ThickClientScreenPop Sample Gadget

There are several components to the ThickClientScreenPop. There is the Finesse gadget ThickClientScreenPop.xml, a Java applet JavaScreenPop.jar, and a Microsoft Windows Application WindowsApplication.exe

Environment:

This gadget involves an applet. And, hence some browser / client-side configuration is involved. This gadget is tested on a **Microsoft® Windows 7 (64-bit)** with **Internet Explorer 11 and JRE 8 Update 101.** Also, this is tested ONLY on Microsoft Windows since it involves running a Microsoft® .NET application.

Google Chrome has stopped support for Java Applets. See http://www.java.com/en/download/faq/chrome.xml.

This gadget has NOT been tested using Mozilla Firefox.

Steps to deploy:

1. Make sure to reboot the machine if you just installed Java.

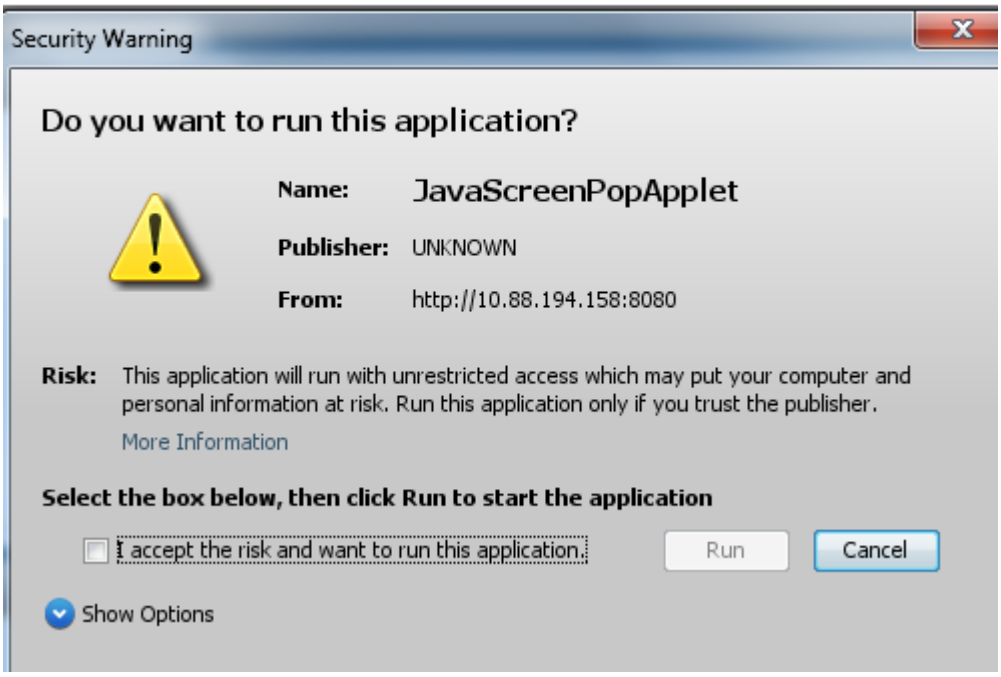
2. Place JavaScreenPop.jar where it is downloadable from your web server

3. Edit ThickClientScreenPop.xml applet codebase parameter to point to the folder on your remote server where the applet (JavaScreenPop.jar) resides

```
<APPLET CODE="JavaScreenPopApplet.class"
codebase="http://10.88.194.158:8080/userContent/ThickClientScreen
Pop" archive="JavaScreenPop.jar" NAME="JavaScreenPopApplet"
WIDTH=350 HEIGHT=100 >
</APPLET>
```

4. Copy the gadget files ThickClientScreenPop.xml, ThickClientScreenPop.js and ThickClientScreenPop.css and the Finesse javascript library and the jQuery javascript library that are included with the gadget files to your Finesse server or to your own gadget server

5. Add the gadget to your desktop layout in the Finesse Administration: ThickClientScreenPop.xml If your gadget is set up correctly when the agent logs in the browser requests to accept the applet. You will see something similar to:



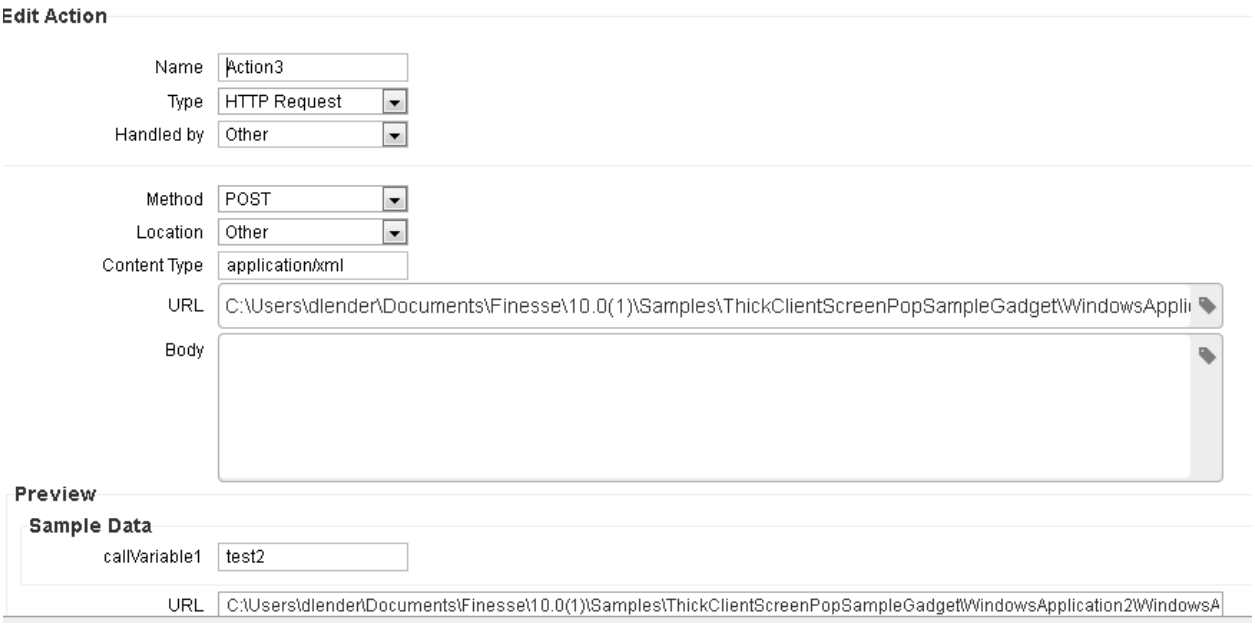
Click the box to accept the risk and the Run button will be highlighted.

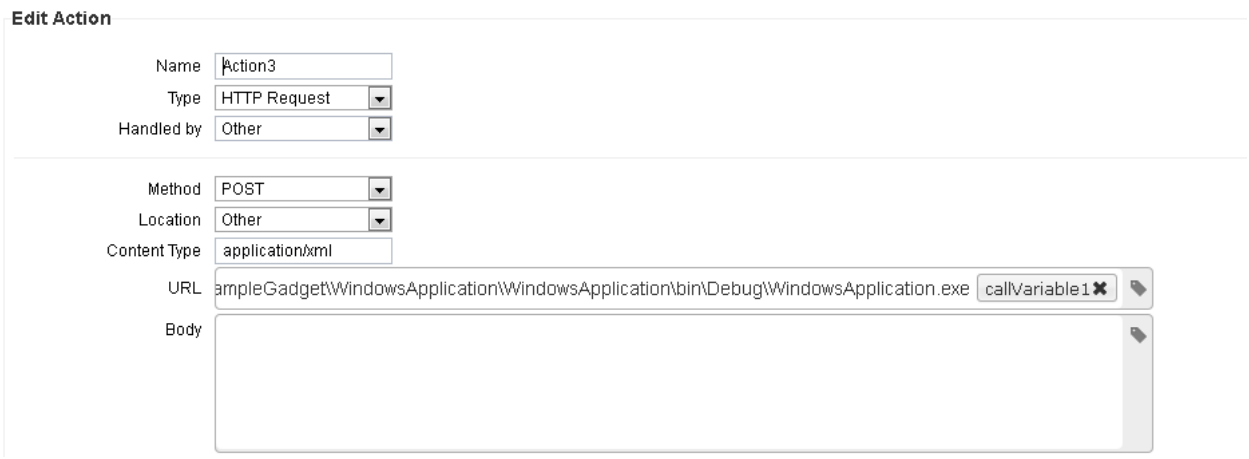
Click Run and the applet is ready to go.The gadget should look like this:

6. From within the Finesse Administration application, create a workflow and workflow action and assign it to the team the agent belongs to.

7. For the workflow action set handled by to Other and type to http request.

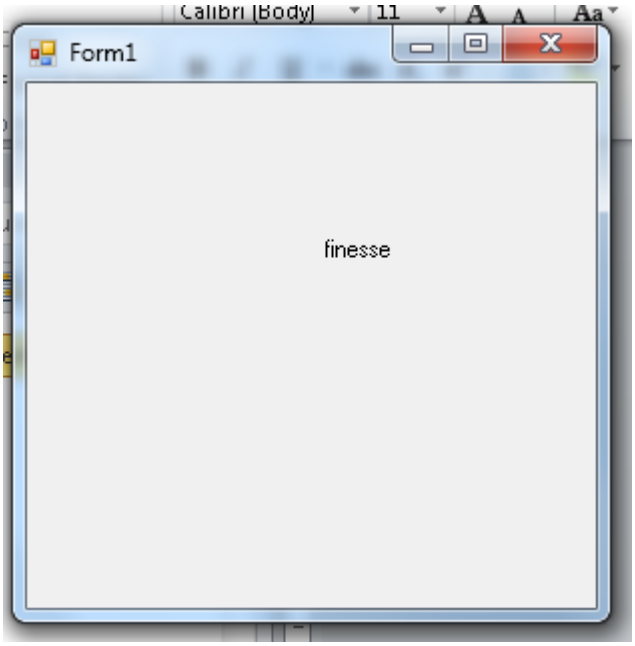
8. In the URL field put the location of the WindowsApplication.exe and add any call variables after the .exe (e.g. callvariable1)

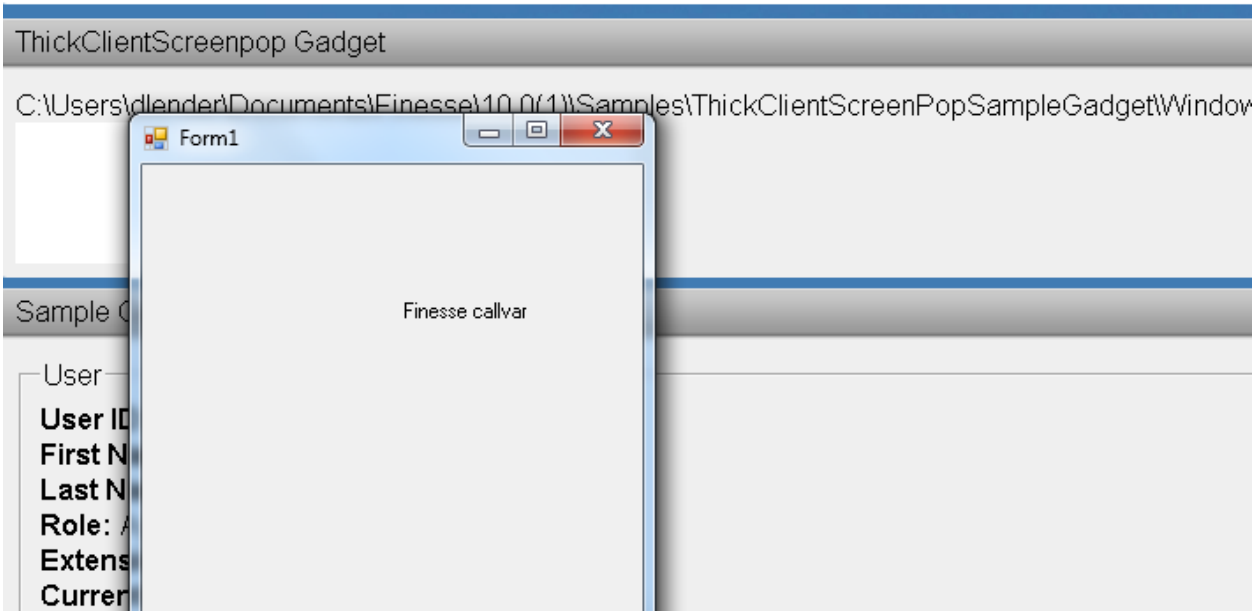




Note: you will have to refresh the agent desktop after configuring the workflow(s).

9. If the action is configured correctly and the workflow is assigned to a team, when a call is placed to the agent with callvariable1 the workflowactionevent is triggered and the thick client application is displayed:





Note: You cannot test the thick client screenpop from the administration page. You must actually place a call to the agent to execute a workflow and test the thick client screen pop.

To verify the location of the thick client screenpop that the gadget is receiving you can check the debugger Console log:

```
131213 15:52:37.313 : ThickClientScreenpop : About to pop thick client.
URL is:
C:\Users\dlender\Documents\Finesse\10.0(1)\Samples\ThickClientScreenPop
SampleGadget\WindowsApplication\WindowsApplication\bin\Debug\WindowsApp
lication.exe finesse
```

Where “finesse” at the end is the callvariable1 that was passed on the dialog.

JavaScreenPop.jar

The source code for JavaScreenPop is included and is provided as is. The applet has a single method that invokes whatever command line is set. The jar is signed with a Self-Signed Certificate. General instructions on signing a java applet are included in Self Signing Jar File.docx or at http://docs.oracle.com/javase/tutorial/deployment/jar/signing.html

WindowsApplication.exe

The WindowsApplication.exe simply displays in a label whatever arguments (i.e.. callvariables) are passed to it on the command line. The WindowsApplication.sln Microsoft Visual Studio 2010 Visual Basic solution is included with the sample and is provided as is.

Java Console

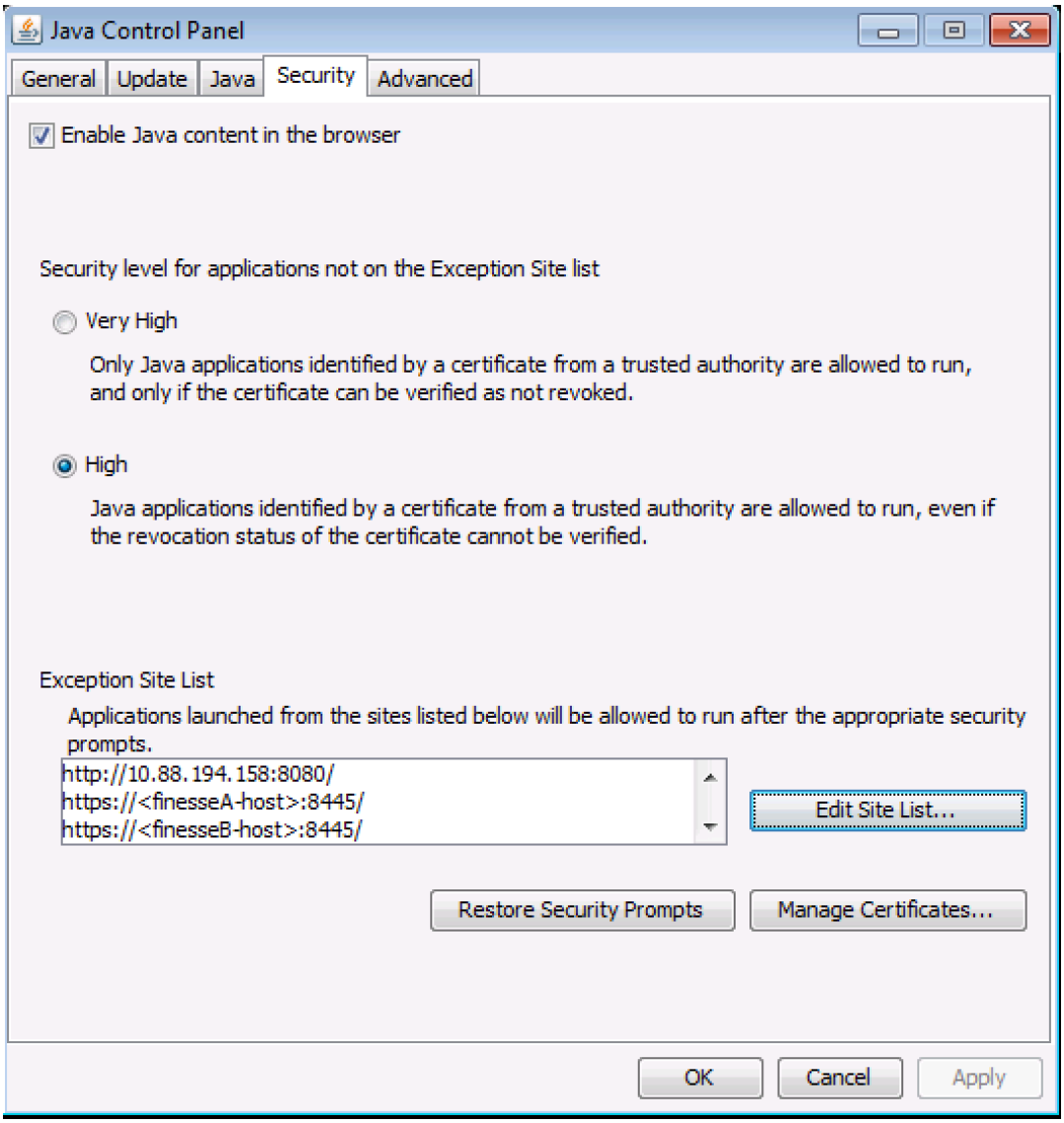
If you are having issues with the Java applet, you can see what is going on by turning on the Java console.

To turn on the Java Console, go to the Java Control Panel and turn on the console.

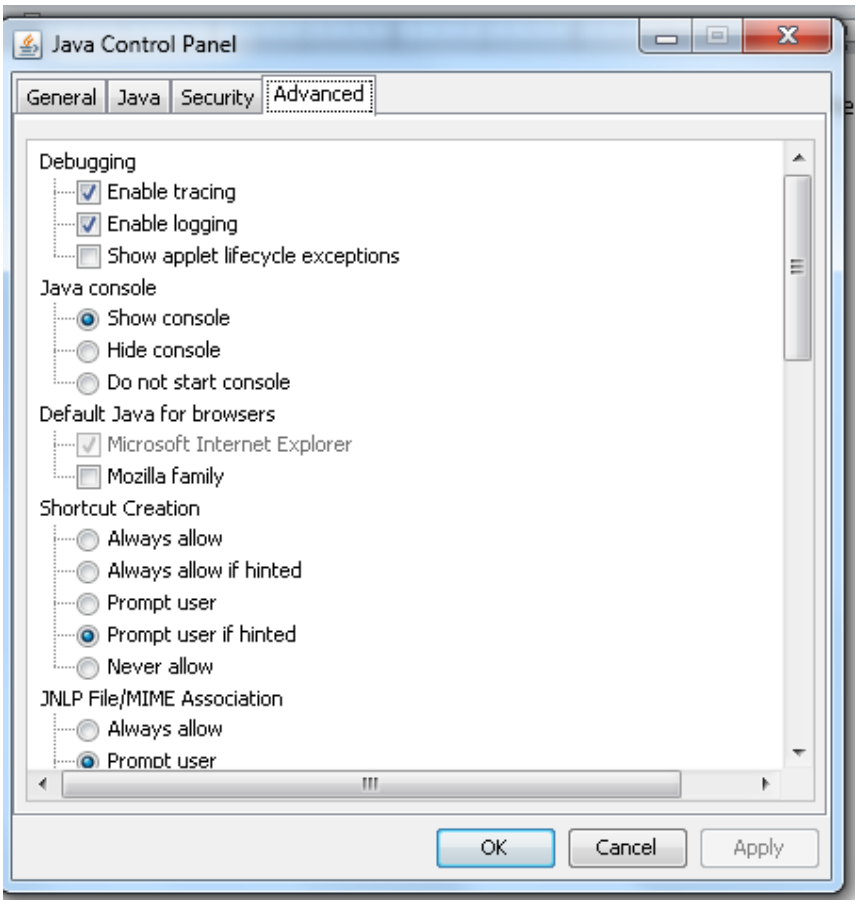
For Windows 7, go to Start->Control Panel and search for Java

Click on the Java Link and the Java Control Panel is displayed

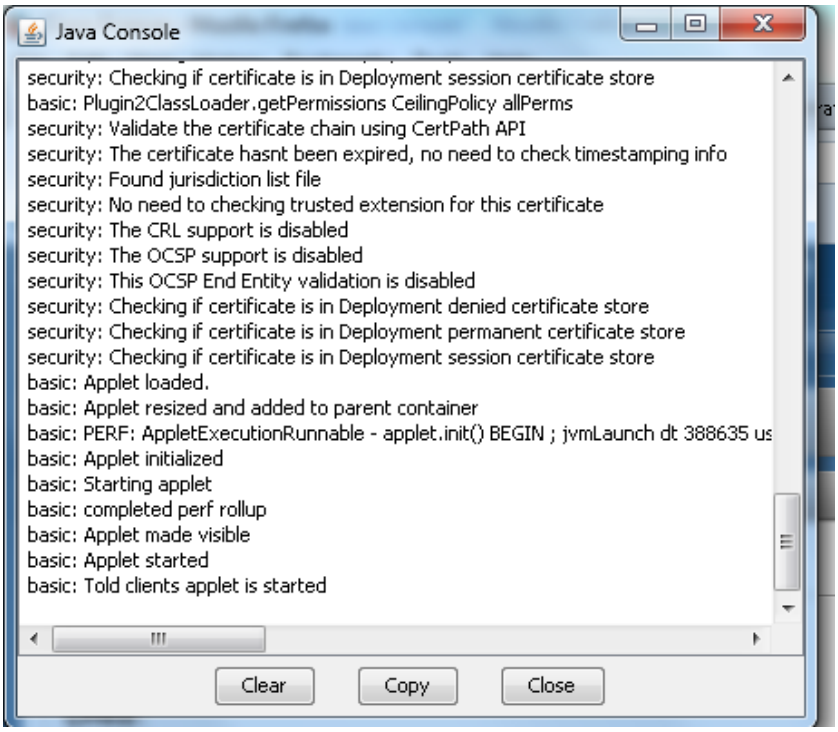
Click on the Security tab and make sure the following hosts are white-listed (i.e., the Exception Site list). Also, make sure that “Enable Java content in the browser” is checked, and the security level is at “High”.



Click on the Advanced tab and check the Show Console selection.



Here is a sample display of the Java console which should display after the security warning for the applet.


Java Console
security: Checking if certificate is in Deployment session certificate store
basic: Plugin2ClassLoader.getPermissions CeilingPolicy allPerms
security: Validate the certificate chain using CertPath API
security: The certificate hasnt been expired, no need to check timestamping info
security: Found jurisdiction list file
security: No need to checking trusted extension for this certificate
security: The CRL support is disabled
security: The OCSP support is disabled
security: This OCSP End Entity validation is disabled
security: Checking if certificate is in Deployment denied certificate store
security: Checking if certificate is in Deployment permanent certificate store
security: Checking if certificate is in Deployment session certificate store
basic: Applet loaded.
basic: Applet resized and added to parent container
basic: PERF: AppletExecutionRunnable - applet.init() BEGIN ; jvmLaunch dt 388635 us
basic: Applet initialized
basic: Starting applet
basic: completed perf rollup
basic: Applet made visible
basic: Applet started
basic: Told clients applet is started
Clear
Copy
Close