

Plugin-Tutorial

Table of contents

Tutorial: Viewer Plugin	1
-------------------------------	---

Tutorial: Viewer Plugin

Maybe that can give you a starting point. This will implement a TextViewer as Plugin.

First: get the source-package

Simple type

```
ant newplugin -Dplugin.name=textviewer
```

. This will generate a sub-directory "plugins/other/textviewer". If you use [Eclipse](#) type:

```
ant eclipse-project
```

this will generate the project-files for Eclipse and you can import Hawron (Import Project).

Your new Plugin-folder should be on the right place now.

Next: create a MyTextViewerPlugin.java in the src-folder and paste the code below

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

//the hawron interfaces
import de.miethxml.hawron.gui.context.viewer.CacheableViewer;
import de.miethxml.hawron.gui.context.viewer.ViewerCloseListener;

public class MyTextViewerPlugin implements CacheableViewer {
    private JFrame frame;
    private JTextArea textarea;
    //the ViewerCloselisters
    private ArrayList listeners = new ArrayList();

    /**
     * the parameter empty constructor is used always
     */
    public MyTextViewerPlugin() {
        super();
    }
}
```

```
public void addViewerCloseListener(ViewerCloseListener listener) {
    listeners.add(listener);
}

public void removeViewerCloseListener(ViewerCloseListener listener) {
    listeners.remove(listeners);
}

public void dispose() {
    //we do nothing here
}

public void destroy() {
    frame.dispose();
    frame= null;
}

public void init() {
    frame = new JFrame("MyTextViewerPlugin");
    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            //we do this for caching
            frame.setVisible(false);
            fireCloseEvent();
        }
    });

    textarea = new JTextArea();
    //only viewing
    textarea.setEditable(false);
    JScrollPane scrollpane = new JScrollPane(textarea);
    frame.getContentPane().add(scrollpane, BorderLayout.CENTER);
    frame.setSize(new Dimension(640, 480));
}

public void setVisible(boolean state) {
    frame.setVisible(state);
}

public void open(String file) {
    //we get now the file and put the content to the textarea.
    //delete the old content
    textarea.setText("");

    try {
        BufferedReader in = new BufferedReader( new FileReader(file));
        String line=null;
        while(( line = in.readLine()) != null){
            textarea.append(line+"\n");
        }
        in.close();
    }
```

```
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException ioe){
            ioe.printStackTrace();
        }
    }

    public boolean isSupported(String file) {
        //we will support text-files
        if (file.toLowerCase().endsWith(".txt")) {
            return true;
        }
        //we will support log-files too
        if (file.toLowerCase().endsWith(".log")) {
            return true;
        }
        return false;
    }

    public Icon getIcon() {
        //if we have a icon we place it in the jar-file later
        //otherwise we get a button without a icon
        URL imageurl = this.getClass().getResource("icon.gif");
        ImageIcon icon = new ImageIcon(imageurl);
        return icon;
    }

    public String getToolTip(String lang) {
        return "MyTextViewer";
    }

    public Viewer createNewViewer() {
        MyTextViewerPlugin viewer = new MyTextViewerPlugin();
        viewer.init();
        return viewer;
    }

    private void fireCloseEvent() {
        //avoid problems by removing the listeners
        Iterator i = ((List)listeners.clone()).iterator();
        while (i.hasNext()) {
            ViewerCloseListener listener = (ViewerCloseListener)i.next();
            listener.close(this);
        }
    }
}
}
```

Write the plugin-descriptor (edit the auto-generated file in the plugin-directory)

```
<?xml version="1.0" ?>
<plugin>
  <name>My plugin</name>
```

```
<instanceclass>MyTextViewerPlugin</instanceclass>
<interface>de.miethxml.hawron.gui.context.viewer.Viewer</interface>
<description>My plugin</description>
<resources>
  <classpath src="textviewer.jar"/>
</resources>
</plugin>
```

Build now

- uncomment the entry "textviewer=true" in the local.plugin.properties (the copy of plugin.properities) to enable building
- Try:
ant
This will build the plugin and put it to dist/plugins.
- if you have a "icon.gif" put it to the build/classes-folder of the plugin and again in the main Hawron directory (only for putting the icon into the jar, you can change the plugin-buildfile for this too)
- change to the dist-directory and start Hawron by:
./run.sh or run (Windows or the hawron.exe-wrapper)

Open a project and view log- or textfiles with your plugin now. You can see if you select a file, which is not supported by your plugin, it will disabled

Maybe you will now write a extended logfileviewer, please contrib it.