Java Web Start

Brad Shuler
Software Engineer
Object Computing, Inc.
St. Louis, MO

Overview

- Java Web Start (JWS)
  - **What is it?**
  - **Demo**
  - **How it works**
  - **Deployment on Server**
  - **Security**
  - **Application Manager**

Java Web Start - What is it?

- Java 2 Application Launcher
  - **Easy to Use (Browser Technology)**
  - **Free Client Program Installer From Sun**
- Reference Implementation of JNLP
  - **Java Network Launch Protocol (JNLP)**

Java Web Start - Benefits

- 1 Click Activation
  - **Browser**
  - **Desktop Icon**
  - **Start Menu**
  - **Application Manager**
- Client-Side Caching Support
- Multiple Java Runtime Environments
- Java 2 Security Model
- Applications Update Automatically

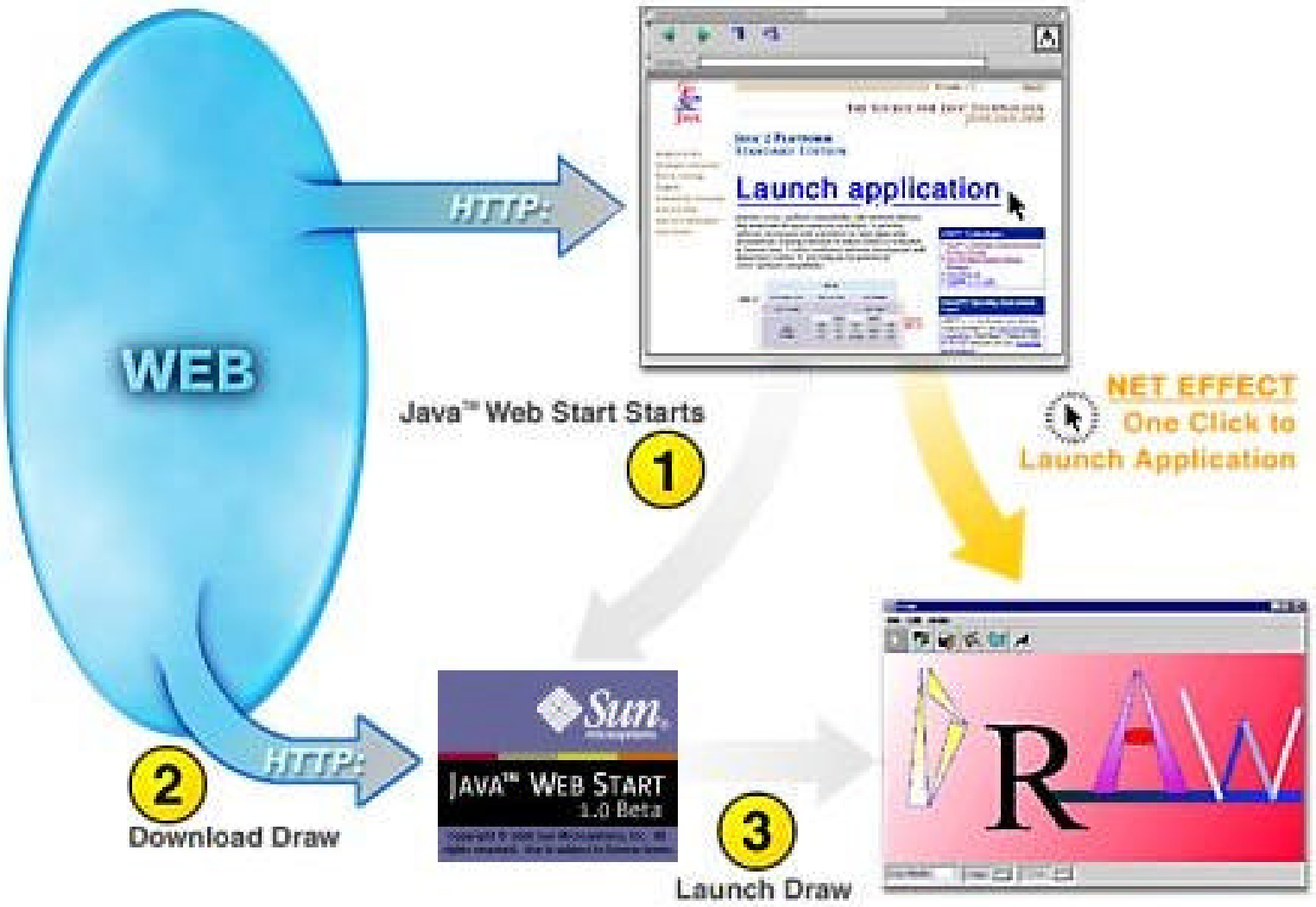
Java Web Start - Benefits

- Compared to Applets
  - **Works with any Browser**
  - **Does not require running Browser**
  - **Complex GUI Development Possible**
  - **One Time Download (uses caching)**
  - **Connection Speed Independent (In fact, no connection needed)**
  - **Realistic Security Sandbox**

Java Web Start - Benefits

- Compared to XML/HTML
  - **Slower First Use Response**
  - **Client Install Required**
  - **Sophisticated ("Fat") GUI Possible**
  - **Network Independent**

Launch With 1 Click



Java Web Start - How it Works

- Server Side Setup:
  - **New MIME Type Entry**

```
application/x-java-jnlp-file JNLP
```

  - **Deployment Manifest (.jnlp file)**
    - **Describes how the application will be launched**
    - **Describes how the application will appear in the Java Web Start Application Manager**
    - **Extensible Markup Language (XML) Format**

JNLP File

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

JNLP File - Codebase Attribute

Codebase attribute provides base URL for all href's that follow

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

JNLP File - Self Reference

Name of JNLP file itself. (Incorporates application into Web Start Application Manager)

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

JNLP File - Information Element

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
         <title>FileChooserDemo</title>
         <vendor>Object Computing, Inc.</vendor>
         <homepage href="index.html"/>
         <description>JFC FileChooserDemo App</description>
         <icon href="oci_logo.gif"/>
         <offline-allowed/>
     </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

Additional information about the application. (Visible in Application Manager, splash screen, and desktop icons)

Allow application to launch without network connection

JNLP File - Resources Element

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

Java 2 Runtime Environment for this application

JNLP File - More on j2se Element

```
<!-- Launch application only if JRE 1.3.1 is present -->
<j2se version="1.3.1"/>


<!-- Search list.  Look on local system for best JRE.
     If cannot find any 1.4 JRE, use any 1.3, then 1.2.2 -->
<j2se version="1.4+ 1.3+ 1.2.2"/>


<!-- Must use JRE 1.3.1.  Provide URL to download if not installed. -->
<j2se version="1.3.1" href="http://java.sun.com/products/autodl/j2se"/>


<!-- Specify VM parameters (these 2 only at this time) -->
<j2se version="1.3+" initial-heap-size="32m" max-heap-size="160m"/>
```

- **To see a list of what JRE's are installed on a client, launch the Java Web Start Application Manager.**

JNLP File - Resources Element

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

Place all class files, image files, and native code libraries in JAR files.

JNLP File - More on jar Element

```
<!-- This JAR contains main (main specified in manifest). -->
<jar href="application.jar" main="true"/>

<!-- Can't launch without this jar (default) -->
<jar href="application.jar" download="eager"/>
<!-- Download after launch (don't put your main class here!) -->
<jar href="jars/audio/win32/sounds.jar" download="lazy"/>

<!-- Specify a version. -->
<jar href="infrastructure.jar" version="1.22"/>

<!-- Native library. -->
<nativelib href="jars/solaris/infrastructure.so.jar"/>
```

- (Advanced) Use the `DownloadService` in the JNLP API to check the cache at runtime for JAR versions.
- If no main attribute is specified, the JAR holding the `main` class must be listed first.

JNLP File - System Properties

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

Specify any number of system property name-value pairs.

JNLP File - Security Element

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.ja
        <property name="key" valu  value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo />
</jnlp>
```

Request full access to client system.

Requires all JAR files to be digitally signed.

Requires user's permission.

JNLP File - Main Class Element

```
<?xml version='1.0' encoding='UTF-8'?>
<jnlp
    codebase="http://www.bradshuler.com/jws"
    href="example.jnlp">
    <information>
        <title>FileChooserDemo</title>
        <vendor>Object Computing, Inc.</vendor>
        <homepage href="index.html"/>
        <description>JFC FileChooserDemo App</description>
        <icon href="oci_logo.gif"/>
        <offline-allowed/>
    </information>
    <resources>
        <j2se version="1.3"/>
        <jar href="FileChooserDemo.jar"/>
        <property name="key" value="value1"/>
    </resources>
    <security>
        <all-permissions/>
    </security>
    <application-desc main-class="FileChooserDemo"/>
</jnlp>
```

JNLP File - More on main Class

```
<!-- Pass in some arguments -->
<application-desc main-class="edu.purdue.ie.MightyCAD">
    <argument>-jconsole disable</argument>
    <argument>Courier New</argument>
</application-desc>
```

- The `application-desc` element is optional. If not present, the first JAR file listed in the `resources` element must contain a manifest file pointing to the main class.
- A similar element for Applets, `applet-desc` allows Java Web Start to launch applets using the built in AppletViewer.

Java Web Start - Security

- Specify the `all-permissions` security element to:
  - **Access the Local File System**
  - **Access Printer(s)**
  - **Read/Write to Shared System-wide Clipboard**
  - **Access the Local Network**
  - **Read System Properties**
  - **Install a custom SecurityManager**
  - **Retrieve JARS from anywhere**
  - **Use Native Libraries**
- Support for fine-grained permissions is mentioned in JNLP specification, but not yet implemented (Java Bug Database #4398087).

Java Web Start - Security (cont.)

- Requirements for full access:
  - **All JAR files must be digitally signed**
    - **Assures user no one has tampered with the JAR**
    - **Uses public key encryption (public/private keys)**
- JAR Signing Requires:
  - **Java 2 SDK `jarsigner` tool**
  - **A Certificate**
    - **Assures the user public/private keys in JAR are yours.**
    - **Should come from a *Certifying Authority* (i.e.. VeriSign)**
    - **Does <u>not</u> mean the user should trust the application**
      (Yes, Brad Shuler signed the JAR.
      But... Do I trust Brad Shuler?)

Java Web Start - Security (cont.)

- Issues with Certificates
  - **They Cost $$$ (Paid to certifying authority)**
  - **Take time to get (procurement cycle)**
- The Alternative..
  - Create a *Self Signed Test Certificate*
  - Should be used for testing and prototypes only

Java Web Start - Example

- Signing a JAR using a Test Certificate:
  - *Self Signed* (use for testing only!):

Example: Creating a Test Certificate

- Step 1: Creating a keystore
  - Use the Java 2 SDK **`keytool`** program
  - Creates a keystore file on the local machine
  - Holds public and private keys
  - Public key exported as a certificate

Example: Creating a Test Certificate

- Step 1: Creating a keystore

G:\WINNT\System32\cmd.exe

```
G:\>keytool -alias myalias -genkey
Enter keystore password:  foobar
What is your first and last name?
  [Unknown]:  Brad Shuler
What is the name of your organizational unit?
  [Unknown]:  Information Systems
What is the name of your organization?
  [Unknown]:  Object Computing, Inc.
What is the name of your City or Locality?
  [Unknown]:  Saint Louis
What is the name of your State or Province?
  [Unknown]:  Missouri
What is the two-letter country code for this unit?
  [Unknown]:  US
Is <CN=Brad Shuler, OU=Information Systems, O="Object Computing
Louis, ST=Missouri, C=US> correct?
  [no]:  yes

Enter key password for <myalias>
        <RETURN if same as keystore password>:

G:\>
```

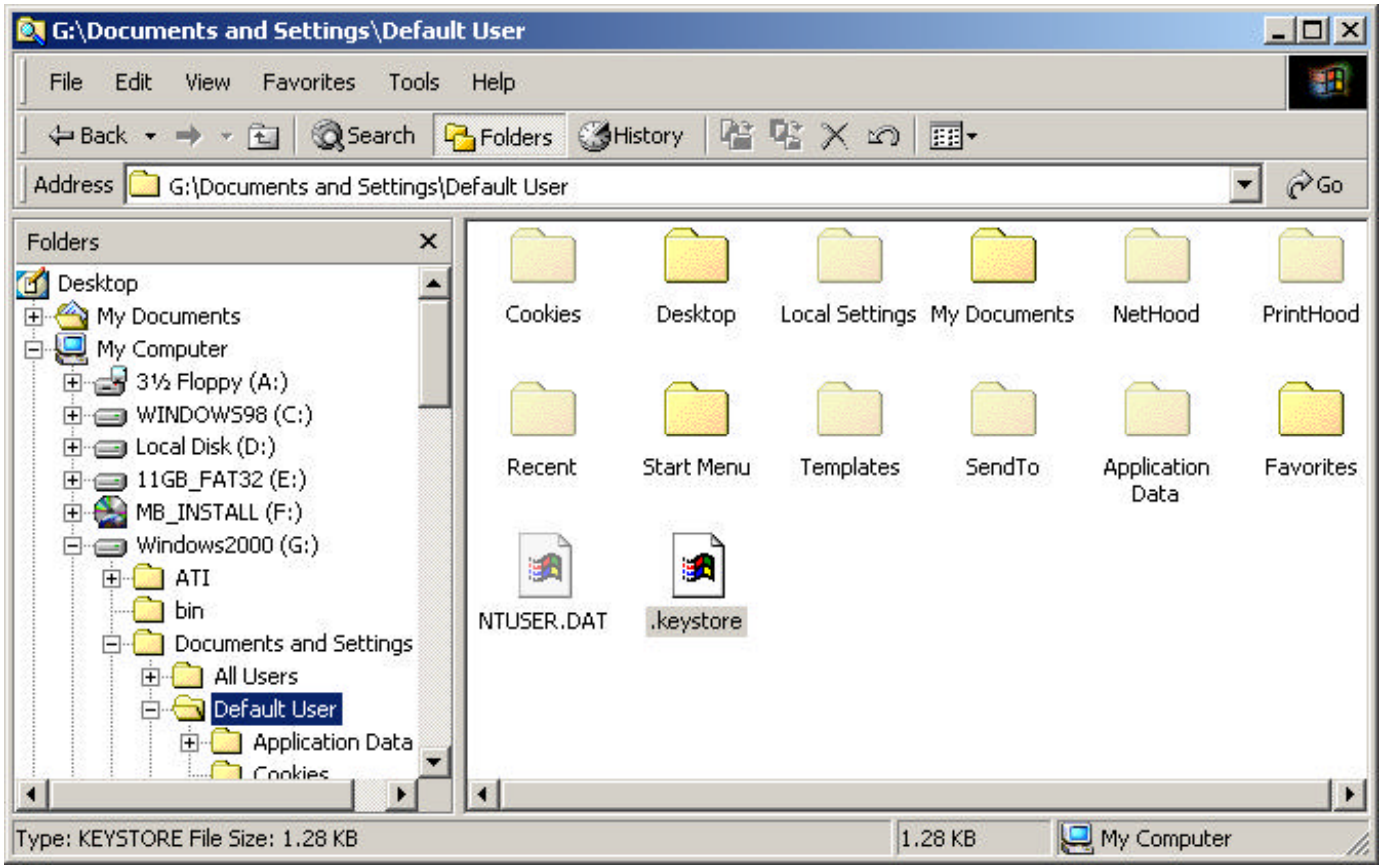
Example: Creating a Test Certificate

- Step 1: Creating a keystore (list contents)

```
G:\WINNT\System32\cmd.exe

G:\>keytool -list
Enter keystore password:  foobar

Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry:

myalias, Wed Nov 28 22:00:08 CST 2001, keyEntry,
Certificate fingerprint (MD5): 76:CE:BF:C2:A0:7B:1B:A1:45:F8:E5:

G:\>
```

Example: Creating a Test Certificate

- Step 1: Creating a keystore (file location)



Note: `.keystore` is the default file name,

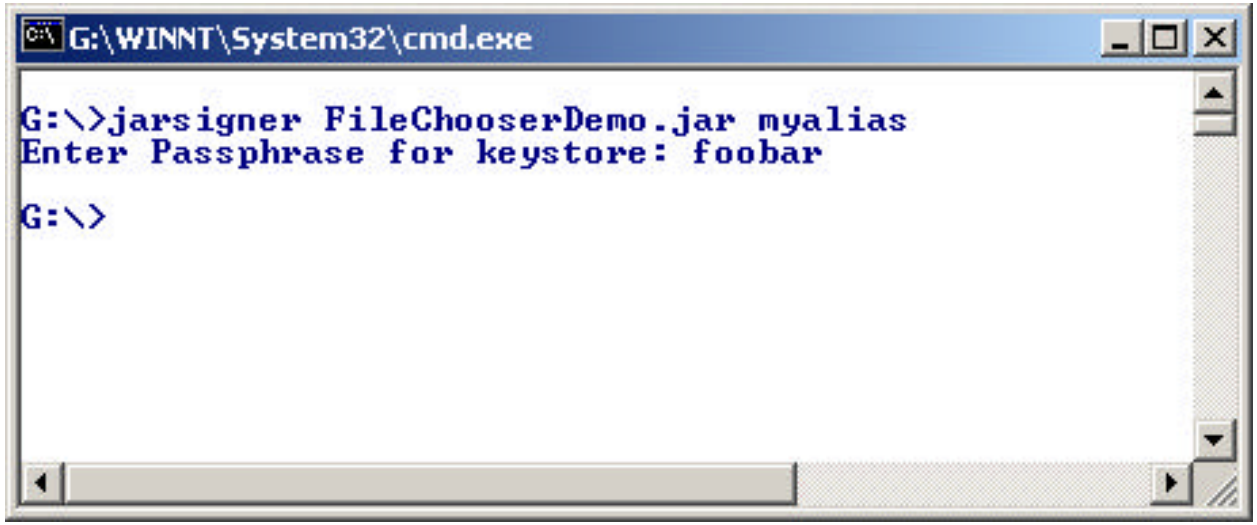
Create a custom name using: `keytool -keystore` *`name`*

Example: Signing a JAR File

- Step 2: Use the Java 2 SDK **`jarsigner`** program
  - Exports certificate from keystore, places in JAR
  - Each file in archive is given a *digest entry* in the manifest.
  - Digest entries are one way hashes -- if file is modified, it's hash value is no longer valid.
  - When JAR is being verified (by Java Web Start), digests are recomputed and compared to values in manifest.

Example: Signing a JAR File

- Step 2: Use the Java 2 SDK **`jarsigner`** program



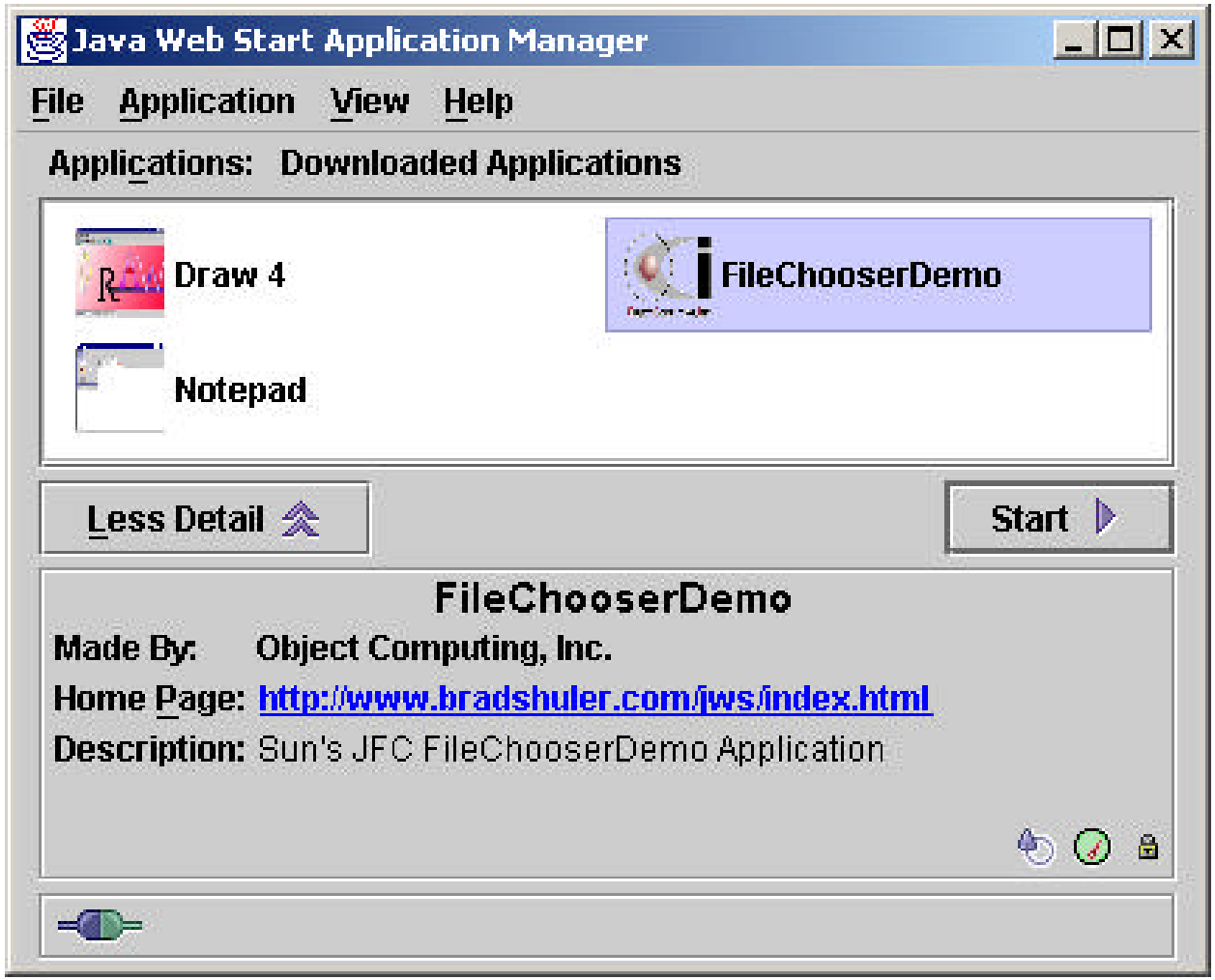
- Finally, move the file to the web server deployment area.

Note: The Ant `SignJar` Built-In Task makes signing JARS seamless..

Java Web Start Application Manager

- Manage the application cache
- Add desktop icons, Start menu entries
- Enable the Java Console
- Enable Logging
- Configure HTTP proxy settings
- View installed JRE's
- View, import, export certificates

Java Web Start Application Manager



Java Web Start: JNLP API

- **`BasicService`** (query environment)
- **`ClipboardService`** (access clipboard data)
- **`DownloadService`** (control how cached)
- **`FileOpenService`** (see local disk)
- **`FileSaveService`** (write to local disk)
- **`PrintService`**
- **`PersistentService`** (similar to cookies)

Summary

- Java Web Start:
  - Provides the "plumbing" to allow client machines to download a centralized Java 2 application over a network and run on their machine -- all with one click.
  - Always guarantees the user is running the latest version.
  - Is secure (Java 2 Security Model)

References

- Sun Java Web Start Home Page - http://java.sun.com/products/javawebstart/index.html
- Sun Java Developer Connection JWS/JNLP Forum: http://forum.java.sun.com/forum.jsp?forum=38
- Java Community Process (JNLP Specification) - http://jcp.org/aboutJava/communityprocess/final/jsr056/index.html
- JavaWorld - http://www.javaworld.com/javaworld/jw-07-2001/jw-0706-webstart.html
- IBM - http://www.ibm.com/developerworks/library/j-webstart/index.html
- Unofficial Java Web Start FAQ - http://www.geocities.com/vamp201/jwsfaq.html