

Java Web Start is een helper applicatie waarmee je Java applicaties via een hyperlink op een webpagina naar een client machine kunt downloaden en opstarten. Java Web Start is gebaseerd op het Java Network Launching Protocol & API (JNLP) en is daarvan de reference implementatie. De onderliggende technologie heeft overeenkomsten met applets, maar de kennis erover is veel minder wijd verspreid. Het is daarom nuttig deze technologie eens wat meer in de schijnwerpers te zetten.

Java Web Start

Technologie voor ontwerpen van webclients

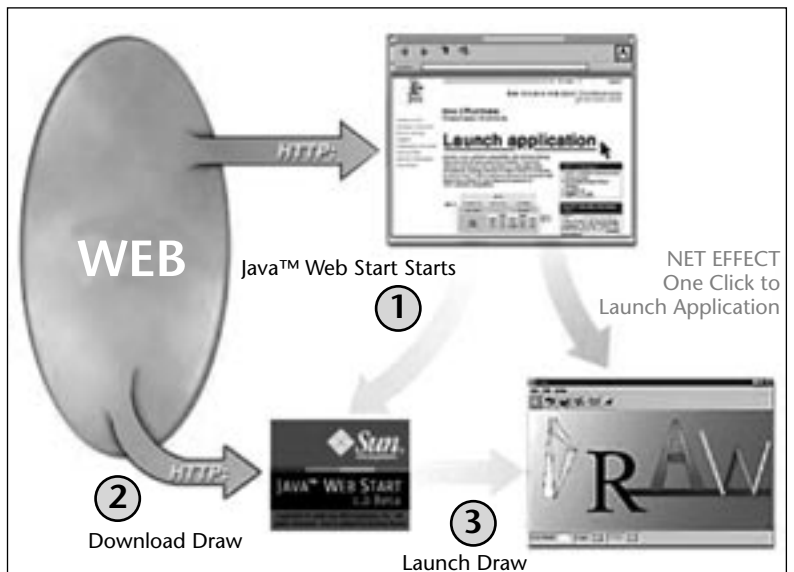
De Java Web Start-applicatie wordt op de client machine in een cache opgeslagen. Bij een hernieuwde start wordt de applicatie uit de cache geladen. Java Web Start applicaties zijn gewone Java applicaties en hoeven niet aan bijzondere voorwaarden te voldoen. Wel draaien ze op de client machine in een "sandbox" met beperkte rechten. Meer rechten worden verkregen door de code te voorzien van een digitale handtekening.

DESKTOP-APPLICATIONS Java Web Start is ontworpen met als achterliggende gedachte dat de meeste mensen voor veel activiteiten de voorkeur geven aan desktop-applicaties boven web-applicaties. Een gemiddelde gebruiker zal voor het verwerken van bijvoorbeeld e-mail kiezen voor een desktop-applicatie interface in plaats van een webinterface in een browser. De desktop-applicatie interface heeft een aantal aantrekkelijke voordelen. Zo is het user interface rijk geschakeerd, reageert deze snel en hangt in ieder geval niet af van de connectie snelheid. Desktop-applicaties kunnen verder gemakkelijk worden opgestart via een startmenu of een desktop-shortcut en vereisen niet dat de browser eerst wordt opgestart. Tenslotte werken desktop-applicaties ook offline zoals bijvoorbeeld in een vliegtuig.

PROBLEMEN MET WEBAPPLICATIONS Een nadeel van desktop-applicaties is echter dat installatie, onderhoud en beheer vaak problematisch zijn. Voor een bedrijf met honderden client computers kost het decentrale onderhoud en beheer van desktop applicaties niet alleen heel wat hoofdbreken, ook vormt het een flinke kostenpost. Veel bedrijven kiezen daarom, indien mogelijk, voor webapplicaties waarvan installatie en beheer simpel en kosteneffectief zijn. Dit kan immers centraal worden geregeld. Maar de webapplicaties hebben

natuurlijk ook hun beperkingen, zoals een problematische toegang tot resources op de client computer, een minder rijk geschakeerde user interface en de noodzaak van een Internet-connectie.

APPLETS IN WEBAPPLICATIONS Een mogelijke oplossing voor de beperkingen van webapplicaties zijn Java applets die aan de basis stonden van de Java revolutie en nog steeds van zich doen spreken. Aan applets wordt gerefereerd vanuit een HTML pagina en ze worden van een webserver gedownload als de pagina wordt opgevraagd. Hierdoor is centraal beheer van de applicatie mogelijk. Ze draaien binnen de context van een webbrowser in een veilige "sandbox"-omgeving en hebben geen rechten op de client machine. Applets hebben tevens een rijk geschakeerde user interface. Een pro-



FIGUUR 1. Schematisch overzicht Java Web Start

bleem met applets was in het verleden dat ze bij iedere invocatie van de webpagina opnieuw naar de client machine moesten worden overgebracht. Inmiddels zijn er in de nieuwste Java-versies net als bij Java Web Start

Java Web Start combineert het centrale beheer van webapplicaties met de kracht van desktopapplicaties

caching mogelijkheden. Ook kunnen applets tegenwoordig rechten krijgen, om resources op de client machine te benaderen. Desalniettemin draaien applets nog steeds alleen binnen de browser context.

JAVA WEB START Java Web Start biedt een combinatie van deze twee werelden: het kosteneffectieve, centrale beheer van webapplicaties en de krachtige mogelijkheden van een desktop-applicatie. Met Java Web Start kan een gebruiker door activering van hyperlink in een webpagina, een nieuwe Java-applicatie op zijn client computer installeren. De link moet refereren aan een JNLP descriptor bestand met de extensie `.jnlp`, waarin de configuratie voor de applicatie staat beschreven. Achter de schermen wordt Java Web Start door de browser geactiveerd, presenteert een Splash screen en controleert of de resources voor de nieuwste versie van applicatie mogelijk al in de cache aanwezig zijn. Zo ja, en indien er geen update voor de applicatie aanwezig is, dan wordt de applicatie direct opgestart. Zo niet, dan zal Java Web Start de resources of de nieuwste versie eerst downloaden. De enige voorwaarde om dit op de client machine mogelijk te maken, is dat de Java Web Start software op de client machine staat geïnstalleerd. De applicatie draait vervolgens in een eigen Java Runtime Environment (JRE), onafhankelijk van de browser.



FIGUUR 2. Waarschuwing van Java Web Start bij het bewaren van een bestand

BEVEILIGING Applicaties die door Java Web Start worden opgestart draaien standaard in de zogeheten “sandbox”, met beperkte toegang tot bestanden en het netwerk. Het starten van applicaties via Java Web Start is daarom veilig en brengt de integriteit van het systeem niet in gevaar. Toch kun je zonder speciale voorzieningen vanuit een Java Web Start applicatie nog wel het nodige doen. Zo kun je geselecteerde bestanden opslaan op de client computer en deze ook weer inlezen. Als dit wordt geprobeerd verschijnt er een “Security Advisor” dialoog waarin de gebruiker met deze toegang kan instemmen. Standaard is ook een netwerkconnectie mogelijk met de server waarvan de applicatie afkomstig is. En ook zijn een aantal API functies van het Java Network Launching Protocol (JNLP) beschikbaar.

Een extra veiligheidskenmerk dat door Java Web Start wordt ondersteund is de digitale handtekening. Als een applicatie wordt aangeleverd in de vorm van een aantal JAR's (Java Archives) die voorzien zijn van een digitale handtekening, kan Java Web Start nagaan of de inhoud van de JAR's is gewijzigd na het aanbrenge van deze handtekening. Als de verificatie van de digitale handtekening mislukt, zal Java Web Start de applicatie niet opstarten omdat er mogelijk mee is geknoeid.

Applicaties die om onbeperkte toegang tot het systeem vragen, moeten voorzien zijn van een digitale handtekening. In dat geval toont Java Web Start bij de eerste aanroep een “Security Warning” dialoog, met informatie over de maker van de applicatie. In het geval je ervoor kiest de maker te vertrouwen, wordt de applicatie opgestart. Bij volgende aanroepen blijft de dialoog achterwege en start de applicatie direct op.

HANDTEKENING ZETTEN De stappen die moeten worden ondernomen om een JAR te voorzien van een digitale handtekening zijn:

- 1) Creëer met het keytool een nieuwe key in een nieuwe keystore:

```
keytool -genkey -keystore myKeys -alias java-magazine
```

Vervolgens wordt gevraagd om informatie over de nieuwe key op te geven zoals password, name, provincie enzovoort.

- 2) Breng de digitale handtekening met jarsigner aan op de JAR :

```
jarsigner -keystore myKeys hello.jar javamagazine
```

Een zelf gecreëerde digitale handtekening moet alleen gebruikt worden voor interne testdoeleinden. Het geeft geen garanties omtrent de identiteit van de gebruiker en kan daarom niet worden vertrouwd. Een echt betrouwbare handtekening kan verkregen worden via een certificate authority en moet gebruikt worden als de applicatie gaat communiceren met de boze buitenwereld.

OPZET VAN SITE Java Web Start maakt gebruik van bestaande Internettechnologie, zoals het HTTP protocol en webservers. De bestanden die nodig zijn om een Java applicatie via Java Web Start op client machines te installeren, moeten in een directory van de webserver staan en toegankelijk zijn. Dat verschilt niet wezenlijk van het opzetten van een site met HTML bestanden. Het komt typisch neer op het kopiëren van de benodigde JAR's en het JNLP bestand naar een directory op de webserver. Het enige opmerkelijke is, dat op de webserver nog een speciaal MIME type moet worden aangeemaakt.

INSTELLINGEN WEBSERVER Applicaties kunnen vanaf iedere standaard webserver worden geïnstalleerd, mits de webserver een speciaal MIME type kent. De webserver moet bij alle aanvragen voor bestanden met een .jnlp extensie het `application/x-java-jnlp-file` MIME type retourneren. Webrowsers gebruiken het MIME type dat samen met de inhoud door de webserver wordt geretourneerd, om te bepalen hoe een bepaald type inhoud moet worden afgehandeld. Bij het genoemde MIME type zal de webbrowser Java Web Start activeren. De wijze waarop een MIME type aan een webserver wordt toegevoegd verschilt per webserver.

JNLP BESTAND Het Java Network Launch Protocol (JNLP) verwacht behalve een JAR bestand voor de classes van de applicatie ook een descriptor waarin staat beschreven hoe de applicatie moet worden opgestart. Deze descriptor is een XML bestand met extensie .jnlp. Er staat onder meer in welke resources de applicatie nodig heeft, en welke informatie bij het starten in een pop-up verschijnt. Andere onderdelen van dit bestand

zijn optioneel, en zijn alleen noodzakelijk, wanneer de applicatie toegang nodig heeft buiten de "sandbox". Het JNLP-bestand begint met de `<jnlp>` tag waarvan attributen aangeven welke JNLP versie wordt gebruikt en waar het JAR-bestand van de applicatie zich bevindt:

```
<jnlp spec="1.0+"
  codebase="http://localhost/
  JavaWebStartDemo/"
  href="hello.jnlp">
```

De `<information>` tag beschrijft de inhoud van het pop-up venster dat verschijnt als de applicatie geladen wordt:

```
<information>
  <title>Hello World</title>
  <vendor>Java Magazine</vendor>
  <homepage href="/JavaWebStartDemo" />
  <description>Demonstration Java Web Start</
  description>
</information>
```

De optionele `<offline-allowed>` tag specificeert dat de applicatie ook zonder netwerkconnectie gedraaid mag worden:

```
<offline-allowed/>
```

De `<security>` tag is alleen nodig, wanneer er extra permissies vereist zijn. Terwijl het lezen en schrijven van bestanden ook mogelijk is zonder gesigneerde code (deze genereert alleen een Security Advisor dialoog), vereist netwerktoegang buiten de server waar de applicatie van afkomstig is, dat de code is gesigneerd. Het `<security>` element kent twee opties `<all-permissions/>` of `<j2ee-application-client-permissions/>`. J2EE permissies zijn meestal voldoende, omdat ze socket permissies, clipboard toegang permissie en printing permissie omvatten:



FIGUUR 3. Het toevoegen van een nieuw MIME type in Internet Information Server



FIGUUR 4. Security Warning bij het voor het eerst downloaden van een applicatie

```
<security>
  <j2ee-application-client-permissions/>
</security>
```

Het <resources> element maakt het mogelijk de benodigde Java Runtime versie te specificeren en de JAR-bestanden aan te geven die de applicatie classes bevatten. Het is mogelijk een geprioritiseerde lijst en specifieke versies van JRE-versies aan te geven. De locatie van het JAR bestand werd al eerder aangegeven in de <codebase> tag:

```
<resources>
  <j2se version="1.2+" />
  <jar href="/JavaWebStartDemo/hello.jar"/>
</resources>
```

Het <application-desc> element specificeert de main-class van de applicatie, maar als er een main-class is gespecificeerd in het manifest voor het JAR bestand, is het niet nodig deze aan te geven in het <application-desc> element. Hierna kan de <jnlp> tag worden gesloten.

```
<application-desc main-class="HelloWorld" />
</jnlp>
```

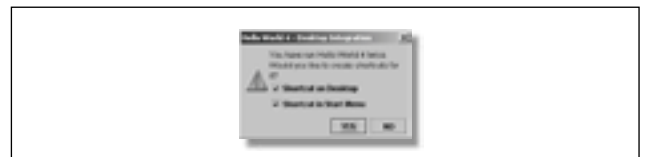
STARTPAGINA Tenslotte wordt het startbestand voor de applicatie gevormd door een eenvoudige HTML-pagina met daarin een hyperlink naar het .jnlp bestand:

```
<html>
<head><title>helloworld launcher</title></head>
<body>
<hr>
<a href="hello.jnlp">Launch Hello World</a>
</body>
</html>
```

Bij de eerste aanroep zal de applicatie worden gedownload. Bij volgende aanroepen benadert Java Web Start eerst de server om te kijken of er een nieuwere versie is. Nadat de applicatie is gedownload, kun je ook aangeven, dat je hem wilt kunnen opstarten via een shortcut buiten de browser om.

```
<?xml version="1.0" encoding="UTF-8"?>
<jnlp spec="1.0+"
  codebase="http://localhost/
JavaWebStartDemo/"
  href="hello.jnlp">
<information>
  <title>HelloWorld</title>
  <vendor>JavaMagazine</vendor>
  <homepage href="/JavaWebStartDemo" />
  <description>Demonstration Java Web Start </description>
</information>
<offline-allowed/>
<security>
  <j2ee-application-client-permissions/>
</security>
<resources>
  <j2se version="1.2+" />
  <jar href="/JavaWebStartDemo/hello.jar"/>
</resources>
<application-desc main-class="HelloWorld" />
</jnlp>
```

FIGUUR 5. Voorbeeld JNLP bestand



FIGUUR 6. Dialoog om shortcuts aan te maken

CONCLUSIE Java Web Start is een interessante technologie voor het ontwerpen van webclients met een rijk geschakeerde user interface. De gebruikersinteractie wordt niet beperkt tot de browser, zoals het geval is bij andere webclient-technologieën zoals applets, servlets, JSP en meer recent JSF. Momenteel is de kennis over Java Web Start nog niet zo wijd verbreid als over de andere genoemde technologieën. Vanaf JDK 5 vormt Java Web Start geen separate download meer, maar is vast onderdeel van de JDK en de JRE geworden. Door deze standaard beschikbaarheid is het niet onwaarschijnlijk dat de populariteit ervan zal toenemen. Het gemak waarmee rijke clients kunnen worden geïnstalleerd zonder rekening te houden met browser incompatibiliteiten, en de automatische caching, zijn des te meer reden Java Web Start in overweging te nemen.

drs. Willem Koppenol is senior trainer en product specialist software development bij Twice IT Training (e-mail: wkoppenol@twice.nl).