client.connect(website);
+         client.login(username, password);
+         client.setFileType(FTP.BINARY_FILE_TYPE);
+         file = new File(userHome,filename);
+         if(file.isFile()){  //delete and re-create current buffer-named file:
+           String f= file.getPath();
+           if(file.delete())
+             Log.log(Log.WARNING, instance.getClass(),"deleted by ftp retreiver: local file: "+f);
+         }
+         file.createNewFile();
+         if(client.retr(filename)==553){
+           new File(filename).renameTo(file); // move "jEditProgramRoot/filename" to "user.home/ftpTemp/filename"
+           Log.log(Log.WARNING, instance.getClass(), "retr ok"+ client.retr(filename));//code 553 indicates FTP re-/created local file from FTP-remote
+         }
+         else{
+           Log.log(Log.WARNING, instance.getClass(), "FTP \"retr\" "+filename+" not ok. status:"+
+               client.retr(filename)+",trying retrieveFile method instead:\n");//+
+           FileOutputStream fos = new FileOutputStream(file);
+           if(client.retrieveFile(filename, fos)){ //per FTPClient API, true if retrieval succeeds
+             fos.close();
+             new File(filename).renameTo(file); // move "jEditProgramRoot/filename" to "user.home/ftpTemp/filename"
+             Log.log(Log.WARNING, FTPFileChooserDialog.class, "retrieveFile() succeeded. Reconstructed dl file at: " + file.getCanonicalPath());
+           }
+         }
+         // this is done so openFile recognizes if we re-open current buffer-named files
+         if(b.isUntitled())
+           b.setNewFile(false);
+         jEdit.openFile(v, file.getParent(), file.getName(), false,null);
+     }
+     if (e.getSource() == saveButton) {
+       client.connect(website);
+       client.login(username, password);
+       client.setFileType(FTP.BINARY_FILE_TYPE);
+       // textField5 not empty so that'll be what the buffer is saved to
+       if(!textField5.getText().equals("")){
+         file = new File(userHome,textField5.getText());
+         String s = b.getText(0, b.getLength());
+         file.delete();
+         file.createNewFile();
+         if(file.canWrite()){
+           FileWriter writer = new FileWriter(file);
+           writer.write(s);
+           writer.close();
+         }
+         else{
```

```
+                       Log.log(Log.WARNING, FTPFileChooserDialog.class,"cannot write file");
+                   }
+               }
+               // else if: textField left empty but buffer IS named
+               else if(textField5.getText().equals("") && !((b.getPath()==null || b.getPath().equals("")))){
+                   file= new File(userHome,path);
+                   file.delete();
+                   file.createNewFile();
+                   String s = b.getText(0, b.getLength());
+                   //  write the contents of the current buffer to 'file'
+                   if(file.canWrite()){
+                       FileWriter writer = new FileWriter(file);
+                       writer.write(s);
+                       writer.close();
+                   }
+                   else{
+                       Log.log(Log.WARNING, FTPFileChooserDialog.class,
+                               "cannot write file");
+                   }
+               }
+               InputStream localInputStr = null;
+               if (file.isFile()) {
+                   localInputStr = new FileInputStream(file.getPath());
+                   client.dele(file.getName());
+                   boolean result =client.appendFile(file.getName(), localInputStr);
+                   localInputStr.close();
+                   Log.log(Log.WARNING, FTPFileChooserDialog.class,"file: "+file.getPath()+" saved to: " + website+""+directory);
+               }
+           } //end if(saveButton==source)
+           if(client.abort())
+               ;//Log.log(Log.WARNING, instance.getClass(), "FTP Client Connection closed");
+       }
+       catch (Exception e1) {
+           Log.log(Log.WARNING, instance.getClass(), e1.getMessage());
+       }
+       instance.setVisible(false);
+       instance.dispose();
+   }// end actionPerformed(ActionEvent) method
+
+   /** A convenience method to remove the FTPtemp directory
+    *  Note: the method does not work on Windows8 due probably to Win8 UAC
+    *  restrictions
+   **/
+   public void removeFTPTemp(){
+       try{
+           File[] filesInTemp;
+           if(ftpTemp!=null){
+               if(ftpTemp.isDirectory()){
+                   //delete files in directory first
+                   //or removing directory will fail
+                   filesInTemp=ftpTemp.listFiles();
+                   for(File namedFile:filesInTemp)
+                       namedFile.delete();
+                   if(ftpTemp.delete())
+                       ;//Log.log(Log.WARNING, instance.getClass(), "deleted directory:"+ftpTemp.getPath());
+               }
+           }
+           else
+               Log.log(Log.WARNING, instance.getClass(), "FTPTemp directory not deleted");
+       }
+       catch(Exception e){
+           Log.log(Log.WARNING, instance.getClass(), "exception: "+e);
+       }
+   } // end remoteFTPTemp() method
+}// end class

Index: C:/Users/chris/workspaces/CSCI602/jEdit/org/gjt/sp/jedit/actions.xml
@@ -614,4 +614,4 @@
+<ACTION NAME="open-remote-file">
+   <CODE>
+      GUIUtilities.showFTPFileChooserDialog();
+   </CODE>
+</ACTION>
+

Index: C:/Users/chris/workspaces/CSCI602/jEdit/org/gjt/sp/jedit/jedit_gui.props
@@ -27,6 +27,7 @@
 new-file.shortcut=C+n
 open-file.shortcut=C+o
 open-read-only-file.shortcut=C+r
+open-remote-file.shortcut=C+e
 print.shortcut=C+p
 exit.shortcut=C+q
 # C+r is a prefix
@@ -180,6 +180,7 @@
 file=new-file \
      open-file \
      open-read-only-file \
+     open-remote-file \
      %recent-files \
      - \
      reload \
@@ -202,6 +203,7 @@
 new-file.label=$New
 open-file.label=$Open...
 open-read-only-file.label=$Open As Read Only...
+open-remote-file.label=Op$en/Save File using FTP...
 reload.label=$Reload
 reload-all.label=Reloa$d All...
 close-buffer.label=$Close

Index: C:/Users/chris/workspaces/CSCI602/jEdit/org/gjt/sp/jedit/GUIUtilities.java
@@ -49,14 +49,30 @@
```

```diff
 import java.util.Map;
 import java.util.StringTokenizer;

-import javax.swing.*;
+import javax.swing.Box;
+import javax.swing.BoxLayout;
+import javax.swing.Icon;
+import javax.swing.ImageIcon;
+import javax.swing.JButton;
+import javax.swing.JComponent;
+import javax.swing.JDialog;
+import javax.swing.JLabel;
+import javax.swing.JList;
+import javax.swing.JMenu;
+import javax.swing.JMenuBar;
+import javax.swing.JMenuItem;
+import javax.swing.JOptionPane;
+import javax.swing.JPanel;
+import javax.swing.JPopupMenu;
+import javax.swing.JScrollPane;
+import javax.swing.JSplitPane;
+import javax.swing.SwingUtilities;

-import org.gjt.sp.jedit.View;
+import org.gjt.sp.jedit.browser.FTPFileChooserDialog;
 import org.gjt.sp.jedit.browser.VFSFileChooserDialog;
 import org.gjt.sp.jedit.gui.EnhancedButton;
 import org.gjt.sp.jedit.gui.FloatingWindowContainer;
 import org.gjt.sp.jedit.gui.SplashScreen;
-import org.gjt.sp.jedit.gui.ToolBarManager;
 import org.gjt.sp.jedit.gui.VariableGridLayout;
 import org.gjt.sp.jedit.menu.EnhancedCheckBoxMenuItem;
 import org.gjt.sp.jedit.menu.EnhancedMenu;
@@ -738,6 +754,12 @@
             JOptionPane.YES_NO_OPTION,
             JOptionPane.QUESTION_MESSAGE);
    } //}}}
+
+   //{{{ showFTPFileChooserDialog() method
+   public static void showFTPFileChooserDialog(){
+       (fileChooser = new FTPFileChooserDialog()).setVisible(true);
+       fileChooser.removeFTPTemp();
+   } //}}}

    //{{{ showVFSFileDialog() method
    /**
@@ -1706,2 +1708,2 @@
+   private static FTPFileChooserDialog fileChooser;
    private static boolean isEdit;

```