

# Aanwinsten - gebruik

## Aanroepen met Java

Het programma is in Java geschreven. Bij compileren worden alle afhankelijkheden ingebouwd in één enkel .jar-bestand.

Starten met Java gaat als volgt:

```
$ java [-Daanwinsten.conf=<pad-naar-extra-configuratie>] \
  -cp <pad-naar-aanwinsten-jar-with-dependencies.jar \
  nl.uvt.aanwinsten.CLI <commando> [<vlaggen en parameters>]
```

Het gebruik van -Daanwinsten.conf is optioneel. Het wordt uitgelegd in document 'configuratie'.

Met -cp <pad-naar-aanwinsten-jar-with-dependencies.jar wordt gezorgd dat het .jar-bestand met de afhankelijkheden in de classpath voor Java komt. Dat bestand kan een naam hebben met een versienummer er in.

nl.uvt.aanwinsten.CLI is de klasse waarvan je de main()-method wilt aanroepen.

De optionele vlaggen en parameters worden hierbeneden uitgelegd.

## Via een script

Het is makkelijker om alle complexiteit in een uitvoerbaar script te stoppen, bijvoorbeeld een shell-script in Linux. Voor tests bij het werken aan de code is er een script genaamd maak-lijsten:

```
#!/bin/sh

java -Daanwinsten.conf=src/main/resources/aanwinsten.conf \
  -cp target/aanwinsten-0.10-jar-with-dependencies.jar \
  nl.uvt.aanwinsten.CLI maak-lijsten $*
```

Is dat uitvoerbaar gemaakt, dan hoeft alleen nog maar ./maak-lijsten te worden opgegeven in die directory om de lijsten van de vorige maand te maken. Er kunnen ook extra parameters worden meegegeven aan dat script. Door de -Daanwinsten.conf=src/main/resources/aanwinsten.conf wordt er voor gezorgd dat de configuratie uit de broncode wordt gebruikt. Verder is er nog het bestand aanwinsten.conf in de directory van het script. Dat zorgt er onder andere voor dat de HTML en PDF worden gemaakt in de target subdirectory.

## Via cron

In het Debian-package wordt ook een job voor cron aangemaakt die - nu nog - elke maandagmorgen wordt afgetrapt. Later wordt dat elke maand een keer.

Inhoud van bestand /etc/cron.d/aanwinsten na installeren van het Debian-package:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Aanwinstenlijsten; max 7 pogingen in de nieuwe maand
30 7 1-7 * * root java -cp /usr/share/aanwinsten/lib/aanwinsten-jar-with-dependencies.jar nl.uvt.aanwinsten.CLI maak-lijsten

# Ad hoc-lijsten; op werkdagen overdag elk uur ff checken
12 6-20 * * 1-5 root java -cp /usr/share/aanwinsten/lib/aanwinsten-jar-with-dependencies.jar nl.uvt.aanwinsten.CLI maak-adhoclijsten
```

Achteraf is daar nog een regel aan toegevoegd om te zorgen dat mails naar de beheerder van het pakket gaan:

```
MAILTO=fred.vos@tilburguniversity.edu
```

Verder kan er bij de implementatie voor worden gekozen dat het programma niet door root, maar een systeem-user wordt gerund. Dan moet ook 'root' in het cron-bestand worden vervangen door die userid.

In **Test** moeten de aanwinstenlijsten op een ander tijdstip worden gedraaid als *Productie*\*. Anders kan er een conflict komen in de verbinding met de OCLC-server. Pas dus op **Test** die tijd aan. Vervang bijvoorbeeld '30 7 1-7 \* \*' door '30 9 1-7 \* \*'. Aangezien **Test** maar af en toe relevant is, zouden beide cron-regels ook kunnen worden uitgecommentarieerd. De ontwikkelaar dient dan via sudo dan wel de mogelijkheid te hebben om 'met de hand' de commando's uit te voeren.

## Vlaggen en parameters

Alle vlaggen en parameters zijn optioneel. Er zijn soms wel eisen aan, waar het programma niet op controleert, zoals dat de aanwezigheid van --year= vereist dat er ook een --month= is.

---

**-v**

Vraag om veel meer logging.

## Commando 'maak-lijsten'

Maak de aanwinstenlijsten. Zonder verdere parameters zijn dat de lijsten over de vorige maand, gebaseerd op Marc-bestanden die van de FTP-server van OCLC worden opgehaald.

**--year=yyyy --month=mm**

Geef hiermee op dat je bestanden van een andere maand wilt verwerken dan die van de vorige maand. Geef jaar altijd op als vier cijfers, dus bijvoorbeeld 2016. Geef maand altijd op als twee cijfers, dus met voorloop-0 van januari t/m september: 01-12 dus. Bij gebruik van --files= dienen deze alleen om de juiste maand in de titels van de documenten te krijgen.

**--files=bestand[+bestand[+...]]**

Haal bestanden niet op via FTP bij OCLC maar verwerk de opgegeven lokaal opgeslagen bestanden. Geef de reeks bestanden op gescheiden door '+'-tekens. Voor meerdere bestanden was het gebruik van --file=<bestand> --file=<bestand> beter geweest. Het zou eigenlijk ook moeten werken, maar deed dat niet. Daarom deze lelijke oplossing met de plus-tekens. Spaties in de bestandsnaam of in het pad geven vast problemen. Alle bestanden samen worden beschouwd als behorende bij één collectie FILES. Bestanden waar het woord 'scripties' **NIET** in de naam voorkomt, worden verwerkt als boeken. Bestanden waar het woord 'scripties' **WEL** in de naam voorkomt, worden verwerkt als scripties.

**Let op:** Gebruik --files= meestal in combinatie met --year= en month= zodat er een nette titel in de tekst staat.

## Commando 'maak-adhoc-lijsten'

Maak de nieuwe ad hoc lijsten. Dit commando kent geen eigen parameters. Ad hoc lijsten worden gemaakt in een in de configuratie opgegeven directory. Dit commando controleert alle subdirectories in die directory en controleert per directory of het een ad hoc directory is en of deze directory nieuw lijkt. Als de directory nieuw lijkt, worden ad hoc lijsten gemaakt. In document 'werking' wordt uitgelegd hoe wordt vastgesteld of de directory nieuw lijkt en ook hoe het programma precies werkt.

## Commando 'download-marcbestand (dm)'

Haal een Marc-bestand op van de FTP-server van OCLC en sla deze op als een lokaal bestand, bijvoorbeeld voor inspectie, tests.

### **--remote-path=pad**

Verplichte parameter. Geef het pad op op de FTP-server. In de log van commando 'maak-lijsten' zie je deze gemeld worden bij het ophalen van de Marc-bestanden.

### **--save-as=bestand**

Verplichte parameter. Sla het opgehaalde bestand op als dit bestand.

## Commando 'dump-marcbestand'

Dump de inhoud van een Marc-bestand naar het scherm. Records worden voorafgegaan door het record-nummer tussen haakjes. Handig voor debuggen, inspecties.

### **--file=bestand**

Verplichte parameter.

## Commando 'print-manifest (pm)'

Print relevante data over het programma, zoals versie en bouw-tools. Handig voor rapportage van bugs, zodat programmeur precies het artefact kan herleiden.