

# High~T

## Installatiehandleiding

Auteur ..... : What If Software Solutions  
Versie ..... : 3.0.0  
Laatst aangepast door .... : What If Software Solutions  
Datum ..... : Mei 2014



## Inhoudsopgave

<b>1. Inleiding</b>	<b>1</b>
1.1. Vereiste software	1
1.2. Doel	2
<b>2. Korte samenvatting van installatiehandelingen</b>	<b>3</b>
2.1. Installatie van een nieuwe High-T omgeving	3
2.2. Update van een bestaande High-T omgeving	4
<b>3. Java-versie</b>	<b>5</b>
<b>4. Deployment</b>	<b>6</b>
4.1. Globaal overzicht van de deploy-structuur	6
4.2. De directory bin	7
4.2.1. Windows-scripts	7
4.2.2. Unix-scripts	8
4.3. De directory client	9
4.3.1. De bin-subdirectory - Windows-scripts	9
4.3.2. De bin-subdirectory - Unix-scripts	9
4.3.3. Specifieke client-zaken per server	9
4.3.4. WebStart-omgevingen	9
4.4. De directory lib	10
4.4.1. De subdirectory lib/opt	10
4.4.2. De subdirectory lib/std	10
4.4.3. De subdirectory lib/thirdparty	10
4.5. De directory proxy	10
4.6. De subdirectory server	10
4.6.1. De subdirectory server/ <i>server_name</i>	10
4.6.2. De subdirectory server/ <i>server_name</i> /bin	10
4.6.3. De subdirectory server/ <i>server_name</i> /conf	10
4.6.4. De subdirectory server/ <i>server_name</i> /lib	10
4.7. De subdirectory Tools	10
4.8. Overige meegeleverde zaken	11
4.8.1. Metachanges	11
4.8.2. Miscellaneous	11
<b>5. De High-T server</b>	<b>12</b>
5.1. De commandoregel voor het opstarten	12
5.2. Het configuratiebestand	13
5.2.1. Eigenschappen: Algemeen, Data Sources en Persistence Manager Factories	13
5.2.2. Eigenschappen: Login-authorisatie	14
5.3. Het opstarten en stoppen van de server	15
5.3.1. De High-T-server in een Windows/NT-omgeving	15
5.3.2. De High-T-server in een Unix-omgeving	16
5.3.3. Opmerking m.b.t. de High-T server in een AS/400-omgeving	16
5.4. De High-T licentie	17
5.4.1. Het bestand Licenses.jar	17
5.4.2. Named en concurrent users	17
5.4.3. Werking van de licentie	18
<b>6. De High-T Client</b>	<b>19</b>
6.1. Het opstarten van de client	19
6.1.1. Beschrijving van het opstartcommando	19
6.1.2. De commandoregel	19
6.2. Het configureren van een Java™ Web Start-omgeving voor de client	20

<b>7. High-T Webservices</b> .....	<b>21</b>
7.1. De configuratiefile .....	21
7.1.1. HighTServlet (nl.wiss.high_t.client.web.HighTClientServlet) .....	21
7.1.2. HighTCaller (nl.wiss.webforms.HighTWebServiceCaller).....	21
7.2. Locatie van de programmatuur .....	22
7.3. XSD's .....	22
7.4. Het opstarten van de webservice.....	22
<b>Appendices</b> .....	<b>23</b>
Appendix A: Voorbeeld van een configuratiebestand .....	23
Appendix B: Een opstartscript voor de High-T server in een Unix-omgeving .....	25
Appendix C: Opstartparameters voor de FAA/Java client in een Unix-omgeving .....	26
Appendix D. Voorbeeldconfiguratie van een High-T webservice “hightweb” .....	27
<b>Versiehistorie</b> .....	<b>31</b>

# 1. Inleiding

## 1.1. Vereiste software

Om High-T operationeel te krijgen dienen de volgende softwareversies te worden gebruikt:

- ✓ Java™ Runtime Environment versie 1.7 of hoger.
- ✓ Oracle™ dbms versie 11 of hoger.
- ✓ Apache Tomcat 6.0 of hoger of een andere J2EE Webcontainer (alleen bij gebruik webservices)

## 1.2. Doel

In dit document wordt de configuratie en de installatie van High-T beschreven. High-T is een client-server applicatie en derhalve wordt zowel de client-side installatie als de server-side installatie beschreven.

Tevens worden er handreikingen gedaan die platform-specifiek zijn. Een installatie op een Windows-omgeving zal er anders uit (kunnen) zien dan een installatie op een Linux-omgeving.

Aangezien High-T een Java™-applicatie is wordt bij het installeren volstaan met het plaatsen van de programma-bestanden op een bepaalde locatie en het aanpassen van de configuratiebestanden aan de doel-omgeving.

Aangegeven wordt hoe de bestanden van de client- en server-software dienen te worden. Aan de orde komen de volgende onderwerpen:

- ✓ Het configuratiebestand t.b.v. de server.
- ✓ Het starten en stoppen van de server.
- ✓ Het opstarten van de client.
- ✓ Het configureren van een Java™ Web Start-omgeving t.b.v. de client.

## 2. Korte samenvatting van installatiehandelingen

### 2.1. Installatie van een nieuwe High-T omgeving

#### Algemeen

- ✓ Zorg ervoor dat de namen in de directory-hiërarchie, waarin de High-T-omgeving komt te staan, geen spaties bevat omdat deze de werking van batch-files ernstig kunnen verstoren! *Waar directorynamen met spaties niet vermeden kunnen worden dienen deze tussen dubbele quotes te worden gezet.*
- ✓ Voor High-T is een JRE<sup>1</sup> versie 1.7 of hoger een vereiste omdat High-T onder Java 1.7 is gecompileerd. In de broncode staan bepaalde karakteristieken die onder eerdere Java-versies nog niet bestonden.
- ✓ Het verdient aanbeveling om verschillende OTAP-omgevingen op verschillende databases of schema's voor de High-T metadata te laten draaien, zodat deze elkaar niet in de weg zitten.

#### Database

- ✓ Gebruik om de database gereed te maken voor gebruik van High-T bij voorkeur een database-dump met actuele metadata. Hoewel we ons uiterste best doen om de metachange-scripts zo gebruikersvriendelijk mogelijk te maken, zijn onvolkomenheden nooit helemaal uit te sluiten.

#### Password system owner

- ✓ In de directory \install\etc is een script **ResetPasswordFAAJ.sql** te vinden. Dit script wijzigt het wachtwoord van de system owner (FAAJ) in "partners". Hiermee is de toegang tot High-T gegarandeerd.  
Het script dient gedraaid te worden door de schema owner van de High-T tabellen.

#### Server

- ✓ Maak in de server-directory een subdirectory aan en kopieer de inhoud van de sample-subdirectory hierheen.
- ✓ Pas eventueel de bin/init.bat (Windows) of de bin/initjvm (Unix/Linux) naar behoeven aan.
- ✓ Pas de conf/server.properties zodanig aan dat de URL (`jdbcDataSource.0.url`) naar de juiste database verwijst.
- ✓ Zorg ervoor dat de gebruikersnamen en wachtwoorden in server.properties in overeenstemming zijn met die van de High-T schema owner in de database.

#### Client

- ✓ Maak in de client-directory een subdirectory aan en kopieer de inhoud van de sample-subdirectory hierheen.
- ✓ Pas eventueel de bin/init.bat (Windows) of de client.jnlp (WebStart) naar behoeven aan. De verwijzingen (`jnlp:codebase` en `jnlp:href`) in de client.jnlp dienen in overeenstemming te zijn met de directory waarin de jnlp-file zich bevindt. Is dit niet het geval, dan start de client niet op.
- ✓ Het is wenselijk dat de subdirectory voor de client dezelfde naam heeft als die voor de server, met het oog op herkenbaarheid.
- ✓ Als er bij het opstarten van de client een connectie-timeout optreedt, of als het minutenlang duurt voordat een client daadwerkelijk opstart, is dit hoogstwaarschijnlijk te wijten aan een firewall-instelling.

<sup>1</sup> Java Runtime Environment

## 2.2. Update van een bestaande High-T omgeving

- ✓ Zorg ervoor dat er geen server draait tijdens het installeren van een High-T update.
- ✓ Voer eventuele metachanges in de database door m.b.v. TOAD of SQL\*PLUS.
- ✓ Zorg altijd dat er een backup van vlak vóór de update aanwezig is voor het geval dat er toch iets misgaat.
- ✓ De update-directory van een installatie-CD kan in principe gewoon over de bestaande omgeving worden gekopieerd. Er is doorgaans geen bin of client/bin in aanwezig, zodat opstartscripts niet worden overschreven. Ook zijn er in een update geen zaken aanwezig die specifiek zijn voor de betreffende client of server.
- ✓ Als er metachanges op de tabellen CLASSX en CLASS\_ATTRIBUTEX zijn en er draaien verschillende High-T servers op één en hetzelfde High-T-schema in de database (bv. acceptatie en productie), dan dienen de bestanden lib/opt/metaclass.jar en lib/high\_t/nc\_meta.jar ook naar de serveromgeving van de oudere (productie-) versie te worden gekopieerd.
- ✓ Na het updaten van de server- en clientomgevingen kan High-T weer gewoon worden gebruikt.

### 3. Java-versie

Het verdient aanbeveling om voor het draaien van de High-T client en server dezelfde Java Runtime Environment versie te gebruiken als voor het ontwikkelen en compileren van de High-T-programmatuur is gebruikt. Gebruik van een andere versie kan resulteren in foutsituaties, die niet altijd reproduceerbaar zijn in de ontwikkelomgeving en dan dus niet opgelost kunnen worden. In het algemeen zijn hogere JRE-versies echter goed bruikbaar.

Als Java™ 2 Runtime Environment-versie gebruiken wij: **JRE 1.7.0\_55-b13**.

Als Java™ Web Start-versie gebruiken wij: **1.7**.

Indien in de ontwikkelomgeving de JRE-versie wijzigt zal dit opgenomen worden in de release-notes. Uiteraard is dit dan een grote wijziging en zal dit ontaarden in een Major release.

Java 5.0 Runtime Environments zijn te downloaden vanaf:

<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

Java Webstart wordt als tool gebruikt om te garanderen dat de client-versie die op de client-machines in gebruik is altijd overeenkomt met de versie van de server-software. In principe wordt hier een Just In Time uitrol gedaan van de client-software ingeval van een versie-upgrade van High-T. Java Webstart is niet noodzakelijk om met High-T te kunnen werken, doch een tool om het de beheerder makkelijker te maken. Vanaf versie 6 is WebStart in de runtime geïntegreerd.

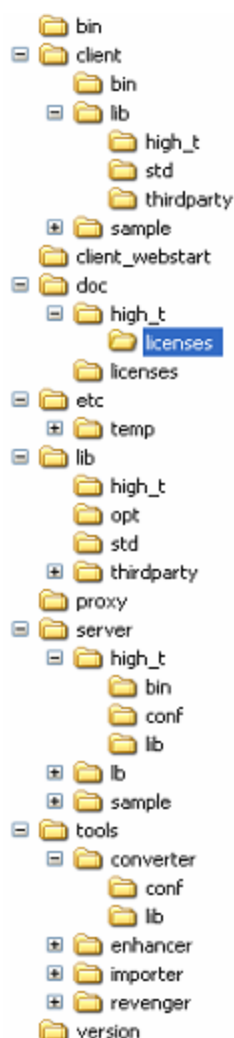
Wij nemen enig voorbehoud in acht m.b.t. wijzigingen in de inrichting van de website van Sun/Oracle.

**Oudere Java-versies dan 1.7 kunnen niet worden gebruikt omdat de broncode van High-T niet compatibel is met eerdere versies. Met latere Java-versies verwachten wij geen problemen, al stelt What If Software Solutions zich niet aansprakelijk voor problemen die optreden t.g.v. een andere Java-versie dan Java 1.7.0.**



## 4. Deployment

### 4.1. Globaal overzicht van de deploy-structuur



#### **bin**

Batchbestanden (Windows) en shellscripts (Unix) voor server en tools.

#### **client**

Client-omgevingen.

##### **client/bin**

Batchbestanden en shellscripts voor het opstarten van clients.

##### **client/lib**

Libraries voor de oude client en client/server libraries.

##### **client/lib/high\_t**

High-T-specifieke client- en client/server-libraries.

##### **client/lib/std**

Standaard libraries t.b.v. de client.

##### **client/lib/thirdparty**

Third party-software voor de client (leeg).

##### **client/lib/overig**

Specifieke client-omgevingen. Deze bevatten:

- \* een bin-directory met specifieke opstartscripts
- \* een conf-directory met specifieke configuratiebestanden
- \* een lib-directory met specifieke libraries

#### **doc**

Documenten met licentiegegevens.

#### **etc**

Parafernalia zoals icoontjes t.b.v. snelkoppelingen.

#### **lib**

Bevat server- en clientserver-libraries t.b.v. de oude client.

##### **lib/high\_t**

Bevat server- en clientserver-libraries t.b.v. High-T

##### **lib/opt**

Bevat libraries t.b.v. catalog en constraint checker.

##### **lib/std**

Bevat standaard libraries t.b.v. de server.

##### **lib/thirdparty**

Bevat third party-software die aan serverzijde wordt gebruikt.

#### **proxy**

Bevat software van de RMI proxy servlet.

#### **tools**

Bevat een viertal programma's voor datamigratie, genereren van Java-sources voor entity-klassen en reverse engineering.

#### **version**

Bevat een tekstbestand met versie-informatie.

## 4.2. De directory bin

Deze directory bevat batchfiles voor Windows (extensie **.bat**) en shellscripts voor Unix en Linux (geen extensies).

### 4.2.1. Windows-scripts

#### ***server.bat***

Het opstartscript voor de server. Dit script wordt opgestart met als argument de naam van een server die onder de subdirectory **server** staat (in ons voorbeeld: **server sample [-run | -install | -uninstall]**). Voor het installeren of deïnstalleren van een JavaService dient **-install** resp. **-uninstall** als tweede argument te worden meegegeven.

In dit script kunnen environment-variabelen worden ingevuld die voor alle servers gelden:

JAVA_EXEC	De executable voor de Java Virtual Machine. Default : <b>java</b> zonder padopgave. Hier wordt dus verondersteld dat de directory voor de JVM in PATH staat.
JAVA_STD_HEAP	De hoeveelheid gealloceerd geheugen. Default: <b>192m</b> .
JAVA_MAX_HEAP	De maximale hoeveelheid te alloceren geheugen. Default: <b>260m</b> .
JAVA_VMARGS	Overige JVM-argumenten voor de commandoregel. Geen default.

N.B. Deze variabelen kunnen worden overruled in de server-specifieke init-scripts.

De volgende variabelen zijn alleen van toepassing bij het (de)installeren van een JavaService.

SVC_NAME	De naam voor de JavaService. Default: <b>FAAJavaServiceserver_name</b> .
SVC_LOG_OUT	Het pad en de naam van de logfile voor de standard output. Default: <b>FAAJava_root/log/service_name.log</b>
SVC_LOG_ERR	Het pad en de naam van de logfile voor de standard error output. Default: <b>FAAJava_root/log/service_name.log</b>
SVC_JVM_DLL	De DLL die wordt gebruikt voor de JavaService.

Deze batchfile handelt de volgende zaken af:

Het zetten van de hierboven genoemde environment-variabelen.

Aanroep van **conf.bat** met de aan de commandoregel meegegeven opstart-argumenten

Voorwaardelijke aanroep van **init.bat** in de directory van de server

Lanceren van de server of (de)installeren van de JavaService

#### ***conf.bat***

Dit bestand wordt aangeroepen vanuit **server.bat** stelt het classpath samen uit de server-specifieke jarfiles, gevolgd door de algemene jarfiles in de directory **lib** en zijn subdirectories.

#### ***appenddynclasspath.bat***

Wordt aangeroepen vanuit **conf.bat** en voegt de als argument meegegeven jar-filenaam toe aan het classpath.

#### ***converter.bat***

Roept **conf.bat** aan om het classpath samen te stellen en start dan de Converter op.

#### ***enhancer.bat***

Roept **conf.bat** aan om het classpath samen te stellen en start dan de Enhancer op. Maximaal 9 argumenten.

#### ***importer.bat***

Roept **conf.bat** aan om het classpath samen te stellen en start dan de Importer op.

#### 4.2.2. Unix-scripts

##### *serverctl*

Het opstartscript voor de server. Het eerste argument is de servernaam (naam van de subdirectory onder **server**), het tweede is **start** (starten van de server), **stop** (stoppen van de server), **restart** (stoppen en opnieuw starten van de server) of **status** (draait de server?). Een Unix-variant op **server.bat**. Het opstarten van de server gebeurt in een achtergrondproces.

Het script roept **set\_classpath** en de **initjvm** van de server aan.

##### *set\_classpath*

Stelt het classpath samen, analoog aan **conf.bat**.

##### *add\_to\_classpath*

Voegt de als argument meegegeven jar-file aan het classpath toe, analoog aan **appenddynclasspath.bat**.

### 4.3. De directory client

Dit is de root-directory van de client. Deze bevat standaard de subdirectories **bin** en **lib** voor de batches resp. de jars. Overige directories bevatten server-specifieke zaken, waarvan **sample** een voorbeeld is.

#### 4.3.1. De bin-subdirectory - Windows-scripts

##### *client.bat*

Het opstartscript voor de client. In dit script kunnen waarden voor environmentvariabelen worden ingevuld:

JAVA_EXEC	De executable voor de JVM. Zie onder <i>server.bat</i> in §3.2.1.
JAVA_STD_HEAP	De hoeveelheid te alloceren geheugen. Default: <b>80m</b> .
JAVA_MAX_HEAP	De maximale hoeveelheid te alloceren geheugen. Default: <b>144m</b> .
JAVA_VMARGS	Eventueel extra mee te geven JVM-argumenten.
ARG_URL	De URL voor verbinding met de server. Default: <b>rmi://localhost:1099</b> .
ARG_LOGWINDOW	Geeft aan of de logging in een venster moet worden getoond. Geen default.
ARG_LOGLEVEL	Geeft het loggingsniveau aan (1=DEBUG, 2=INFO, ..., 5=FATAL). Geen default.
ARG_LNF	De look & feel (Windows, Metal, Motif). Geen default.

Al deze paramers zijn overridable in de init-batches van de clients. Met name van de laatste vier variabelen kan worden verwacht dat ze client-specifiek worden opgegeven.

Het script roept **conf.bat** en, indien aanwezig, de **init.bat** uit de subdirectory van de client aan. Deze werken analoog aan **conf.bat** en **init.bat** voor de server.

#### 4.3.2. De bin-subdirectory - Unix-scripts

##### *jnlpgen*

Dit script genereert een JNLP-file voor gebruik in combinatie met Java Web Start. Het script maakt gebruik van client-specifieke skeletdelen **jnlp\_header** en **application-desc**, die in de subdirectories van de afzonderlijke clients zijn opgeslagen. Verder worden de jars dynamisch toegevoegd m.b.v. de scripts **add\_resources** en **add\_resource**, die equivalenten zijn van **conf.bat** en **appenddynclasspath.bat**.

Het script wordt opgestart met één opstartargument: de specifieke subdirectory voor de client. In deze subdirectory wordt de JNLP-file geschreven (bijv.: **jnlpgen sample**).

Dit script hoeft slechts één keer per deployment te worden gedraaid: een JNLP-file hoeft niet te worden gewijzigd zolang de configuratie niet wijzigt.

#### 4.3.3. Specifieke client-zaken per server

##### *Subdirectory server\_name/bin*

De per server gedefinieerde subdirectory *server\_name/bin* kan een **init.bat** (Windows) of een **init** (Unix) bevatten. Hierin worden client-specifieke environment-variabelen gezet.

##### *Subdirectory server\_name/conf*

In deze directory staan een policy-file en, voor een Unix-omgeving, enige client-specifieke settings voor een JNLP-file. Het bestand **jnlp-header** bevat het bovenste deel van de JNLP-file, tot en met de security.

Het bestand **application-desc** bevat client-specifieke commandoregel-argumenten.

#### 4.3.4. WebStart-omgevingen

In een WebStart-omgeving hoeven alleen HighT\_webstart.jar en lib\_webstart.jar aanwezig te zijn in client/lib. Deze twee jar-files bevatten de client- en client/server-code uit de High-T-specifieke en de gebruikte standaard-libraries.

Deze jars kunnen ook in niet-WebStart-omgevingen worden gebruikt i.p.v. de overige client-jars.

## 4.4. De directory lib

Deze directory bevat de toolset-jars, die door client en server worden gebruikt en die niet applicatie-specifiek zijn. In de directory zelf staan de algemene jars die door de server en de client worden gebruikt. Deze jars worden dynamisch in het classpath opgenomen bij het starten van de server of de JavaService.

### 4.4.1. De subdirectory lib/opt

Deze bevat jars met checkers, de FAA/Java Oracle-driver en de entity-classes.

### 4.4.2. De subdirectory lib/std

Deze bevat standaard libraries, afkomstig van o.a. Apache en JBoss, die door de serverzijde van de FAA/Java toolset worden gebruikt.

### 4.4.3. De subdirectory lib/thirdparty

Deze bevat JDBC-drivers voor Oracle, DB2, Microsoft SQL Server in afzonderlijke subdirectories.

## 4.5. De directory proxy

Deze bevat een war-file (**faajproxy.war**), die klassen voor een ServletHandler en twee configuratiebestanden (**web.xml** en **rules.xml**) bevat.

## 4.6. De subdirectory server

Dit is de root voor de verschillende FAA/Java servers die draaien. De afzonderlijke subdirectories hangen hieronder.

### 4.6.1. De subdirectory server/*server\_name*

In de directory zelf worden de logfile en (onder Linux/Unix) de pid-file van de server gezet. Ook is dit de geijkte plek voor wallpapers die door de toolset worden gebruikt (de plek hiervoor wordt als property “**server.home**” gezet in de JVM-argumenten voor het opstarten van de server.

### 4.6.2. De subdirectory server/*server\_name*/bin

Deze kan een init-file bevatten (**init.bat** onder Windows of **initjvm** onder Unix), waarin server-specifieke environment-variabelen worden gezet.

### 4.6.3. De subdirectory server/*server\_name*/conf

Deze bevat een policy-file en configuratie-bestanden voor LOG4J (**log4j.properties**) en de server (**server.properties**).

### 4.6.4. De subdirectory server/*server\_name*/lib

Deze bevat optionele server-specifieke libraries. Hier kan ook de **Licenses.jar** in staan, hoewel deze de facto in de directory **lib** thuishoort.

## 4.7. De subdirectory Tools

Dit is de subdirectory voor Converter, Enhancer, Revenger en Importer. De jars voor deze programma's staan in de lib-directory van hun afzonderlijke subdirectories; de opstartbestanden staan in de directory **bin** (zie §3.2).

## 4.8. Overige meegeleverde zaken

### 4.8.1. Metachanges

In de metachanges-directory staan de metachanges die bij de laatste High-T-release horen. Deze dienen te worden gedraaid om de inhoud van de database compatibel te houden met de High-T-programmatuur.

### 4.8.2. Miscellaneous

Deze directory bevat diverse zaken, die per release kunnen wisselen. Enige constanten zijn:

#### ***BASE\_hhh-bbb.dmp***

Een dump van de BASE-database, de schone database, die compatibel is met de laatste officieel uitgebrachte High-T-versie. De letters *hhh-bbb* staan voor het buildnummer waar deze BASE-dump bij hoort.

De dump kan worden gebruikt om een nieuwe High-T-database op te zetten.

#### ***High-T Installatie NL.pdf***

Voor een uitgebreide samenvatting zie de rest van dit document.

#### ***Uitbreiden van FAA entity klassen.pdf***

Een beschrijving van het definiëren, genereren en deployen van entity-klassen m.b.v. de Enhancer.

#### ***UserSysGrantsPermissionToFAAJ.sql***

Een script dat door de DBA (username=SYS, connected as SYSDBA) gedraaid dient te worden na het aanmaken van een gebruiker FAAJ in de database om deze gebruiker de benodigde privileges te geven.

## 5. De High~T server

### 5.1. De commandoregel voor het opstarten

De opstartinstructie, vanuit de directory `bin`, luidt:

**serverctl *server\_name* start** (Linux/Unix), of  
**server *server\_name*** (Windows)

De commandoregel, die uit de aangeroepen scripts volgt, bevat standaard de volgende instructies:

```
java -Djava.security.policy=policyfile -Xrs [runtime_opties]  
-Dlog4j.configuration=log4jfile -cp dynclasspath  
nl.introspec.faaaj.server.FAAJServerApplication [java_vmargs]  
propertyfile=propertyfile
```

Hierin is het bestand *policyfile* een Java policy-file, bedoeld voor de Java Virtual Machine. Dit bestand heeft standaard de volgende inhoud:

```
grant {  
  permission java.security.AllPermission;  
};
```

In de runtime-opties `-Xmsxxxm` en `-Xmxyyyym` zijn de getallen `xxx` en `yyy`, gevolgd door de letter 'm', de standaard en maximale hoeveelheden geheugen in megabytes aan, die door het programma gebruikt mogen worden. Deze opties dienen afhankelijk van de grootte van de database aan de commandoregel te worden toegevoegd om memory overflow te voorkomen

#### Voorbeeld

```
java -Djava.security.policy=../server/sample/conf/server.pol -Xrs -Xms192m -Xmx260m  
-Dlog4j.configuration=../server/sample/conf/log4j.properties  
-Dserver.home=../server/sample -cp [...] nl.introspec.faaaj.server.FAAJServerApplication  
propertyfile=../server/sample/conf/server.properties
```

De omkaderde commandoregel start een High-T server op. Hiervoor wordt standaard 192 MB geheugen gealloceerd. De hoeveelheid geheugen kan echter gedurende de sessie worden uitgebreid tot een maximum van 260 MB. De security policy staat beschreven in het bestand `server.pol`, de opstartgegevens (naam, database, authenticatiemodule etc.) in het bestand `sample.properties`.

De commandoregel-instructies wordt aangevuld met één voor High~T van belang zijnde instructie: **propertyfile=*propertyfile***. In het configuratiebestand met de naam *propertyfile* worden alle van belang zijnde gegevens opgenomen m.b.t. drivers, database, data sources, client-authorisatie etc..

## 5.2. Het configuratiebestand

### 5.2.1. Eigenschappen: Algemeen, Data Sources en Persistence Manager Factories

#### 5.2.1.1. Algemeen

Het configuratiebestand kan alle in §3.1 genoemde opties bevatten, alsmede specificaties van drives, data sources, password-encryptie, Op een rij gezet zijn deze eigenschappen de volgende:

<b>jars</b>	Bevat jars die aan het jar-path toegevoegd kunnen worden.
<b>host</b>	Als de server meerdere IP-adressen heeft dient het te gebruiken IP-adres expliciet te worden opgegeven; is er echter één IP-adres, dan wordt dit vanzelf gebruikt en is deze parameter overbodig.
<b>port</b>	Bevat het TCP/IP-poortnummer t.b.v. de client. Default= <b>1099</b> .
<b>name</b>	Bevat de server-naam. Deze kan door de gebruiker naar believen worden gekozen. Default="FAA/Java".
<b>licenseCode</b>	De code van de licentie. Als deze bv. TESTLIC is, moet er een licentie TESTLICFAAJLicense.class in de Licenses.jar aanwezig zijn.
<b>workerThreads</b>	Het aantal worker threads in de thread pool
<b>keepAliveTimeout</b>	De timeout in seconden voordat er een keep alive-thread afgaat.
<b>freezeld</b>	Een string van één karakter (numeriek of upper case) dat als 4 <sup>e</sup> karakter van de sneak preview-functienaam wordt gebruikt. Default=0.
<b>passwordEncryption</b>	Gebruik van password-encryptie. Toegestaan zijn <u>true</u> en <u>false</u> . Default= <b>true</b> .
<b>cacheDataSource</b>	Bevat de naam van de cache data source. Default= <b>java:comp/env/jdbc/cache</b> .
<b>instanceDataSource</b>	Bevat de naam van de instance data source. Default= <b>java:comp/env/jdbc/instance</b> .
<b>instanceQueryDataSource</b>	Bevat de naam van de instance query data source. Geen defaultwaarde. Het gebruik van een aparte query data source voor transacties waarbij geen mutaties op de database plaatsvinden is niet verplicht. Als er geen query data source is gedefinieerd wordt voor queries dezelfde data source gebruikt als voor mutaties. De belangrijkste reden om onderscheid tussen de data sources te maken is het definiëren van verschillende <i>isolation levels</i> .
<b>instancePersistenceManagerFactory</b>	Bevat de naam van de instance persistence manager factory. Default= <b>java:comp/env/jdo/tx</b> .
<b>instanceQueryPersistenceManagerFactory</b>	Bevat de naam van de instance query persistence manager factory. Geen defaultwaarde.

Vervolgens zijn er de volgende registries, waarin drivers en data sources met hun eigenschappen worden vastgelegd. Dit vastleggen geschiedt als volgt: Per eigenschap wordt vastgelegd:

**registry.id.property=value**

Hierin is **registry** een van de hieronder genoemde registries, **id** een door de gebruiker op te geven, binnen de registry unieke, identificatiecode voor de instantie; **property** is naam van een bij deze instantie behorende eigenschap, en **value** is de waarde die aan deze eigenschap wordt toegekend. Als er bijvoorbeeld staat: "jdbcDataSource.0.user=FAAJ" betekent dit dat de eigenschap "user" van de data source, die in de JDBC data source-registry onder de identificatiecode "0" bekend is, de waarde "FAAJ" heeft aangenomen.

**jdbcDriver** Van de JDBC drivers worden de klassenamen (**name**) vastgelegd t.b.v. het laden van de driver.

**jdbcDataSource** Van de data sources worden de eigenschappen op generieke wijze vastgelegd. De eigenschap **className**, die de klassenaam bevat waartoe de data source behoort, is verplicht. De overige eigenschappen zijn vendor-specifiek.

Bij het gebruik van pooled data sources verdient het aanbeveling om, ook bij een gering aantal



clients, de maximale en de initiële pool size (`maxPoolSize` resp. `initialPoolSize`) geen te lage waarde te geven (bij waarde 10 voor beide eigenschappen werkt dit goed) ter voorkoming van problemen bij transacties op de database.

`jdbcConnectionPoolDataSource`

`jdoPersistenceManagerFactory` Voor de persistence manager factories geldt hetzelfde als voor data sources.

`autoStart` De in deze registry gedefinieerde zaken hebben slechts één eigenschap: `function`, oftewel de functienaam. Deze functie, een serverfunctie gedefinieerd in de tabel `FUNCTIONX`, wordt bij het opstarten van de server uitgevoerd zonder parameters.

#### 5.2.1.2. De DBCP BasicDataSource

Vanaf High-T versie 1.4.2 maken wij gebruik van de Commons DBCP-component (database connection pooling service) van Apache. Deze component maakt gebruik van een connection pool.

Configuratie-instellingen voor deze data source en hun defaultwaarden zijn overzichtelijk gedocumenteerd op de webpagina <http://jakarta.apache.org/commons/dbcp/configuration.html>.

Voor wat betreft het aantal connecties in de pool is het van belang om de parameters `maxActive` en `initialSize` zo in te stellen dat deze aantallen toereikend zijn voor het aantal clients dat actief is op een High-T server.

#### 5.2.2. Eigenschappen: Login-authorisatie

Deze eigenschappen worden ingesteld op het moment dat de eerste client inlogt. De formule voor de settings van de login-authorisatie is:

```
auth.appl.id.property = value
```

Hierin staat “appl” voor de applicatie (moet “FAAJClient” zijn), “id” voor de unieke id van een klasse die hiermee gemoeid is, en “property” voor de eigenschap die gezet moet worden.

De eigenschap “module” geeft de klassenaam van de login-module aan; de eigenschap “controlFlag” kan vier waarden hebben: “REQUISITE”, “REQUIRED”, “OPTIONAL” en “SUFFICIENT”.

De definities t.b.v. de FAA inlogprocedure (`nl.introspec.faa.server.auth.FAAApplicationLoginModule`) zijn verplicht.

Voor de passwordauthorisatie zijn verschillende login-modules te gebruiken:

- ① De JDBC login module (user en password t.b.v. de database), of
- ② De FAA password login module (maakt gebruik van de tabel `PASSWORDX`), of
- ③ De NT login module (`com.sun.security.auth.module.NTLoginModule`), of
- ④ een vendor-module .

De te definiëren eigenschappen kunnen per module verschillen.

Voor een voorbeeld van een configuratiebestand zie onder <sup>o</sup>Appendix A.

### 5.3. Het opstarten en stoppen van de server

Voor het opstarten van de High-T server wordt doorgaans een batch-file (in Unix/Linux-termen: shell script) gebruikt. Voor het stoppen van de High-T server bestaat in Windows geen state of the art-commando, afgezien van het uitvoeren van de functie SHUTDOWN vanaf de client: dit wordt gewoon gedaan door het proces af te breken; een JavaService kan echter vanuit het Services-scherm worden gestart en gestopt. In een Unix- of Linux-omgeving kan “`serverctl server_name stop`” worden ingegeven.

Voor de logging dient aan de serverzijde een bestand aanwezig te zijn, waarvan de naam uit de system-property `log4j.configuration` wordt gelezen. Standaard heet dit bestand `log4j.properties`. Dit bestand dient in de `server/conf`-directory aanwezig te zijn. Een voorbeeld van dit bestand is opgenomen in de `sample/conf`-directory van de server. Dit bestand is naar believen door de systeembeheerder aan te passen.

Richtlijnen voor het configureren van log4j vallen buiten het bereik van dit document. Hiervoor wordt verwezen naar <http://jakarta.apache.org/log4j/docs/manual.html>.

#### 5.3.1. De High-T-server in een Windows/NT-omgeving

In de Windows- of NT-omgeving wordt de server doorgaans opgestart d.m.v. een batch-file. Deze bevat doorgaans niets anders dan de commandoregel (%S2.1), welke voorafgegaan wordt door een “@echo off” en gevolgd door een “pause” instructie. Het sluiten geschiedt door op het kruisje rechts boven in het venster te klikken of door een ingreep m.b.v. de Task Manager (oproepbaar door de Ctrl-, Alt- en Del-toetsen gelijktijdig in te drukken).

#### *JavaService (Copyright © 2000 Alexandria Software Consulting)*

Het is mogelijk om een High-T server als een NT-service te laten draaien in een Windows NT/2000-omgeving. Met dit doel is door Alexandria Software Consulting een Win32-executable (JavaService.exe), nu bij Multiplan in beheer, ontwikkeld, welke te downloaden is vanaf de website [http://forge.objectweb.org/project/showfiles.php?group\\_id=137](http://forge.objectweb.org/project/showfiles.php?group_id=137). Dit programma kan, indien gewenst, *as is* met de High-T-applicatie meegeleverd worden; Introspec Implementations BV stelt zich echter niet aansprakelijk voor enigerlei schade, welke voortvloeit uit het gebruik van dit product.

De commandoregel voor het installeren van een High-T server is:

```
JavaService.exe -install %SVC_NAME% %SVC_JVM_DLL% -Xms%JAVA_STD_HEAP%m -Xmx%JAVA_MAX_HEAP%m
%JAVA_VMARGS% -Djava.security.policy=..\server\server_name\conf\server.pol
-Dlog4j.configuration=..\server\server_name\conf\log4j.properties
-Djava.class.path=dynclasspath -showversion
-start nl.introspec.faaj.server.FAAJServerApplication -params
propertyfile=..\server_name\conf\server.properties -out %SVC_LOG_OUT%
-err %SVC_LOG_ERR% -current current_directory
```

De commandoregel wordt opgestart door ingave van: `server server_name -install`. De environment-variabelen worden gezet in `server.bat`, of, liever nog, in `..\server_name\bin\init.bat`.

Het deinstalleren van een High-T NT-server gaat als volgt:

`server server_name -uninstall`

### 5.3.2. De High-T-server in een Unix-omgeving

In een Unix- of Linux-omgeving wordt de server opgestart en gestopt m.b.v. een shell script. Dit script controleert op een elegante manier of er al een server is opgestart en start, als dit niet zo is, zelf een server op. Een script voor het opstarten van een High-T server kan eventueel in een opstartscript voor Unix worden gehangen.

Een voorbeeld van een opstartscript staat in <sup>o</sup>Appendix B.

Ook is er een voorbeeld opgenomen in de samples-subdirectory van de server.

### 5.3.3. Opmerking m.b.t. de High-T server in een AS/400-omgeving

Als in een AS/400-omgeving jt400native.jar wordt gebruikt mag er geen jt400.jar in het classpath aanwezig zijn. De JRE krijgt er namelijk problemen mee als beide jars worden geladen.

## 5.4. De High-T licentie

### 5.4.1. Het bestand Licenses.jar

De gegevens van de High-T-licentie staan in het bestand `Licenses.jar`. Dit bestand dient in het classpath van de server te staan. Het bevat o.a. de volgende gegevens:

- ✓ De expiratedatum van de licentie.
- ✓ Het aantal named users.
- ✓ Het aantal concurrent users.
- ✓ Een indicator die aangeeft of de licentienemer een onderhoudscontract heeft.
- ✓ Een toegangscode, die als `licenseCode` in de propertyfile van de server wordt gezet. Dit is een naam in hoofdletters, die het begin vormt van de naam van de Java-klasse met de licentiegegevens. Dit is doorgaans een verkorte variant van de naam van de licentienemer.
- ✓ De naam van de licentienemer.

Het bestand `Licenses.jar` wordt niet gelezen aan de cliëntzijde. De cliënt haalt de licentiegegevens bij de server op.

### 5.4.2. Named en concurrent users

#### *Named users*

Named users zijn in de tabel `USERX` gedefinieerde gebruikers. Het toegestane aantal named users mag niet worden overschreden in de database. Gebeurt dit wel, dan kan de server niet worden opgestart en zullen er named users uitgeschakeld moeten worden buiten de toolset om. Dit kan door ze fysiek uit de database te verwijderen of door ze als disabled (`IND_DISABLED = 'Y'`) of expired aan te merken.

Een named user voldoet aan de volgende criteria:

- ✓ Een named user behoort tot het systeem uit `BOOTSTRAPX`.
- ✓ Een named user is geen system owner.
- ✓ Een named user behoort tot een groep in `USER_TYPE_STRUCTX` waar geen groepen onder hangen.
- ✓ Een named user is niet disabled.
- ✓ Een named user is niet expired.

De toolset controleert bij wijzigingen in de gebruikersgegevens of het toegestane aantal named users niet wordt overschreden.

#### *Concurrent users*

Concurrent users zijn gebruikers die gelijktijdig op de High-T server zijn ingelogd. Dit aantal kan niet worden overschreden: er worden geen clients tot de server toegelaten voordat er andere zijn uitgelogd.

Als een cliënt niet op de normale manier wordt afgesloten kan het zijn dat de server deze cliënt niet onmiddellijk deregistreert. Tot het moment van deregistratie telt deze cliënt mee als een concurrent user.

Ten overvloede: Het kan zijn dat meerdere clients zijn ingelogd met dezelfde inlognaam. In dit geval wordt het aantal malen dat deze gebruiker is ingelogd geteld bij het bepalen van het aantal concurrent users.

### 5.4.3. Werking van de licentie

De licentiegegevens worden bij het opstarten van de High-T server gecontroleerd. Als de licentie verlopen is kan de toolset nog tot de 30<sup>e</sup> dag na het verlopen van de licentie worden gebruikt. Er verschijnen echter waarschuwingen bij het opstarten van een client. Hierna kan de server niet meer worden opgestart,

Als er meer named users in de database aanwezig zijn dan toegestaan kan de server niet worden opgestart.

Als de gebruiker wijzigingen in de licentie heeft aangebracht kan de server niet worden opgestart.

## 6. De High-T Client

### 6.1. Het opstarten van de client

#### 6.1.1. Beschrijving van het opstartcommando

Evenals de server heeft ook de client een aantal argumenten in de opstartregel staan, alleen is hier de volgorde van de argumenten bepalend.

De commandoregelargumenten zijn:

url	De URL voor de connectie met de server (bv. rmi://localhost:1099)
user	De loginnaam van de gebruiker (niet case sensitive)
passwd	Het wachtwoord van deze gebruiker
rmiProxyServlet	Een URL naar de RMI proxy servlet handler
loglevel	Het loggingniveau (1=DEBUG, ..., 5=FATAL)
domainLogin	Login vanaf domeinnaam? (default= <b>true</b> )
Inf	Look and Feel (default=Windows)

#### 6.1.2. De commandoregel

Voor het opstarten van de client wordt de runtime **javaw** (of **java**) gebruikt.

De commando regel ziet er derhalve als volgt uit:

```
javaw -cp {classpath} -Djava.security.policy=policyfile [runtime_opties] newclient.Client [url=URL]
[loglevel=loglevel] [user=gebruikersnaam] [passwd=wachtwoord] [rmiproxyservlet=url]
[loginFromDomain=true|false|y|n] [Inf=Windows|Metal|Motif|Swing|lookAndFeelClassName]
```

##### Voorbeelden

```
javaw -cp {classpath} -Djava.security.policy=client.pol -Xms80m -Xmx192m newclient.Client
url=rmi://faajhost:1303 loglevel=2 user=FAAJSYS passwd=1217JS12
```

Deze regel start een High-T client op met 80MB geheugen initieel gealloceerd, uit te breiden tot maximaal 192MB. De server staat op "faajhost" en communiceert via TCP/IP-poort 1303. Er wordt ingelogd als user FAAJSYS met wachtwoord 1217JS12. De client heeft een Windows look & feel. Er wordt gelogd op INFO-niveau, waarbij de logging in een apart venster wordt getoond.

```
javaw -Djava.security.policy=client.pol -classic -Xms100m -Xmx160m -jar Client.jar
url=rmi://192.224.46.41:1099 loglevel=4
```

Deze regel start een High-T client op met 100MB initieel geheugen, uit te breiden tot maximaal 160MB. De server draait op een host met IP-adres 192.224.46.41, poort 1099. De logging wordt op ERROR-niveau geschreven. Gebruikersnaam en wachtwoord worden ingegeven.

Zie §5.1 voor de policy-file en de runtime-opties.

## 6.2. Het configureren van een Java™ Web Start-omgeving voor de client

Op een webpagina dient een referentie naar een XML-bestand **Client.jnlp** te worden opgenomen. Deze referentie is een conventionele HTML-link en kan zowel een tekst als een afbeelding zijn.

```
<a href="Demo_Client/Client.jnlp"></a>
```

In dit voorbeeld is de afbeelding “partnersgr.jpg” een link naar het bestand genaamd Client.jnlp in de onderliggende map Demo\_Client.

Het JNLP-bestand ziet er als volgt uit:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- JNLP file for High~T -->
<jnlp
  spec="1.0+"
  codebase="http://server_name/map_name"
  href=" http://server_name/map_name/client.jnlp">
  <information>
    <title>Demo Client</title>
    <vendor>Introspec Implementations</vendor>
    <homepage href="http://www.faapartners.com"/>
    <description>Demo Client</description>
  </information>
  <security>
    <all-permissions/>
  </security>
<!-- Alle voorafgaande regels komen uit conf/jnlp_header -->
  <resources>
    <!-- Deze regels worden automatisch samengesteld vanuit jnlpgen -->
    <j2se version="1.5+" initial-heap-size="80m" max-heap-size="144m"/>
    <jar href="lib/HighT_webstart.jar" main="true" download="eager"/>
    <jar href="lib/lib_webstart.jar" main="false" download="eager"/>
  </resources>
<!-- De 'application-desc'-sectie komt uit conf/application_desc -->
  <application-desc main-class="nl.introspec.faaj.client.Main">
    <argument>url=rmi://server_name:1099</argument>
    <argument>useCallback=false</argument>
    <argument>lnf=Windows</argument>
  </application-desc>
</jnlp>
```

## 7. High-T Webservices

**⚠ Deze functionaliteit staat in feite nog in de kinderschoenen, dus de inhoud van dit chapter kan enigszins afwijken van de actuele situatie.**

### 7.1. De configuratiefile

De syntax van een configuratiefile voor High-T webservices is ongetwijfeld afhankelijk van welke applicatie-server er gebruikt wordt voor het lanceren van webservices. Dit is een tak van sport die buiten de scope van deze handleiding valt. De configuratiefile die in deze handleiding wordt beschreven maakt gebruik van Apache Tomcat.

In een Tomcat-omgeving staat de configuratiefile in de directory `$TOMCAT/[webservice]/WEB-INF`. Hierin staat `$TOMCAT` voor de directory waarin de Tomcat-versie staat en `[webservice]` voor de naam van de betreffende webservice. De configuratiefile heet hier altijd `web.xml`.

In de configuratiefile worden twee servlets geconfigureerd:

#### 7.1.1. HighTServlet (nl.wiss.high\_t.client.web.HighTClientServlet)

Deze servlet is een de facto client die inlogt op de High-T server en geen gebruik maakt van grafische (awt, swing) klassen. Deze servlet verzorgt het afhandelen van de XML input requests.

De servlet kent de volgende input parameters:

Parameter	Betekenis
hight_host	De naam van de host waarop de High-T server resideert.
hight_port	De TCP/IP-poort die gebruikt wordt voor communicatie met de High-T server.
hight_protocol	Het protocol dat gebruikt wordt voor communicatie met de High-T server.
schema_loc	De locatie op de webservice-omgeving waar XSD's staan.
hight_user	De gebruikersnaam waarmee de webservice op High-T inlogt.
hight_password	Het wachtwoord dat de webservice gebruikt om in te loggen.

#### 7.1.2. HighTCaller (nl.wiss.webforms.HighTWebServiceImpl)

Deze servlet wordt aangeroepen in de web browser. Deze start op met een scherm waarin een gebruikersnaam en password wordt gevraagd. Bij correct inloggen wordt het High-T menu op een webpagina getoond.

De servlet kent de volgende input parameters:

Parameter	Betekenis
hight_host	De naam van de host waarop de High-T server resideert.
hight_port	De TCP/IP-poort die gebruikt wordt voor communicatie met de High-T server.
hight_protocol	Het protocol dat gebruikt wordt voor communicatie met de High-T server.
servlet_host	De host waarop de servlet resideert.
servlet_service	De naam waaronder de webservice geregistreerd staat.
servlet_port	De TCP/IP-poort die door de applicatie-server wordt gebruikt.

Voor een voorbeeld van een High-T webservice config-file zie <sup>o</sup>Appendix D.



## 7.2. Locatie van de programmatuur

In de webspecifieke directory (deze hangt onder de home-directory van de application server en heeft de naam van de webservice) is een subdirectory **WEB-INF**. In deze WEB-INF zijn enkele subdirectories aanwezig: twee van deze subdirectories zijn **classes** en **lib**.

In **classes** staan de Java-classes die door de webservice worden gebruikt. De High-T-specifieke classes staan in de packages **nl/wiss/high\_t/client/web** en **nl/wiss/webforms** en eventuele onderliggende directories.

In **lib** staan de standaard libraries die door de High-T webservices worden gebruikt, in casu de client-jars HighT\_webstart.jar en lib\_webstart.jar.

In de **namespace**-subdirectory staat nog een XSD voor de provisorische High-T inlogpagina.

## 7.3. XSD's

In de webspecifieke directory is een subdirectory **xsd**. Deze bevat enkele XSD-bestanden die gebruikt worden voor het valideren van enkele algemene High-T requests en headers.

De scherm-specifieke requests worden gevalideerd tegen generieke, niet uitgeschreven XSD's.

## 7.4. Het opstarten van de webservice

Dit dient te gebeuren via de applicatieserver (in casu Tomcat Manager) nadat de High-T server, waarop deze inlogt, is opgestart. De webservice logt namelijk als een onzichtbare client in op de High-T server.

## Appendices

### Appendix A: Voorbeeld van een configuratiebestand

```
port=4711
name=FAA/J Demo

licenseCode=FAAJAVA
workerThreads=60
keepAliveTimeout=60
sessionLogging=on

cacheDataSource=java:comp/env/jdbc/cache
// instanceDataSource=java:comp/env/jdbc/instance
// instanceQueryDataSource=java:comp/env/jdbc/instance_query

instancePersistenceManagerFactory=java:comp/env/jdo/tx
// instanceQueryPersistenceManagerFactory=java:comp/env/jdo/query

# Definitie van de driver en de FAA driver wrapper:
jdbcDriver.0.name=oracle.jdbc.driver.OracleDriver
jdbcDriver.1.name=nl.introspec.faj.sql.oracle.OracleDriver

# Definitie van de Oracle® cache data source.
# Deze verwijst naar een Oracle® database met instancenaam "faajdemo" op de server
# "demo".
# De connectie loopt via TCP-poort 1521. De server logt hier in als "faj" met
# wachtwoord "demofaap".
jdbcDataSource.0.name=java:comp/env/jdbc/cache
jdbcDataSource.0.className=nl.introspec.dbcp.BasicDataSource
jdbcDataSource.0.username=faj
jdbcDataSource.0.password=demofaap
jdbcDataSource.0.url=jdbc:oracle:thin:@elbonia:1521:demo
jdbcDataSource.0.driverClassName=oracle.jdbc.driver.OracleDriver
jdbcDataSource.0.exceptionProcessorClassName=nl.introspec.faj.sql.oracle.OracleExceptionProcesso
r
jdbcDataSource.0.sessionIdQuery=SELECT sys_context ('USERENV', 'SESSIONID') FROM dual
jdbcDataSource.0.namePrefix=Cache
jdbcDataSource.0.validationQuery=select 41 from dual
jdbcDataSource.0.initialSize=10
jdbcDataSource.0.maxActive=12
jdbcDataSource.0.maxWait=5000
jdbcDataSource.0.defaultAutoCommit=false
jdbcDataSource.0.defaultReadOnly=false
jdbcDataSource.0.defaultTransactionIsolation=2
jdbcDataSource.0.removeAbandoned=true
jdbcDataSource.0.removeAbandonedTimeout=120
jdbcDataSource.0.logAbandoned=true

# Definitie van de transaction persistence manager factory:
jdoPersistenceManagerFactory.1.name=java:comp/env/jdo/local/tx
jdoPersistenceManagerFactory.1.className=nl.introspec.faj.jdo.jdbc.JdbcPersistenceManagerFactor
y
jdoPersistenceManagerFactory.1.namePrefix=LocalTxPM
```

```
jdoPersistenceManagerFactory.1.connectionFactoryName=java:comp/env/jdbc/instance
jdoPersistenceManagerFactory.1.keepConnection=0
jdoPersistenceManagerFactory.1.connectionUserName=faaj
jdoPersistenceManagerFactory.1.connectionPassword=demofaap

jdoPersistenceManagerFactory.2.name=java:comp/env/jdo/local/query
jdoPersistenceManagerFactory.2.className=nl.introspec.faa.jdo.jdbc.JdbcPersistenceManagerFactor
y
jdoPersistenceManagerFactory.2.namePrefix=LocalQueryPM
jdoPersistenceManagerFactory.2.connectionFactoryName=java:comp/env/jdbc/cache
jdoPersistenceManagerFactory.2.keepConnection=0
jdoPersistenceManagerFactory.2.connectionUserName=faaj
jdoPersistenceManagerFactory.2.connectionPassword=demofaap
jdoPersistenceManagerFactory.2.nontransactionalRead=1

# Definities t.b.v. automatisch op te starten routines
# autostart.0.function=STRTSESLOG

# Definities t.b.v. de FAA inlogprocedure:
auth.FAAJClient.0.module=nl.introspec.faa.server.auth.FAAApplicationLoginModule
auth.FAAJClient.0.controlFlag=REQUIRED
auth.FAAJClient.0.tryFirstPass=true

# Definities t.b.v. de inlogprocedure voor de database:
// auth.FAAJClient.1.module=nl.introspec.faa.server.auth.JdbcLoginModule
auth.FAAJClient.1.module=nl.introspec.faa.server.auth.FAAPasswordLoginModule
auth.FAAJClient.1.controlFlag=REQUIRED
auth.FAAJClient.1.useFirstPass=true
auth.FAAJClient.1.datasource=java:comp/env/jdbc/local/instance
auth.FAAJClient.1.moduleBanner=true

plugin.0.className=nl.introspec.faa.server.HighTServerProvider
plugin.0.name=High~T
```

## Appendix B: Een opstartscript voor de High-T server in een Unix-omgeving

Een opstartscript voor een server op de database “faajdemo”, die resideert op server “demo”. De directorynaam \$deploy slaat op de root directory van de deployment. Voor een bijbehorende propertyfile zie °Appendix A.

### \$deploy/server/faajdemo/initjvm

```
#!/bin/bash

# Setting the Java Virtual Machine

JAVA_EXEC="/usr/java/j2sdk1.5.0/jre/bin/java"

JAVA_VMARGS="-Dserver.home=/opt/faajava/faajtest/server/faajdev"

# Setting the standard and maximum heap sizes.

JAVA_STD_HEAP="144m"
JAVA_MAX_HEAP="256m"
```

In dit bestand staan de Java Virtual Machine, de optionele JVM-argumenten en de hoeveelheden standaard en maximaal te alloceren geheugen gedefinieerd. In combinatie met de policy- en property-files in de directory \$deploy/server/demo/conf zijn alle gegevens gedefinieerd om de server op te kunnen starten m.b.v. het script \$deploy/bin/serverctl (commando: serverctl faajdemo start).

## Appendix C: Opstartparameters voor de FAA/Java client in een Unix-omgeving

De opstartparameters behorende bij de client op de in appendices A en B beschreven server op database FAAJDEMO:

### \$deploy/client/faajdemo/bin/init

```
#!/bin/bash
```

```
STD_HEAP_SIZE="80m";export STD_HEAP_SIZE  
MAX_HEAP_SIZE="144m";export MAX_HEAP_SIZE
```

Deze parameters zetten de standaard en maximale hoeveelheden te alloceren geheugen.

### \$deploy/client/faajdemo/conf/jnlp\_header

```
<?xml version="1.0" encoding="utf-8"?>  
  
<jnlp spec="1.0+"  
  codebase="http://demo/faajava/client"  
  href="http://demo/faajava/client/faajdemo/client.jnlp">  
  <information>  
    <title>FAAJ Client</title>  
    <vendor>Introspec Implementations</vendor>  
    <homepage href="http://www.ockham.nl"/>  
    <description>High~T Client FAAJDEMO</description>  
    <offline-allowed/>  
  </information>  
  <security>  
    <all-permissions/>  
  </security>
```

Dit is de inhoud van het bovenste deel van het door \$deploy/client/bin/jnlpgen aan te maken JNLP-bestand. In de `codebase` wordt de root van de clients opgegeven; in de `href` de link

### \$deploy/client/faajdemo/conf/application-desc

```
<application-desc main-class="nl.introspec.faaj.client.Main">  
  <argument>url=rmi://demo:1099</argument>  
  <argument>useCallback=false</argument>  
  <argument>lnf=Windows</argument>  
</application-desc>
```

Dit is het deel van de te genereren JNLP-file waarin de commandoregel-argumenten voor de client zijn gedefinieerd.

De JNLP-file wordt gegereerd door in \$deploy/client/bin in te toetsen: `jnlpgen faajdemo`.

## Appendix D. Voorbeeldconfiguratie van een High-T webservice "hightweb"

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements.  See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <description>
    Servlet and JSP Examples.
  </description>
  <display-name>Servlet and JSP Examples</display-name>

  <!-- Define servlets that are included in the example application -->

  <servlet>
    <servlet-name>HighTServlet</servlet-name>
    <servlet-class>nl.wiss.high_t.client.web.HighTClientServlet</servlet-class>
    <init-param>
      <param-name>hight_host</param-name>
      <param-value>vlnx001.ockham.nl</param-value>
    </init-param>
    <init-param>
      <param-name>hight_port</param-name>
      <param-value>3239</param-value>
    </init-param>
    <init-param>
      <param-name>hight_protocol</param-name>
      <param-value>rmi</param-value>
    </init-param>
    <init-param>
      <param-name>schema_loc</param-name>
      <param-value>/usr/tomcat/apache-tomcat-
6.0.20/webapps/hightweb/xsd</param-value>
    </init-param>
    <init-param>
      <param-name>hight_user</param-name>
      <param-value>faaj</param-value>
    </init-param>
    <init-param>
      <param-name>hight_password</param-name>
      <param-value>partners</param-value>

```

```
</init-param>
<!--
    <filter>
      <filter-name>LogFilter</filter-name>
      <filter-class>
        net.viralpatel.servlet.filters.LogFilter
      </filter-class>
      <init-param>
        <param-name>test-param</param-name>
        <param-value>This parameter is for testing.</param-value>
      </init-param>
    </filter>
    <filter-mapping>
      <filter-name>LogFilter</filter-name>
      <url-pattern>/*</url-pattern>
    </filter-mapping>
→
</servlet>

<servlet>
  <servlet-name>HighTCaller</servlet-name>
  <servlet-class>nl.wiss.webforms.HighTWebserviceCaller</servlet-class>
  <init-param>
    <param-name>hight_host</param-name>
    <param-value>vlx001.ockham.nl</param-value>
  </init-param>
  <init-param>
    <param-name>hight_port</param-name>
    <param-value>3239</param-value>
  </init-param>
  <init-param>
    <param-name>hight_protocol</param-name>
    <param-value>rmi</param-value>
  </init-param>
  <init-param>
    <param-name>servlet_host</param-name>
    <param-value>vlx001.ockham.nl</param-value>
  </init-param>
  <init-param>
    <param-name>servlet_service</param-name>
    <param-value>hightweb</param-value>
  </init-param>
  <init-param>
    <param-name>servlet_port</param-name>
    <param-value>4444</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>HighTServlet</servlet-name>
  <url-pattern>/servlets/HighTServlet/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>HighTCaller</servlet-name>
  <url-pattern>/servlets/HighTCaller</url-pattern>
</servlet-mapping>

<jsp-config>
```

```
<taglib>
  <taglib-uri>
    http://vlnx001.ockham.nl:4444/hightweb/namespace
  </taglib-uri>
  <taglib-location>
    /WEB-INF/namespace
  </taglib-location>
</taglib>
</jsp-config>

<security-constraint>
  <display-name>Example Security Constraint</display-name>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
    <!-- Define the context-relative URL(s) to be protected -->
    <url-pattern>/jsp/security/protected/*</url-pattern>
    <!-- If you list http methods, only those methods are protected -->
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
  </web-resource-collection>
  <auth-constraint>
    <!-- Anyone with one of the listed roles may access this area -->
    <role-name>tomcat</role-name>
    <role-name>role1</role-name>
  </auth-constraint>
</security-constraint>

<!-- Default login configuration uses form-based authentication -->
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>Example Form-Based Authentication Area</realm-name>
  <form-login-config>
    <form-login-page>/jsp/security/protected/login.jsp</form-login-page>
    <form-error-page>/jsp/security/protected/error.jsp</form-error-page>
  </form-login-config>
</login-config>


<!-- Security roles referenced by this web application -->
<security-role>
  <role-name>role1</role-name>
</security-role>
<security-role>
  <role-name>tomcat</role-name>
</security-role>

<!-- Environment entry examples -->
<!--env-entry>
  <env-entry-description>
    The maximum number of tax exemptions allowed to be set.
  </env-entry-description>
  <env-entry-name>maxExemptions</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>15</env-entry-value>
</env-entry-->
<env-entry>
  <env-entry-name>minExemptions</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>1</env-entry-value>
</env-entry>
```



```
</env-entry>
<env-entry>
  <env-entry-name>foo/name1</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>value1</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>foo/bar/name2</env-entry-name>
  <env-entry-type>java.lang.Boolean</env-entry-type>
  <env-entry-value>true</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>name3</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>1</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>foo/name4</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>10</env-entry-value>
</env-entry>
```

```
</web-app>
```



## Versiehistorie

Versie	Datum	Wijziging
2.0.0	06.12.2004	Geheel aangepast aan nieuwe deploy-structuur
2.0.3	21.11.2005	Bijzonderheden m.b.t. connection pooling toegevoegd in §4.2.1. Optie 'restart' in Unix/Linux-script serverctl toegevoegd.
2.0.4	03.04.2006	§4.4 m.b.t. licentie toegevoegd.
2.0.5	23.01.2007	Links aangepast waar nodig. Licentie- en Java-versieinformatie bijgewerkt.
2.0.6	20.08.2007	In §3.3.3 notitie toegevoegd m.b.t. de Jide-library in High-T.
2.2.0	21.01.2009	Herzien a.d.h.v. huidige High-T-deployment.
2.2.1	26.10.2009	Korte samenvatting toegevoegd en enige correcties aangebracht.
2.2.2	30.09.2010	Reset password toegevoegd
2.2.3	22.11.2010	Argument " <i>lnf=lookAndFeelClass</i> " toegevoegd aan client (§6).
2.3.0	13.09.2011	Enige kleine wijzigingen aangebracht. Webservice-installatie toegevoegd.
2.3.1	29.02.2012	Vereiste software in een aparte sectie aan begin van document vermeld.