

Einführung in JNLP und Java Web Start

Reiner Eberhard, Thomas Forster

1

JNLP und Java Web Start

Arten der Softwareverteilung:

- Windows .exe Dateien
- Java Applikationen
- HTML Anwendungen (Servlets / JSP)
- Java Applets



2

- **Derzeitige Problemfelder:**
 - Windows .exe Dateien

- **Derzeitige Problemfelder:**
 - bei Java Applikationen

- **Derzeitige Problemfelder:**
 - bei HTML Anwendungen (Servlets / JSP)

- **Derzeitige Problemfelder:**
 - bei Java Applets

Java Network Launching Protocol (JNLP):

Definiert plattformunabhängiges Protokoll für die

- **Verteilung**
- **Installation**
- **Starten**
- **Updaten**

von Java Programmen über das Netzwerk

Referenzimplementierung zu JNLP von Sun: Java Web Start



• **Systemanforderungen von Java Web Start:**

- Windows 95/98/NT/2000/ME/XP JRE 1.2.2 oder höher
- Solaris 2.6, 7, 8 JRE 1.2.2 oder höher
- Linux JRE 1.2.2 oder höher

• **Installationsvarianten:**

- Web Start Paket
- Im Java Standard Development Kit 1.4 enthalten
- Bei Macintosh OS X 10.1 integriert

JNLP und Java Web Start



Kernkonzepte:

- Webzentriertes Anwendungsmodell
- Bereitstellungsprotokoll
- Ausführungsumgebung

9

JNLP und Java Web Start



• Webzentriertes Anwendungsmodell:

- keine Installationsphase
- Downloading und Caching von Applikationen
- Idee: "Quasi" Erweiterung des Classpath um URL-Angabe:

```
java -classpath http://www.oio.de/app/MyApp.jar de.oio.Main
```
- Unterstützung transparenter (Versionscheck) und inkrementeller (nur Änderungen) Updates.

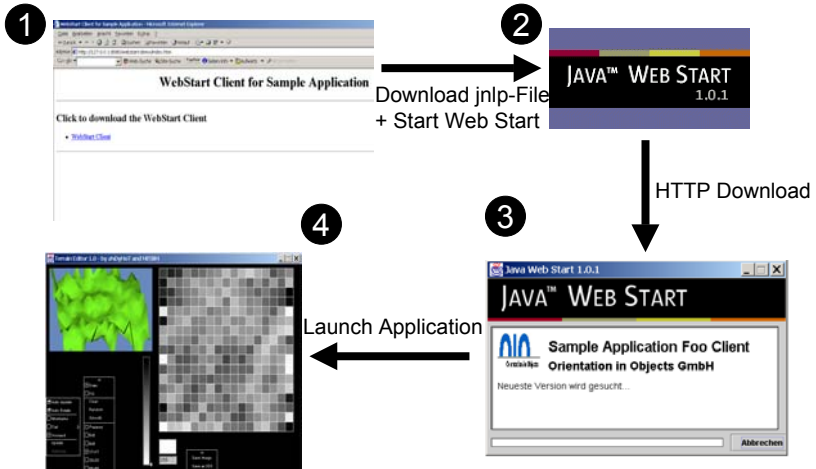
10

- **Bereitstellungsprotokoll:**
 - standardisiertes Protokoll
 - Beschreibt wie eine Applikation für JNLP Clients auf einem Web Server hinterlegt werden muß

- **Ausführungsumgebung:**
 - HTTP Proxy Konfiguration für Anwendung
 - Standardumgebung für das Ausführen von Applikationen
 - **beschränkte Umgebung** für unbekannte (untrusted) Applikationen
 - **uneingeschränkte Umgebung** für bekannte (trusted) Applikationen

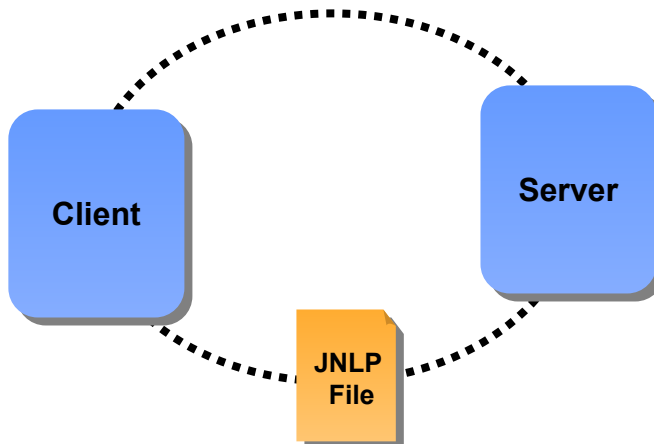
JNLP und Java Web Start

Anwendungsszenario aus Client Sicht:



JNLP und Java Web Start

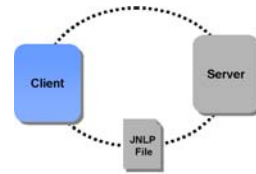
JNLP Instanzen:



JNLP und Java Web Start



Orientation in Objects



Client:

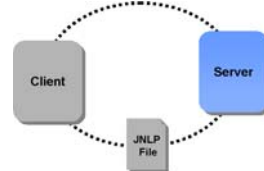
- JNLP Implementierung
- JRE 1.2.2 oder höher
- Web-Browser

15

JNLP und Java Web Start



Orientation in Objects



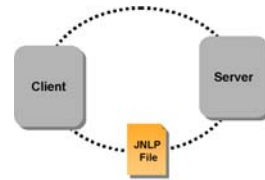
Server:

- Ressourcen
- HTTP Server
- .jnlp Deskriptor File

- Servlet Container

16

JNLP und Java Web Start

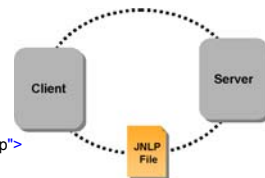


- **JNLP-File:**

- Informationen für JNLP Client in XML um Download und Aufruf einer Java Anwendung vom Server.
- Unterteilung in funktionale Kategorien:
 - **Applikationsdeskriptor**
 - **Applet Deskriptor**
 - **Komponenten Deskriptor**
 - **Installer Deskriptor**
- **mit jeweils**
 - Metainformationen
 - Sicherheit
 - Ressourcen

17

JNLP und Java Web Start



- **Beispiel eines JNLP-Files:**

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" codebase="http://localhost:8080/demounsigniert" href="ter_web.jnlp">
  <information>
    <title>Java Terrain Editor 1.0</title>
    <vendor>AndyHot</vendor>
    <homepage href="index.html"/>
    <description>Terrain Editor 1.0</description>
    <description kind="short">Java Terrain Editor 1.0</description>
    <offline-allowed/>
  </information>

  <security/>

  <resources>
    <j2se version="1.3"/>
    <jar href="ter_web.jar"/>
  </resources>
  <application-desc main-class="andyhot.goal3d.TerrainEditor"/>
</jnlp>
```

18

- **JNLP Download Protokolle:**
 - Basic Download Protokoll
 - Versionsbasiertes Download Protokoll
 - Erweitertes Download Protokoll



- **Basic Download Protokoll:**
 - Download von Ressourcen über HTTP `GET`-Request

Bsp.: `http://www.oio.de/foo.jar`

- **Versionsbasierter Download:**

- Setzt auf Basic Download Protokoll auf mit zusätzlichem `version-id` Feld.

Bsp.: `http://www.oio.de/foo.jar?version-id=1.7%2B`

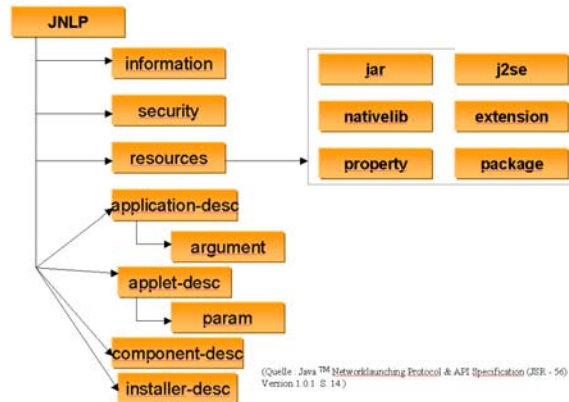
- **Erweitertes Download Protokoll:**

- Erweiterung der vorhergehenden Protokolle um plattformspezifische Informationen
- Erweiterungen in Form von URLs oder URL / Version-ID Paaren

Bsp.: Download von Java Runtime Environements:

```
http://www.oio.de/jreinstaller.exe?arch=x86&os=Windows  
+95&locale=en_US&version-id=1.3&known-plattformen=1.2
```

- JNLP-File Struktur:

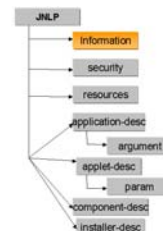


23

- Details zum Information Element:

- Meta-Informationen zur Applikation:

- wird vom JNLP Client verwendet, um die Anwendung in den Desktop zu integrieren, und Feedback Möglichkeiten für den Anwender zur Verfügung zu stellen u.a. .
- z.B. <vendor>-Element, <title>-Element, <offline-allowed>-Element

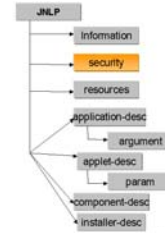


24

JNLP und Java Web Start



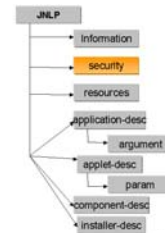
- **Details zum Security Element:**
 - Security basiert auf den Sicherheitsrichtlinien der Java 2 SE Standard Umgebung.
 - Bereiche:
 - **Benutzer**
 - **Applikation**
 - **Signatur der Ressourcen**



JNLP und Java Web Start



- **Rechtevergabe vom Benutzer:**
 - keine (default)
 - Untrusted Environment
 - Rechtevergabe auf Anfrage der Applikation
 - Benutzerdialog



→ Trusted Environment

- Rechte die Applikation verlangt:

- <Security> Element

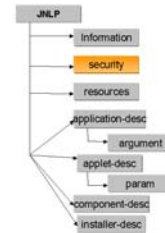
- no-permissions (default)

- <all-permissions>

- Anfrage nach Trusted Environment

- <j2ee-application-client-permissions>

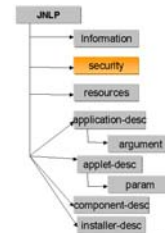
- Anfrage nach Trusted Environment



- Signatur der Ressourcen:

- Anwendungen mit unsignierten Ressourcen die Rechte anfordern werden komplett abgewiesen.

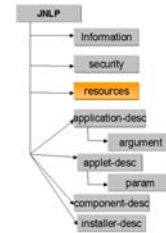
- Anwendungen mit signierten Ressourcen die Rechte anfordern lösen Benutzerabfrage aus.



- **Details zum Ressourcen Element:**

- Angabe aller Ressourcen, die Teil der Anwendung sind

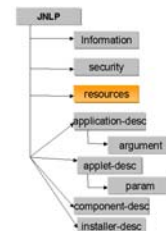
- `<resources os="SunOS">...</resources>`
- Class Files, native Libraries, jar Files, Systemproperties



- **Details zum Ressourcen Element:**

- Unterelemente :

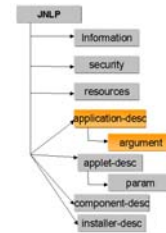
- `<property name="key1" value="value1"/>`
- `<jar href=" .. .jar" main="true|false"/>`
- `<nativelib href=".. .jar"/>`
- `<j2se version="x.x">`



- **Details zum Application Element:**

- enthält die Informationen, die zum Starten einer Applikation (mit Main-Methode) benötigt werden :

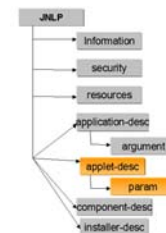
- `<application-desc main-class="de.oio.Main">`
- optional können Argumente mitgegeben werden die im Parameterarray `args[]` der Main-Methode zur Verfügung stehen



- **Details zum Applet Element:**

- enthält die Informationen, die zum Starten eines Applets benötigt werden :

- `<applet-desc main-class="de.oio.TestApplet" documentbase="index.html" name="Test" width="500" height="300"/>`
- optional können auch hier Argumente mitgegeben werden, die über die `Applet.getParameter("key")`-Methode gelesen werden können.



Web Start Applikationen auf Serverseite:

- Die einfache Variante
- Aufbau eines Web Archiv
- Automatisierung mit Ant Tasks
- Einsatz von Tools



33

- die einfache Variante :

HTTP - Server:

Verzeichnis : demo



server.xml :

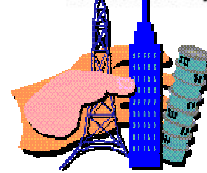
```
<Context path="/mydemo"  
docBase="demo" />
```

web.xml :

```
<mime-mapping>  
<extension>jnlp</extension>  
<mime-type>application/x-java-jnlp-file</mime-type>  
</mime-mapping>
```

34

JNLP und Java Web Start



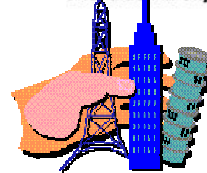
- **Aufbau eines Web Archiv:**

- Web Archive Struktur:

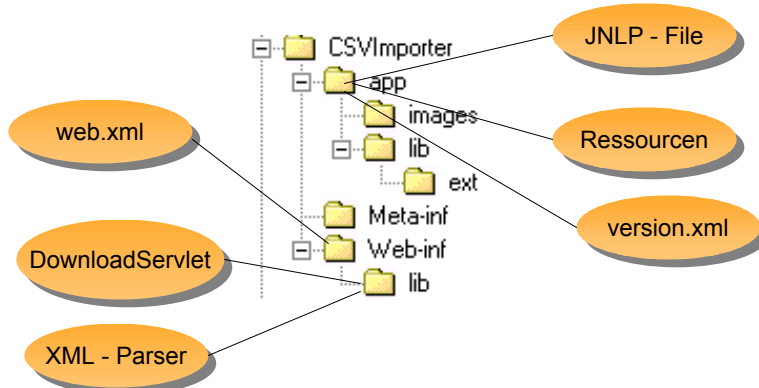


35

JNLP und Java Web Start



- **Web Archive Struktur im Detail:**



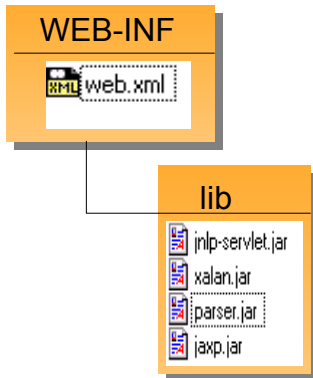
36

JNLP und Java Web Start



Orientation in Objects

- Web Archive Struktur im Detail:



```
<web-app>
<!-- JNLP-Download-Servlet -->
<servlet>
<servlet-name>JnlpDownloadServlet</servlet-name>
<servlet-class>com.sun.javaws.servlet.JnlpDownloadServlet</servlet-class>
<init-param>
<param-name>
logLevel
</param-name>
<param-value>!
DEBUG
</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>JnlpDownloadServlet</servlet-name>
<url-pattern>/app/*</url-pattern>
</servlet-mapping>
</web-app>
```

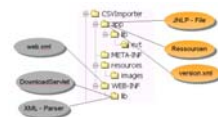
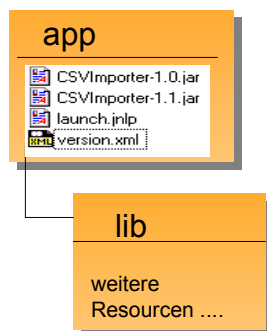
37

JNLP und Java Web Start



Orientation in Objects

- Web Archive Struktur im Detail:

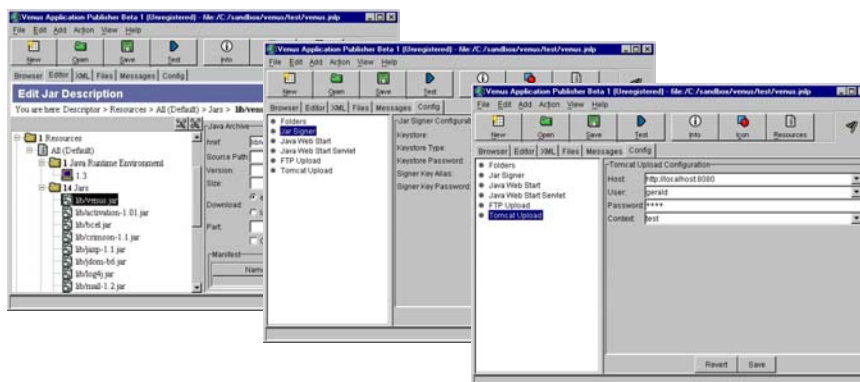


38



- **Automatisierung mit Ant-Tasks:**
 - Benutzung des Ant-Targets “WAR“
 - Vamp’s Ant Task Suite (<http://geocities.com/vamp201/ant.html>)

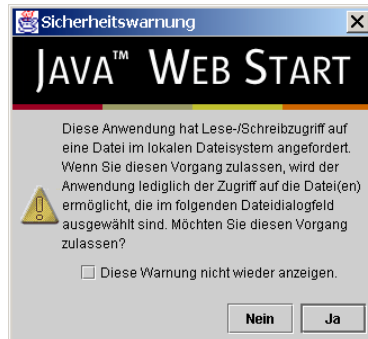
- **Einsatz von Tools (z.B. Vamp):**



(Quelle: <http://geocities.com/vamp201/>)

Die JNLP API:

- Erlaubt das starten bestimmter Services aus einer Untrusted Environment mit Benutzerabfrage:



41

• Die JNLP API:

- `javax.jnlp.BasicService`
- `javax.jnlp.DownloadService`
- `javax.jnlp.ExtensionInstallerService`

• Weitere optionale Services:

- `javax.jnlp.FileOpenService`
- `javax.jnlp.FileSaveService`
- `javax.jnlp.ClipboardService`
- `javax.jnlp.PrintService`
- `javax.jnlp.PersistenceService`

42

- **Beispielcode mit FileOpenService:**

```
import javax.jnlp.*;
...
FileOpenService fos;

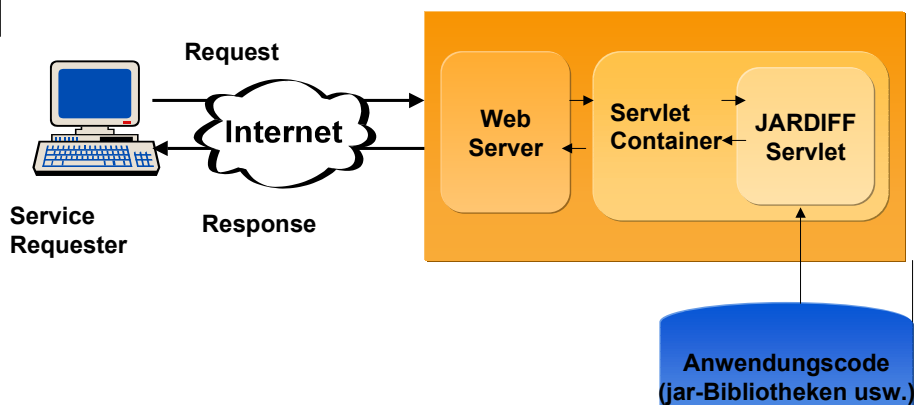
try {

fos=(FileOpenService)ServiceManager.lookup("javax.jnlp.FileOpenService");
} catch (UnavailableServiceException e)
{ fos = null; }

if (fos != null) {
try {
// ask user to select a file through this service
FileContents fc = fos.openFileDialog(null, null);
// as user to select multiple files through this service
FileContents [] fcs = fos.openMultiFileDialog(null, null);
} catch (Exception e)
{ e.printStackTrace(); }
}
```

43

- **Architektur und Ablauf bei einem Update**



44

Transparenter und inkrementeller Update:

- WebStart bietet für inkrementellen Update vom Server ein Download Servlet:
`com.sun.javaws.servlet.JnlpDownloadServlet`
- `version-id` (verlangte Version) und `current-version-id` (lokale Version) Feld im HTTP Request
- JARDIFF-File (`META-INF/INDEX.JD`) enthält Anweisungen und notwendige Daten um aus *alt.jar* -> *neu.jar* beim Client herzustellen.
- Versionsangaben bei `<jar href>` Tags des `jnlp`-Files.

version.xml und JNLP – Datei :

```
versions.xml
<jnlp-versions>
  <resource>
    <pattern>
      <name>CSVImporter.jar</name>
      <version-id>1.0</version-id>
    </pattern>
    <file>CSVImporter-1.0.jar</file>
  </resource>
</jnlp-versions>
```

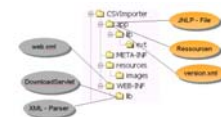
launch.jnlp

```
<jar href="CSVImporter.jar" version="1.0"/>
```

von 1.0 nach 1.1

```
<jnlp-versions>
  <resource>
    <pattern>
      <name>CSVImporter.jar</name>
      <version-id>1.0</version-id>
    </pattern>
    <file>CSVImporter-1.0.jar</file>
  </resource>
  <resource>
    <pattern>
      <name>CSVImporter.jar</name>
      <version-id>1.1</version-id>
    </pattern>
    <file>CSVImporter-1.1.jar</file>
  </resource>
</jnlp-versions>
```

```
<jar href="CSVImporter.jar" version="1.1"/>
```



- **JARDIFF-file:**
 - *.jardiff* Datei
 - enthält:
 - **alle geänderten Ressourcen**
 - **Index-Datei META-INF/INDEX.JD mit Änderungsanweisungen:**
 - remove <entry>
 - move <from> <to>

Schwierigkeiten von Java Web Start in der Praxis:

- Probleme beim Finden von JRE Versionen.
- Weitere Bugs siehe:
 - <http://developer.java.sun.com/developer/bugParade/>



JNLP und Java Web Start



Fazit:

- Mit dem Java Network Launch Protokoll (JNLP) existiert eine durchdachte Spezifikation für einen simplen Umgang mit Java Clients in Netzwerken.
- Unterstützung für Verteilung, Installation, Starten, Updaten.
- Mit Java Web Start von Sun existiert eine voll praxistaugliche JNLP Implementierung.

49

JNLP und Java Web Start



• Weitere Implementierungen und Tools:

- OpenJNLP
 - Open Source JNLP Implementierung
 - <http://openjnlp.nanode.org>
- NetX
 - Open Source JNLP Implementierung des Object Component Desktop (OCD)
 - <http://www.acm.vt.edu/~jmaxwell/netx/netx.html>



50

- **Weitere Tools zu JNLP:**

vamphq.com

- Venus Application Publisher (Vamp)
 - Grafisches Tool zum Packen, Signieren und Bereitstellen von JNLP Anwendungen.
 - <http://www.geocities.com/vamp201/>
- Rachel
 - Ressource Loading for Java Web Start
 - <http://www.geocities.com/vamp201/rachel.html>
- Clio / Jess
 - CD Application Installer für JNLP
 - <http://www.geocities.com/vamp201/clio.html>

51

- **Quellenangaben und Links:**

- Java Network Launching Protocol & API Spezifikation (JSR-56)
- <http://java.sun.com/products/javawebstart>
- <http://openjnlp.nanode.org>
- www.webstartapps.com
- <http://java.sun.com/products/javawebstart/partners.html>
- <http://www.up2go.net/updown/index.jsp>



52

Weiter Fragen:

OIO Stand auf der JAX