

Java Web Start

01 October 2014 11:37

Intro:

- a helper application that gets associated with a Web browser
- provides the power to launch full-featured applications with a single click. Users can download and launch applications, such as a complete spreadsheet program or an Internet chat client, without going through lengthy installation procedures.
- users can launch a Java application by clicking a link in a web page. The link points to a Java Network Launch Protocol (JNLP) file, which instructs Java Web Start software to download, cache, and run the application.

Advantage:

- place a single Java application on a web server for deployment to a wide variety of platforms, including Windows, Linux, and Solaris.
- supports multiple, simultaneous versions of the Java platform.
- Users can create a desktop shortcut to launch a Java Web Start application outside a browser.
- takes advantage of the inherent security of the Java platform. By default, applications have restricted access to local disk and network resources.
- Applications launched with Java Web Start software are cached locally for improved performance.
- Updates to a Java Web Start application are automatically downloaded when the application is run standalone from the user's desktop.
- comes with jre, no need to install separately

Launching JWS App:

- From a Web browser by clicking on a link.
- From desktop icons or the Start Menu.
- From the Java Cache View

Launching from a Web browser

Point your web browser to a page with a link to a jnlp application, and click on that link.

A security dialog will pop up with information about the origin of the application based on who digitally signed the code, and the level of access requested. The application will run only if you decide to trust the vendor.

That is really all there is to using Java Web Start, but how does it work? The HTML links that launch the applications are, in fact, standard HTML links. However, instead of pointing to another Web page, they link to a special configuration file called a JNLP file. The Web browser examines the file extension and/or the MIME type of the file, and sees that it belongs to Java Web Start. It then launches Java Web Start with the downloaded JNLP file as an argument. Java Web Start proceeds with downloading, caching, and running the application as directed by the JNLP file.

Develop Stand-alone SWT App to be executed by JWS:

- Make a Main class, with the swt code
 - Make jar of all the classes
 - make a jnlp file. This file contains info about the jars to be used for the os, the permissions, the main class to be executed
 - generate a keystore using the command: "keytool -genkey -keystore keystore -alias myself"
 - sign all the jars using the command : "jarsigner -keystore keystore abc.jar myself"
- All the jars must include the swt jar, swt app jar etc

Example: TestJWS



TestJWS

- All these files should be in the same directory
 - TO execute the app, execute the jnlp file with jws.
- Or open cmd and execute : "javaws <file:///C:/JWS/SWTHelloWorld/SWTHelloWorld.jnlp>" change the jnlp file location and name.

Develop App to be executed in Tomcat Server:

- In continuation to the above steps
- edit the jnlp file, change the codebase url to "<http://localhost:8080/<location>> of folder containing the jnlp file>"
- Make this a webapp, so add WEB-INF folder containing the web.xml file. Web.xml file contains the name of the welcome html file
- make a html file in the main folder , this file contains the url to the jnlp file
- Generate a war of this whole folder
- Deploy the war in tomcat
- open url to the deployed application



TestJWSWebApp