HelloID Training

Provisioning Connectoren

V2 / 28-04-2025

Doel van de training

- **Zelfstandig werken** met de HelloID Provisioning-module volgens best practices
- **Focus op het bouwen** van eigen provisioning-connectoren met PowerShell

Na afloop van de training kun je

- De volledige provisioningflow configureren van bron tot doel
- Bronsystemen ontsluiten via PowerShell (zoals CSV-bestanden en API's)
- Doelsystemen aansturen met eigen PowerShell-scripts (Create, Update, Delete)
- Field mappings toepassen, inclusief complexe JavaScript-logica voor dataconversie en naamconventies
- Correlatie instellen om bestaande accounts te herkennen
- Slimme configuratieformulieren maken voor connectorparameters
- Snapshots en raw data gebruiken voor importanalyse
- Problemen oplossen met preview en audit logging

Voor wie is deze training bedoeld?

Deze training is geschikt voor jou als je:

- HelloID beheert of binnenkort gaat beheren
- Provisioningkoppelingen wilt bouwen of beheren
- Technisch verantwoordelijk bent voor geautomatiseerd accountbeheer
- Consultant of implementatiepartner bent van HelloID

Je hebt:

- Ervaring met HelloID (zoals behandeld op Dag 1)
- Basiskennis van PowerShell (vereist voor Dag 2)
- Enige ervaring met JavaScript (pre, voor complexe field mappings)

Samengevat

- Je wilt zelf provisioningconnectoren in HelloID kunnen ontwerpen, testen en beheren

Agenda

Deel 1 Bronsystemen:

- Toevoegen van PowerShell-bronsysteem
- Raw data, field mapping en snapshots (incl. complexe mapping met JavaScript)
- Thresholds en gegevensvalidatie

Deel 2 Doelsystemen:

- Ontwerp en configuratie van een PowerShell-doelsysteem
- Field mapping & inputformulier
- PowerShell-scripts
  (Create / Update / Delete, incl. correlatie en testen)

Voordat we verder gaan

Nodig voor deze training

- Training dag 1 en de bijbehorende oefeningen zijn afgerond.
- HelloID Agent is geïnstalleerd, up-to-date en de services zijn gestart.
- Problemen met de agent? Volg dan Lab 0.

PowerShell Bronsystemen

PowerShell Source systems

- Inleiding
- Toevoegen van het bronsysteem
- On-premise vs. cloud agents
- Flow van de bronconnector
- Personscript
- Departmentscript
- Configuratieformulier
- Raw data en mapping
- Imports & preview
- Thresholds en blocked persons
- Snapshot vernieuwen

Bronsystemen en PowerShell-connectoren in HelloID

leveren informatie over medewerkers, contracten en beheren van accounts, toegang en rechten

Komt niet altijd uit HR-systemen, maar ook:

- Interne systemen
- Platte bestanden (zoals CSV)
- SQL-databases
- API-webservices

Tijdens de training behandelen we:

- Hoe je een PowerShell-connector configureert in HelloID
- Voorbeeld: een CSV-dataset met gebruikers en afdelingen als bron

Source connector - Toevoegen

PowerShell-source toevoegen in HelloID

- Gebruik de **Source Template connector** uit de connectorencatalogus
- gevuld vanuit onze **GitHub-repositories**
- Stel een **eigen naam en beschrijving** in na het toevoegen

Agentkeuze

- Lokale agent nodig bij:
- Gebruik van niet-standaard PowerShell-modules (Import-Module)
- Toegang tot interne netwerkresources (bijv. lokaal SSL-certificaat)
- Cloud agent gebruiken in alle andere gevallen

Bronconnectoren - on-premise en cloud

On-premise PowerShell agent

- Zet "Execute on-premises" aan
- Scripts draaien lokaal via de HelloID agent
- Vereist een lokaal geïnstalleerde agent
- Werkt met PowerShell 5.1 of hoger

Cloud PowerShell agent

- Zet "Execute on-premises" uit
- Scripts draaien via de HelloID cloud
- Geen lokale agent nodig
- Beperkt tot PowerShell Core 7
- Geen ondersteuning voor extra modules

Source connector - Flow

Tijdens de import van een bronsysteem doorloopt HelloID de volgende stappen:

- **PowerShell** – Ophalen van persoons- en contractgegevens via PowerShell-scripts
- **Raw Data** –Gegevens zichtbaar op de *Raw Data*-tab
- **Mapping** – Ruwe data koppelen aan juiste velden in HelloID
- **Snapshot** – Een momentopname van: persoonsgegevens, bronconfiguratie, primaire contracten, managers, HelloID display name en aggregatie-instellingen.
- **Personen** – Tijdens het maken van de snapshot worden nieuwe persoonsrecords aangemaakt, bestaande records bijgewerkt of verwijderd

Persons importscript - structuur

- Levert persoonsgegevens aan HelloID via PowerShell.
- Het script retourneert een hashtable waarin elke sleutel een persoon is, met bijbehorende contracten.
- Structuur:

Tip:

Gebruik: ConvertTo-Json -Depth 10 als het object diepere lagen heeft

```
$persons = @(
    @{
        ExternalId  = "12345"
        DisplayName = "Angie van den Hoeger"
        Contracts   = @(
            @{
                ExternalId    = "C-001"
                StartDate     = Get-Date("2018-01-01") -Format "o"
                EndDate       = $null
            } # , @{Meer contracten}
        )
    } # , @{Meer personen}
)
```

Person script - vereiste velden (Person)

Persoon:

- ExternalId (unieke ID)
- DisplayName
- Contracts (array van contracten)

Optioneel / aanbevolen:

- Naamvelden (roepnaam, achternaam, voorvoegsels)
- Name.Convention (Naamconventiecode)
- Geboortedatum, geslacht, initialen (voor persoonsaggregatie)

Please check out the references on Persons import script
Manuals for Administrators → Provisioning → Configure a Custom PowerShell source system

Person import script - vereiste velden (Contract)

Contract:

- ExternalId (per contract uniek)
- StartDate
- EndDate (mag leeg zijn)

Optioneel:

- Afdelingscode
- Functiecode / omschrijving
- FTE

Departments import script

Afdelingsinformatie ophalen in HelloID:

- Geen mapping nodig
- Wordt gebruikt voor afdelingsselectie in business rule-condities

Verplichte velden per afdeling

- ExternalId – Unieke afdelingscode
- DisplayName – Naam van de afdeling

Optionele Velden

- ManagerExternalId – ExternalId van de afdelingsmanager
- ParentExternalId – Bovenliggende afdeling

Deze velden worden gebruikt als “Primary Manager” staat op:From department of primary contract

Let op: slechts één manager per afdeling mogelijk

Department import script

- levert een lijst van afdelingen aan HelloID.
- worden gebruikt in business rules (bijvoorbeeld voor filtering of managerkoppelingen).
- Structuur:

```
$departments = @(
    @{
        ExternalId        = "ICT"
        DisplayName       = "Informatievoorziening"
        ManagerExternalId = "12345"
        ParentExternalId  = "ORG"
    } # , @{Meer afdelingen}
)
```

- Let op: de manager- en parentvelden alleen gebruiken als “Primary Manager” op “From department of primary contract” staat ingesteld.

Technische tips bij het bouwen van een source connector

- Output naar HelloID in JSON-formaat (let op –depth parameter)
- Output per object in plaats van alles in één keer:

```
ForEach ($person in $persons) {
    Write-Output ($person | ConvertTo-Json)
}
```

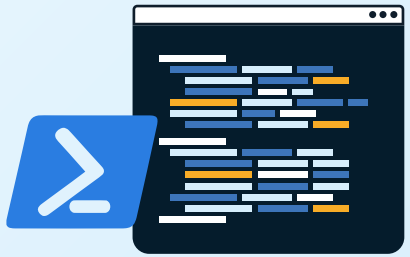
- Zorg voor duidelijke logging (maximaal 100 regels per import, de rest wordt afgekapt)
- Maak goed gebruik van try/catch en plaats duidelijke foutmeldingen. Bedenk dat jij misschien niet degene bent die het dagelijkse beheer uit gaat voeren
- Stuur de data zo zuiver mogelijk naar HelloID. Pas zo min mogelijk conversies en herstructureringen toe. Mapping van data doe je in HelloID zelf met de **Person** en **Contract** mapping

Configuratieformulier in PowerShell Scripts

- **Doel:** Inputparameters invoeren en opslaan (zoals API-sleutels, wachtwoorden, URL's)
- **Voordeel:** Veiliger & praktischer dan hardcoded waarden

Werking:

- Maak een aangepast formulier voor parameters
- Verkrijg de waarden binnen **$configuration**



Configuratieformulier in PowerShell Scripts

Sjabloon via JSON Editor

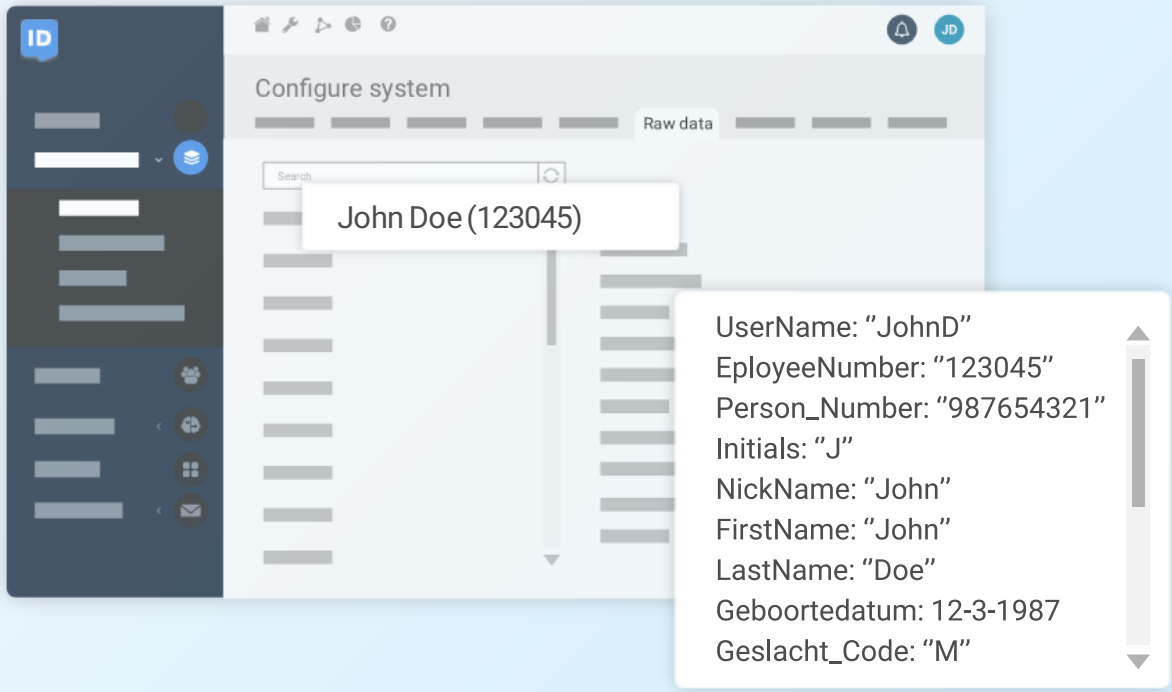
- Ondersteunde elementen:
  - Tekst
  - Multi-line tekst
  - Wachtwoord
  - E-mail
  - Toggle
  - Radioknop
  - Dropdownmenu
- **Pas aan** op basis van de behoeften van je script

Raw data

Raw Data-tab in HelloID

- Toont **onbewerkte gegevens** uit het bronsysteem
- **Geen modificaties, filters of mappings** toegepast
- **Nuttig voor troubleshooting,** waar je de data in onbewerkte staat kunt bekijken
- Selecteer een persoon om de **raw data in detail** te bekijken

Raw data is beschikbaar na de eerste succesvolle import

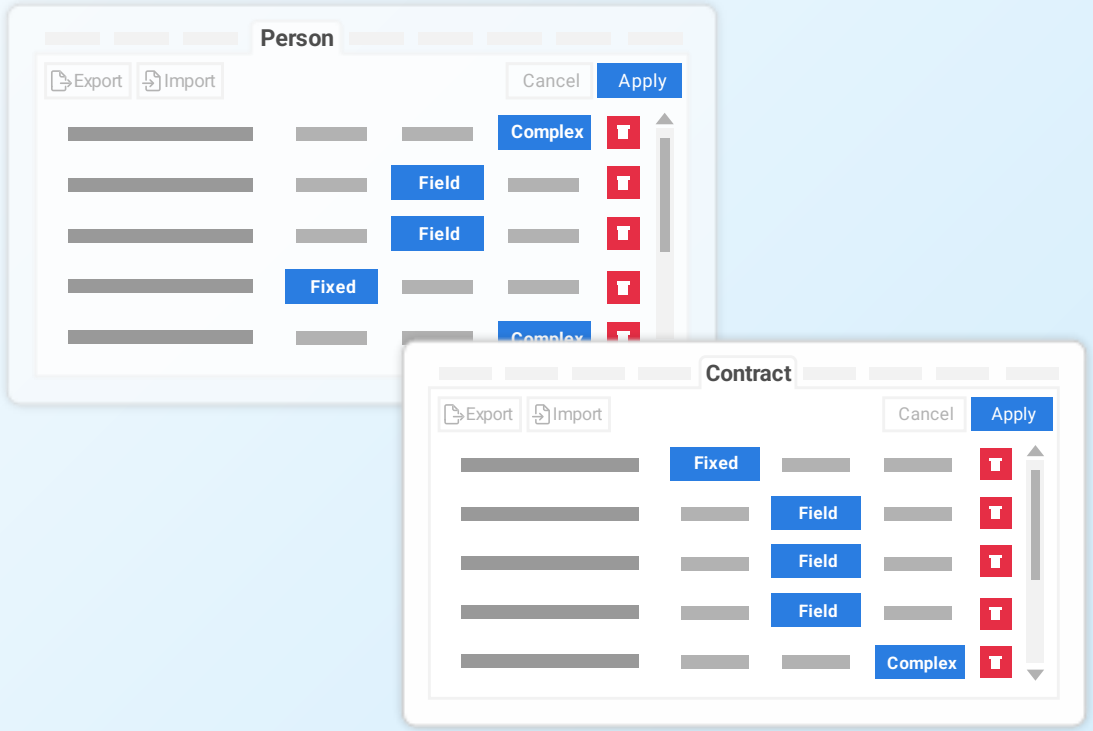


Field mapping - Algemeen

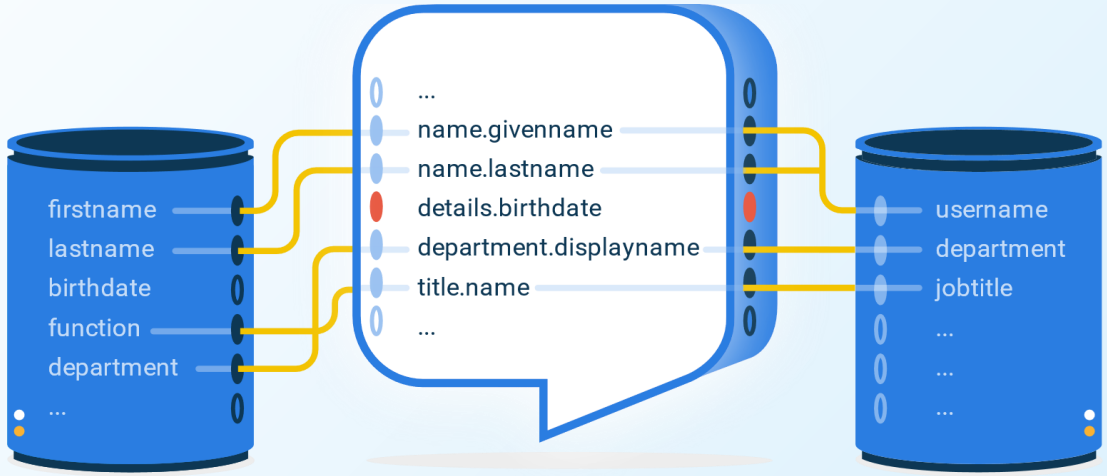
- Wat is field mapping?
- Flow: Raw data → Mapping → HelloID
- Typen mapping: Fixed, Field, Complex (JavaScript)

Tips:

- Mapping aanpassen na toevoegen van scripts en ophalen van ruwe data
- Mapping vertaalt ruwe data naar HelloID-model
- Controleer mapping vóór volledige import
- Gebruik "New snapshot" om mapping toe te passen op de ruwe data



Mapping van persoons- en contractgegevens



Persoonsgegevens die je vaak mapt:

- NickName → Field
- FamilyNamePrefix → Field
- FamilyName → Field
- Name.Convention → Complex

Belangrijk:

- Gebruik source. om persoonsvelden aan te roepen in complexe mappings
- Custom fields moeten eerst worden aangemaakt in de source-configuratie

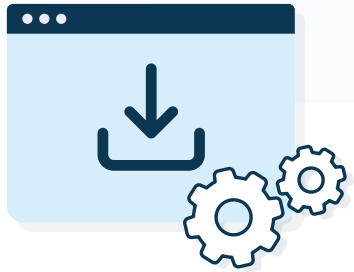
Contractgegevens die je vaak mapt:

- StartDate → Field
- Department.ExternalId → Field
- Title.ExternalId → Field
- Title.Code → Field
- FTE → Field

Belangrijk:

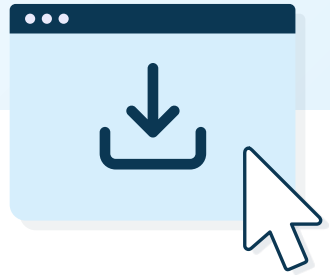
- Gebruik sourceContract. om contractvelden aan te roepen in complexe mappings

Imports



Automatische import

- Selecteer een **uur van de dag**, HelloID bepaalt het exacte tijdstip binnen dat uur
- Na import worden **Person records geëvalueerd** en optioneel toegepast op business rules voor de doelsystemen
- Max drie keer per dag, minimaal 2 uur tussen elke geplande import
- rules op doelsystemen



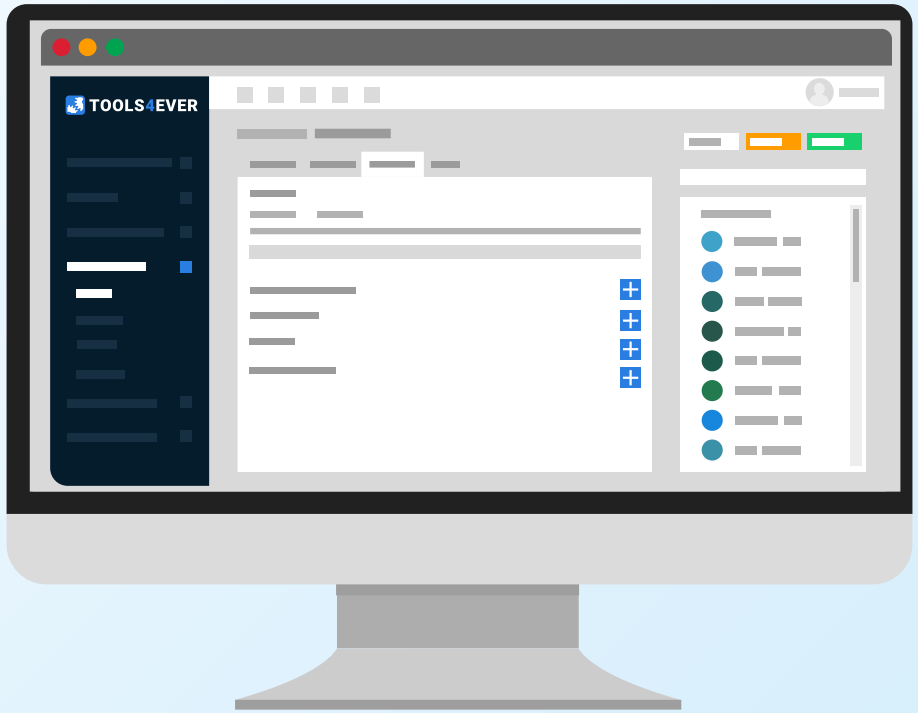
Handmatige import

- Alleen raw data wordt opgehaald en gemapt
- Resultaat wordt verwerkt in een nieuw snapshot
- Geen toepassing van business rules op doelsystemen

Lab 1 – Een PowerShell-bronsysteem maken

- Wat ga je doen in dit lab?
- - Je voegt een PowerShell-bronsysteem toe in HelloID
- - Je configureert de scripts en de mapping voor personen en afdelingen

- Duur van dit lab: X minuten



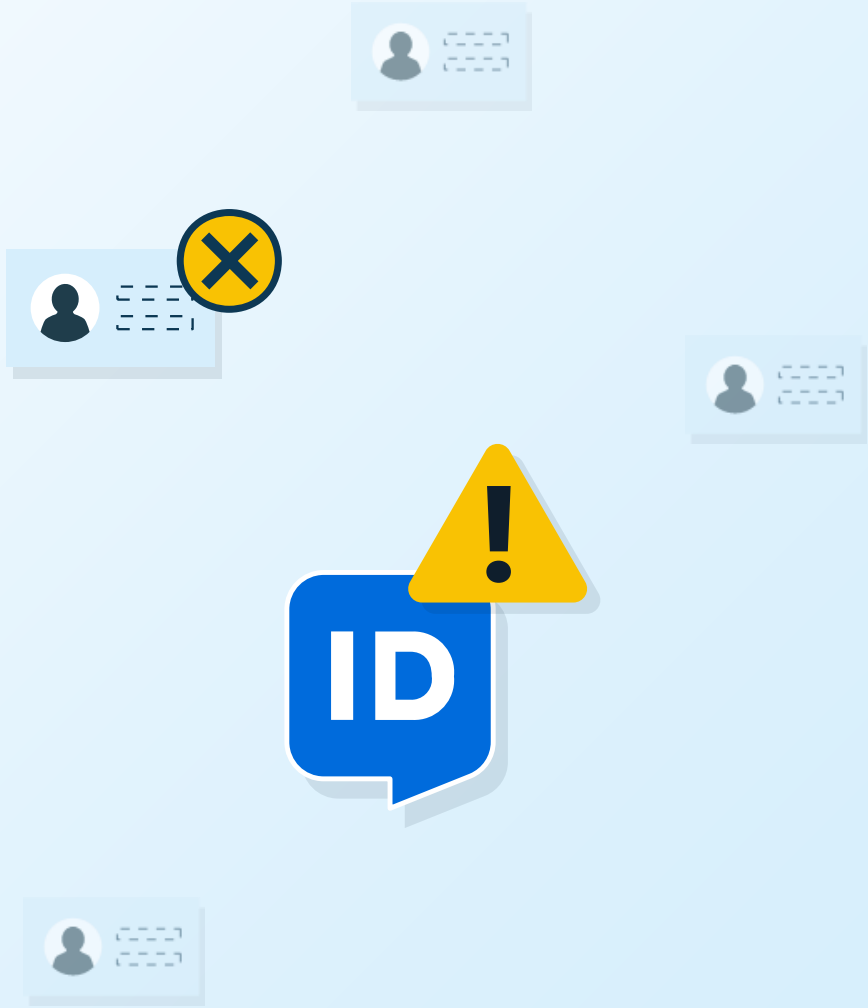
Thresholds en blocked persons

Automatisch blokkeren van imports

- Voorkomt bulktoevoeging of -verwijdering van personen
- HelloID vergelijkt nieuwe data met vorige import

Controle op:

- Aantal toegevoegde/verwijderde personen
- Lege verplichte velden (‘required this field’ staat aan)



Thresholds en blocked persons

Blokkering bij overschrijding:

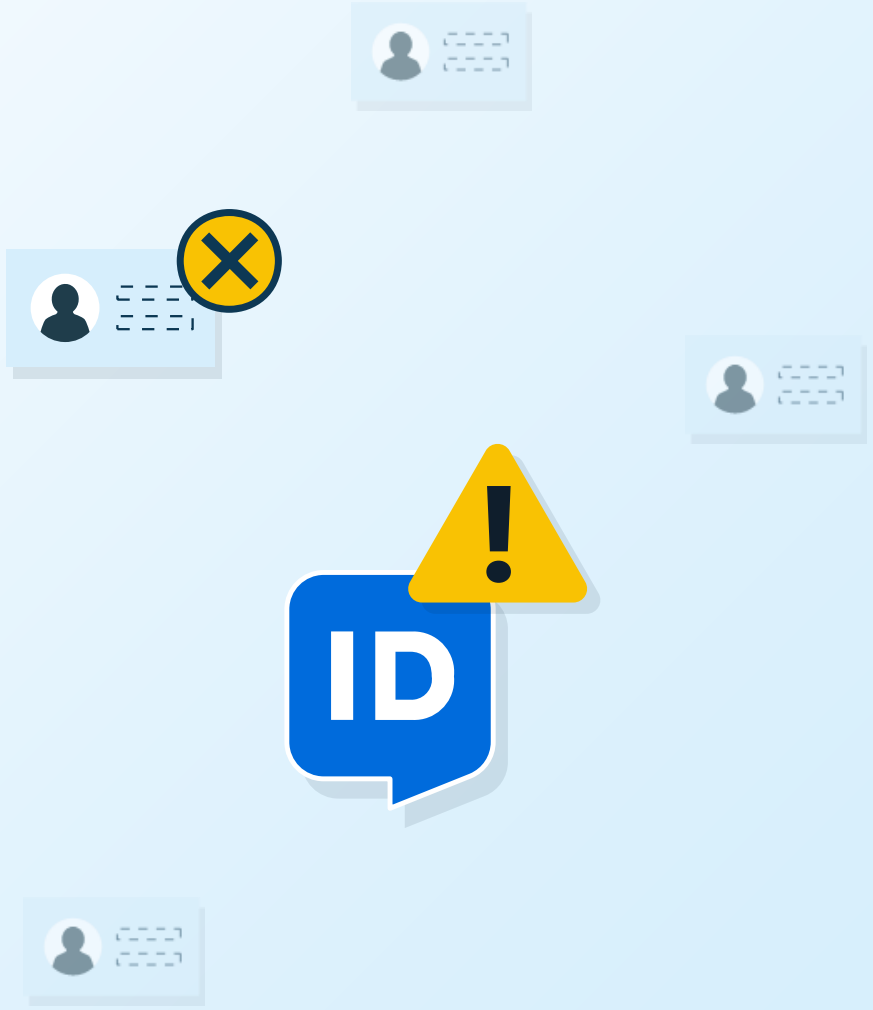
- Bij overschrijding wordt de import geblokkeerd tot handmatige goedkeuring
- Niet overschreden? → Alleen geblokkeerde personen worden overgeslagen, rest wordt geïmporteerd

Voordelen:

- Voorkomt fouten door onjuiste of ontbrekende data uit het bronsysteem

Standaard ingeschakeld:

- Threshold voor verwijderen van personen staat standaard op 1 bij een nieuw bronsysteem



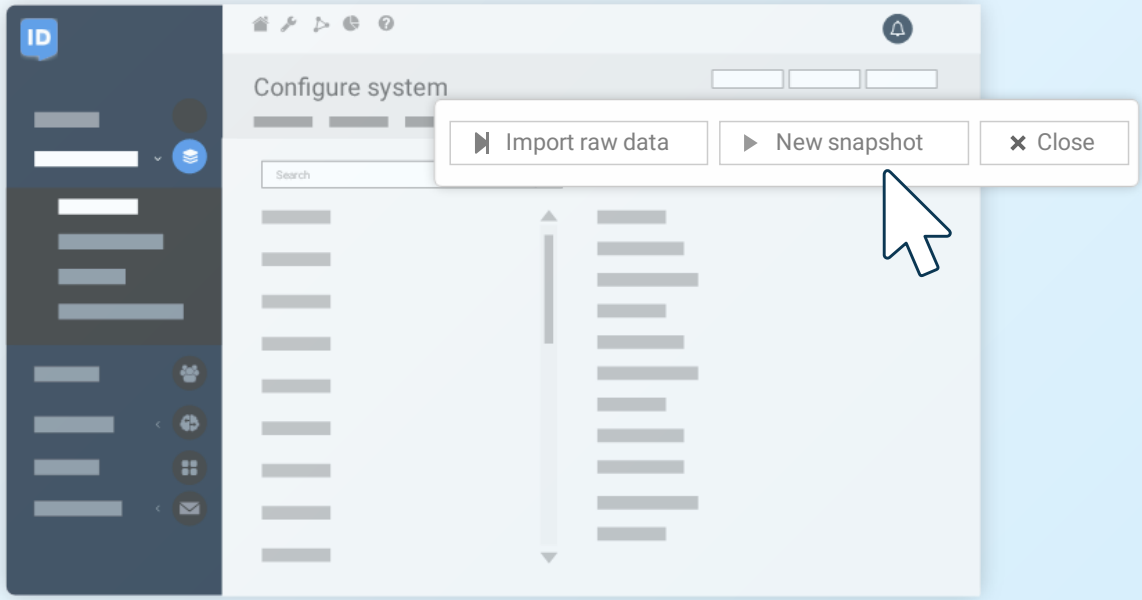
Snapshot – Wijzigingen in configuratie

Wanneer de configuraties van het bronsysteem worden gewijzigd, worden deze veranderingen niet direct doorgevoerd in het huidige vault-snapshot.

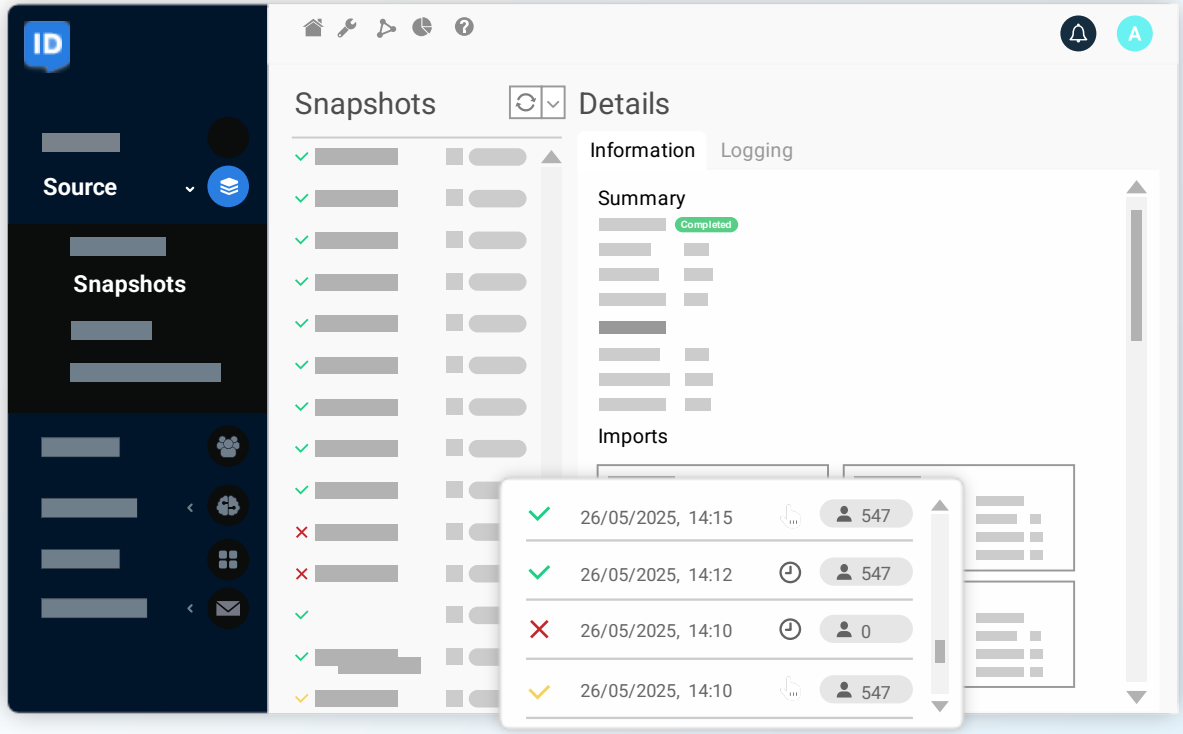
- **Voorbeeld:**
  Aanpassingen zoals de weergavenaam, primaire contractbepaling, managerbepaling, fieldmapping of powershell scripts worden niet onmiddellijk bijgewerkt in de snapshot en de personenlijst.

Snapshot - Vernieuwen

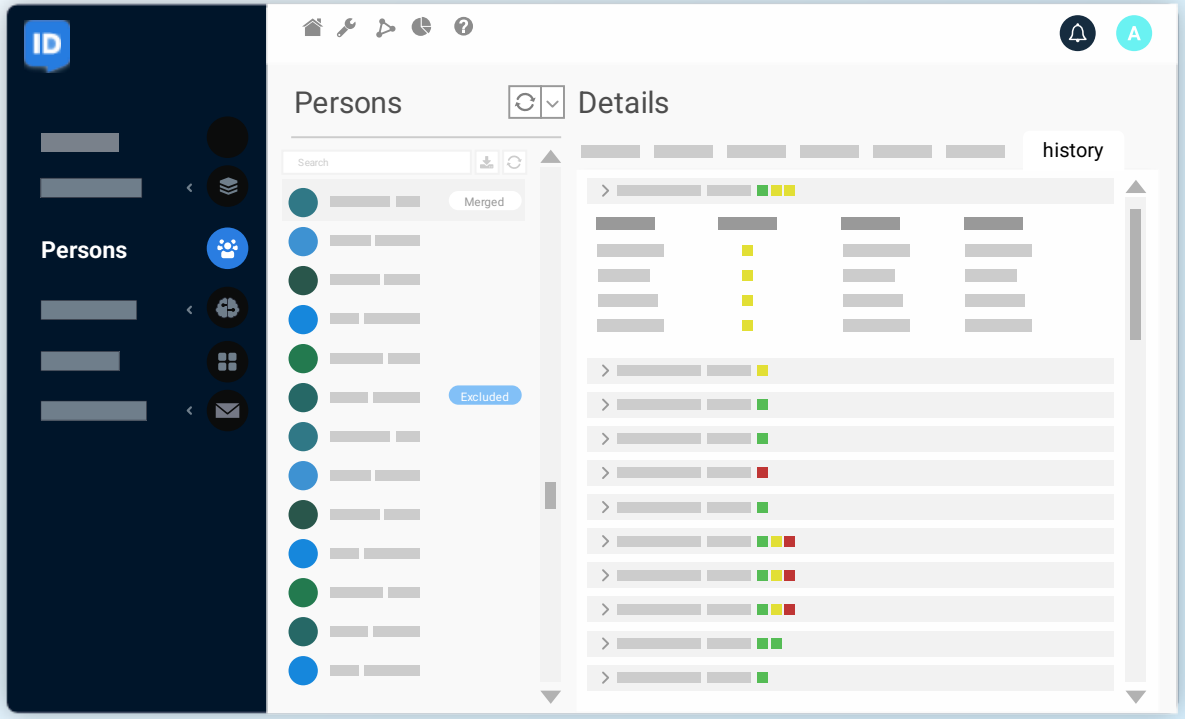
- Gebruik vault-snapshot om wijzigingen toe te passen en zichtbaar te maken
- De **New snapshot**-knop ververst de huidige vault-snapshot en past alle uitstaande wijzigingen toe zonder nieuwe gegevens op te halen uit het bronsysteem
- Let op: gegevens van het bronsysteem/PowerShell scripts veranderd? nieuwe volledige import vereist
- Handig bij configuratie aanpassingen zonder nieuwe gegevens van het bronsysteem te importeren



Snapshot - Vernieuwen



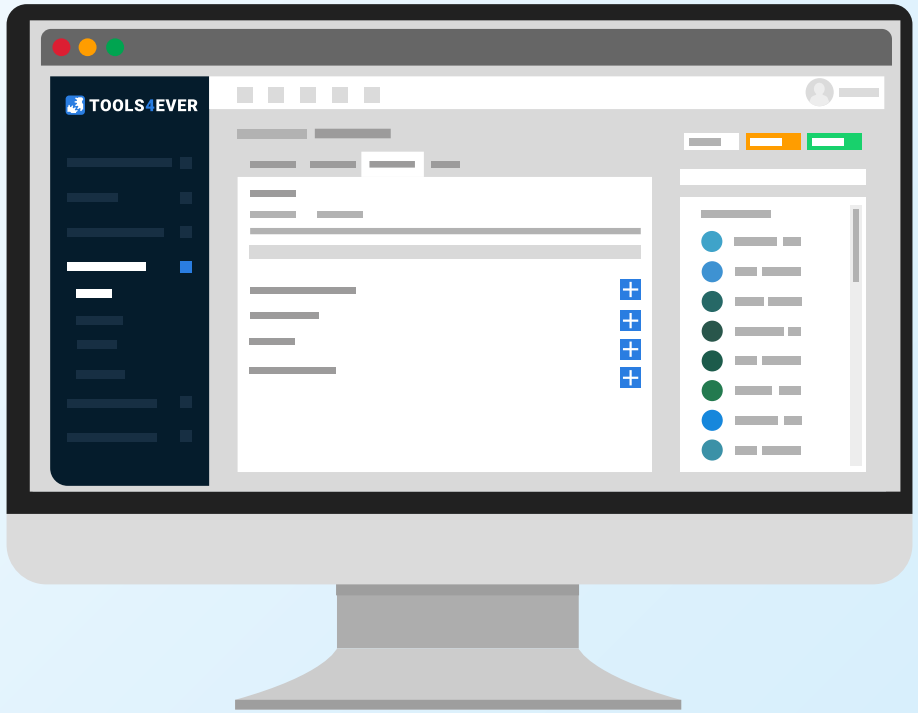
Tip: Controleer het effect via de snapshot history



Lab 2 – Gegevensvalidatie en thresholds voor bronsystemen

- Wat ga je doen in dit lab?
- - Je stelt thresholds in om grote fouten te voorkomen
- - Je leert hoe je verplicht veldgebruik valideert en blokkeert

- Duur van dit lab: X minuten



PowerShell Doelsystemen

PowerShell Doelsystemen

- Inleiding
- Toevoegen van een PowerShell-doelsysteem
- Account lifecycle
- Field mapping
- Configuratieformulier
- Permissions en Resources
- Correlation
- Scripting (Create, Update, Delete, Etc)
- Input/output en logging
- Audit logs

Inleiding

- Doelsystemen zijn de systemen waarin de provisioning-processen van HelloID worden uitgevoerd. Ze zijn verantwoordelijk voor het toekennen en intrekken van accounts, toegang tot accounts en rechten

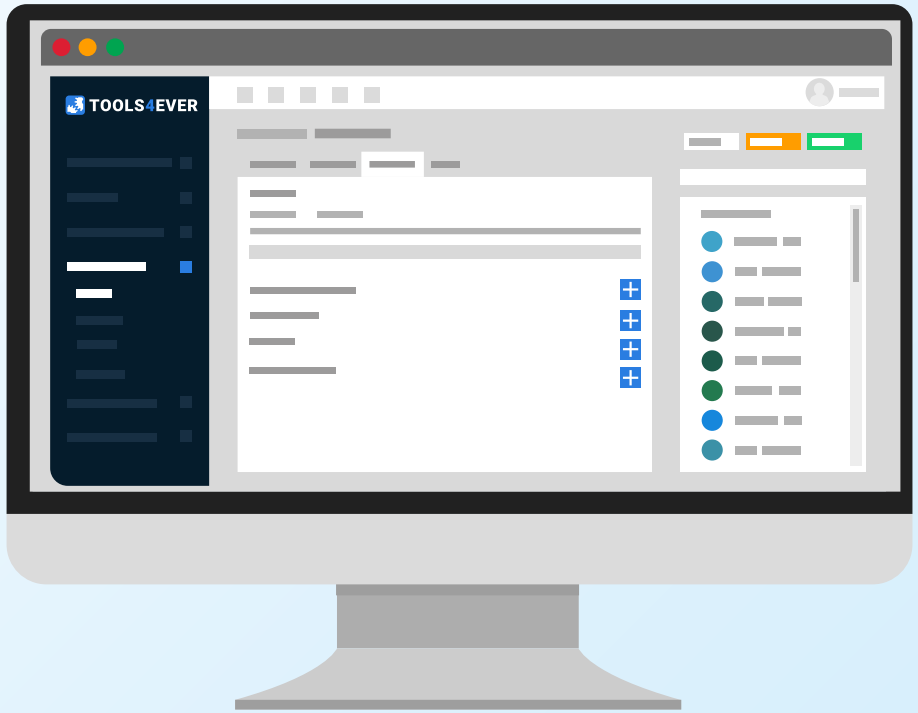
- In dit deel worden de configuratie- en beheermogelijkheden behandeld

Introductie op Lab 3 – De casus

De casus: Intranetkoppeling maken

Je organisatie gebruikt een zelfgebouwd intranet.

Jij gaat een koppeling maken voor het automatisch aanmaken, updaten en verwijderen van accounts.



Je gaat in Lab 3 aan de slag met:

- Het maken van een ontwerp voor het doelsysteem
- Het opstellen van een plan van eisen
- Het bepalen van benodigde velden en data
- Het uitwerken van een plan van aanpak
- *(Dit lab doen we klassikaal. De casus en alle details staan ook in het labdocument.)*

Plan van Aanpak

Hoe pakken we het project aan?

- Maak het ontwerp compleet en stel een actieplan op (Lab 3)
- Voeg de connector toe in HelloID (Lab 4)
- Stel de field mapping in (Lab 5)
- Maak het invoerformulier voor de configuratieparameters (Lab 6)
- Configureer de correlatie actie in het create-script (Lab 7)
- Configureer de create actie in het create-script (Lab 7)
- Configureer het update-script (Lab 8)
- Configureer het delete-script (Lab 9)

Technisch overzicht HelloID PowerShell-doelsysteem

- Een HelloID doelsysteem bestaat uit verschillende onderdelen
- Account lifecycle
- Mapping
- Account (Create, Enable, Update, Disable, Delete Scripts (en Import))

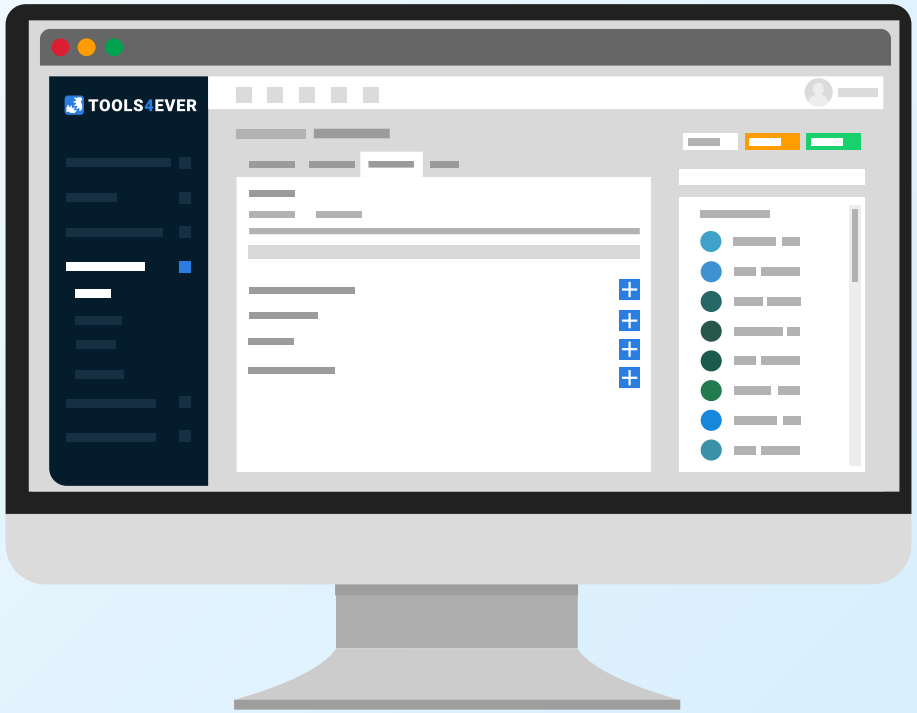
  Permissies
- Normaal (List, Grant, Revoke, (optioneel) Update, Import)
- Dynamisch (List, subPermissions, Import)
- Resource
- Script
- Correlatie
- Configuratie
- Concurrent sessions

- Focus van vandaag ligt op Account lifecycle en de Configuratie

Lab 3 – Ontwerp maken voor het doelsysteem

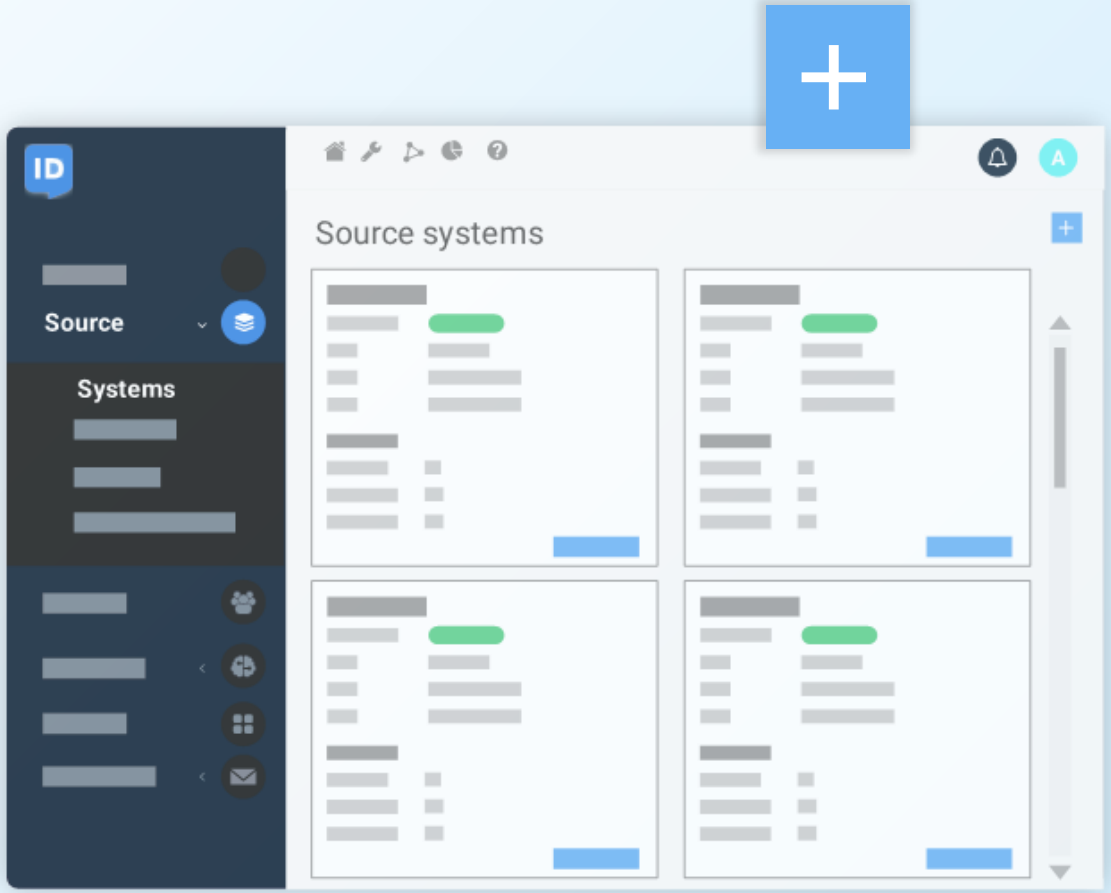
- Wat ga je doen in dit lab?
- - Je werkt aan een ontwerp voor een PowerShell-doelsysteem
- - Je stelt een plan van eisen en aanpak op voor de connector

- Duur van dit lab: X minuten



Powerhell doelsysteem toevoegen

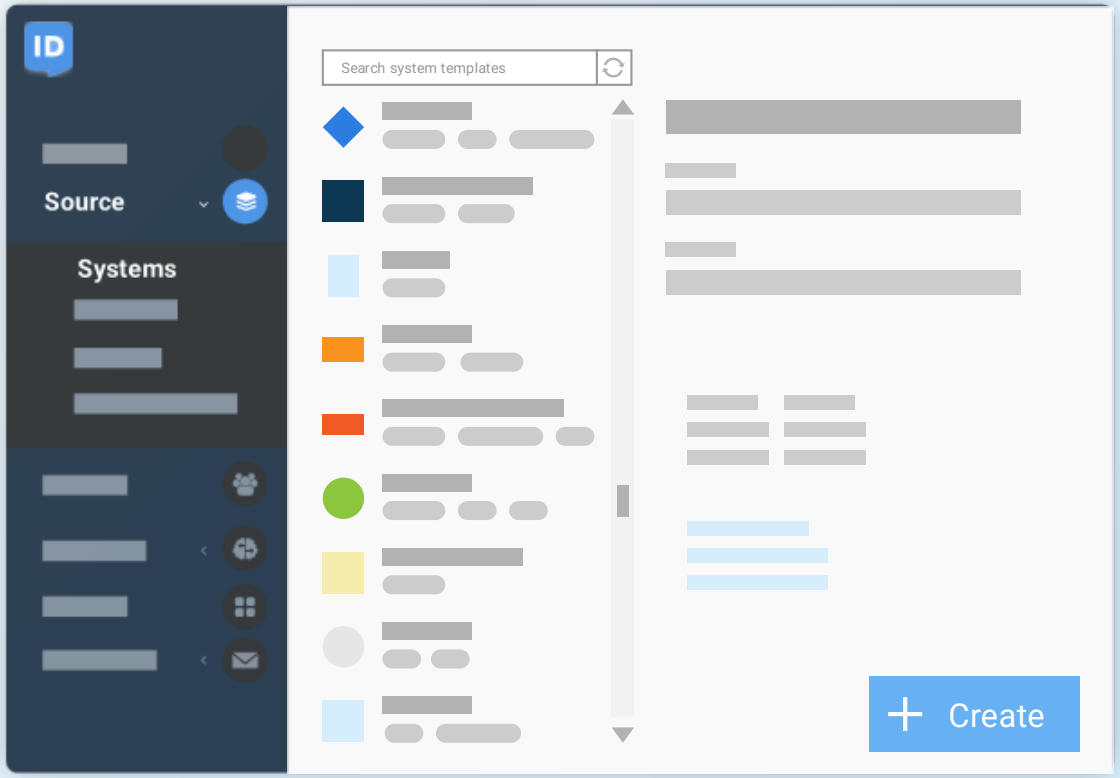
1. via het menu-item **Target -> Systems** in het navigatievenster
2. Voeg een nieuw doelsysteem toe
3. Stel in **hoe HelloID gegevens schrijft** naar het doelsysteem



Powerhell doelsysteem toevoegen

Standaardinstellingen bij toevoegen

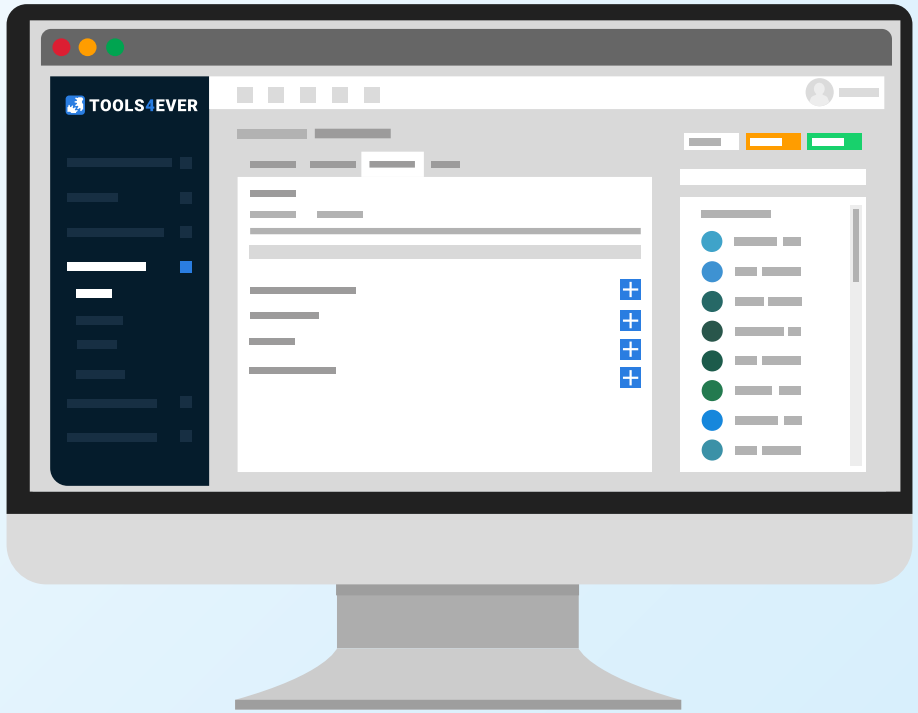
- **"Disable target system"** staat aan:
- Voorkomt uitvoering van acties via business rules
- **Alle thresholds = 1**:
- Beschermt tegen onbedoelde wijzigingen



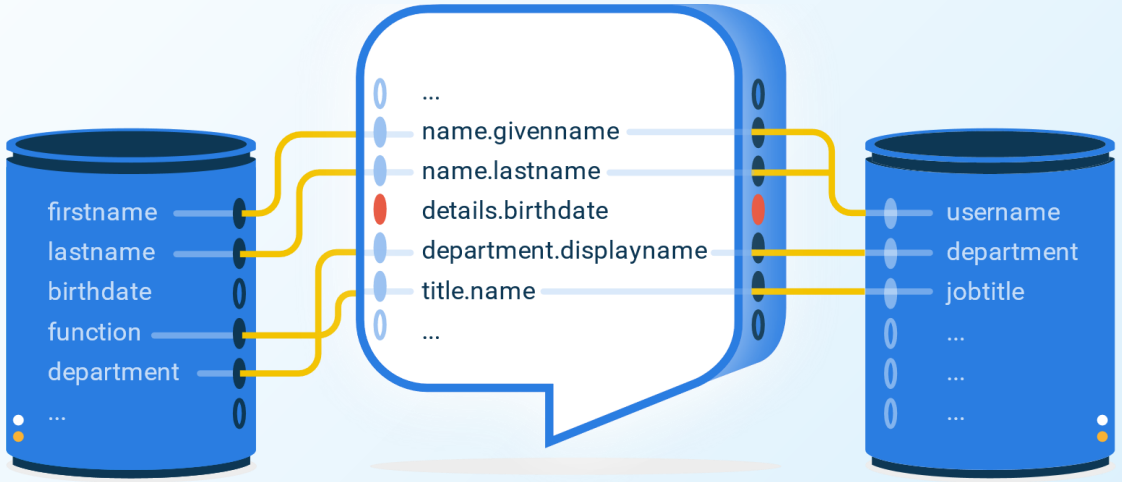
Lab 4 – Doelsysteem toevoegen

- Wat ga je doen in dit lab?
- - Je voegt het PowerShell-doelsysteem toe in HelloID
- - Je stelt de basisinstellingen, thresholds en agentconfiguratie in

- Duur van dit lab: X minuten



Field mapping in een doelsysteem



- Wat doet deze mapping?
  - Verbindt HelloID-gegevens met scriptinput voor Create/Update/... acties
  - Bepaalt welke velden naar het script gaan via $actionContext.Data
- Specifiek voor doelsystemen:
  - Je kunt per actie instellen wat er gebeurt (Create, Update, Delete, etc)
  - Je kunt aangeven:
    - ✓ “Store in account data” → gebruiken in andere doelsystemen
    - ✓ “Use in notifications” → beschikbaar in notificatietemplates
- Tip:
  - Test de mapping in het script met de **previewfunctie** per actie

‘Use account data from other systems’

Waarom gebruik je dit?

- Soms wil je in een tweede doelsysteem data hergebruiken die eerder in een ander systeem is gegenereerd, zoals een gebruikersnaam (SamAccountName).

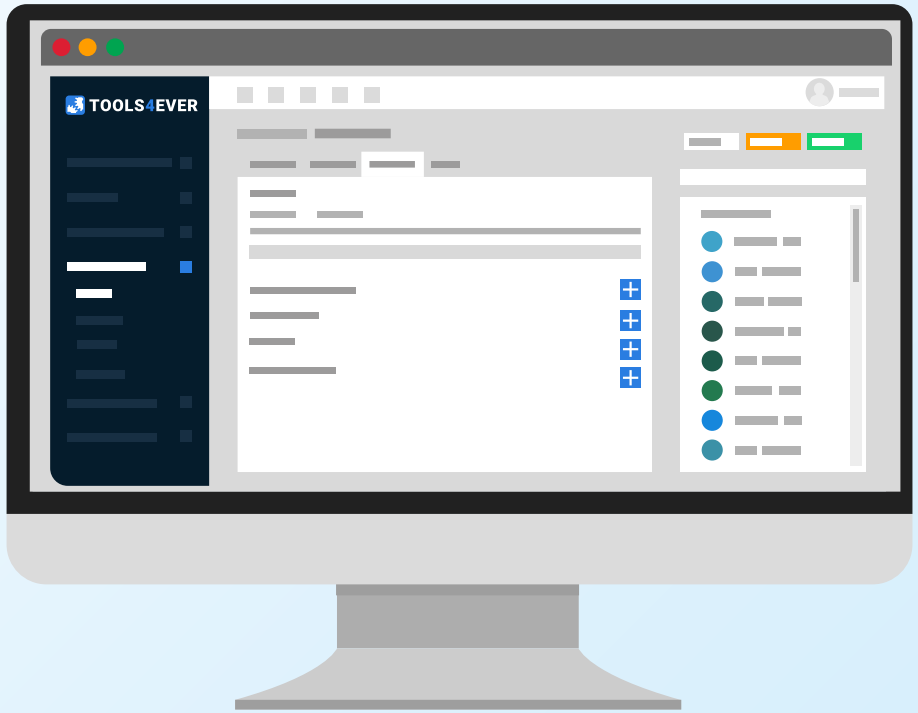
Hoe werkt het – in drie stappen:

1. In het primaire doelsysteem zet je bij de gewenste mapping aan:
   - → “Store in account data”
2. In het secundaire doelsysteem stel je in:
   - → “Use account data from systems”
3. In de mapping van het secundaire doelsysteem haal je het veld op via complex mapping scipts:
   - → Specificeer het veld als: Person.Accounts.MicrosoftActiveDirectory.SamAccountName
   - Waarbij *MicrosoftActiveDirectory* de naam is van het primaire doelsysteem.

Lab 5 – Field mapping

- Wat ga je doen in dit lab?
- - Je stelt de field mapping in op basis van het ontwerp
- - Je test de mapping voor create- en update-acties

- Duur van dit lab: X minuten



Account (lifecycle)

Voordat we een PowerShell-doelsysteem instellen, is het belangrijk om een helder overzicht te hebben van de levenscyclus van een account en de bijbehorende entitlements.

Levenscyclus van een user 'account' entitlement

- **Create** – Een nieuw (disabled) account wordt aangemaakt of bestaand account wordt gecorreleerd
- **Delete** – Het account (of entitlement) wordt verwijderd

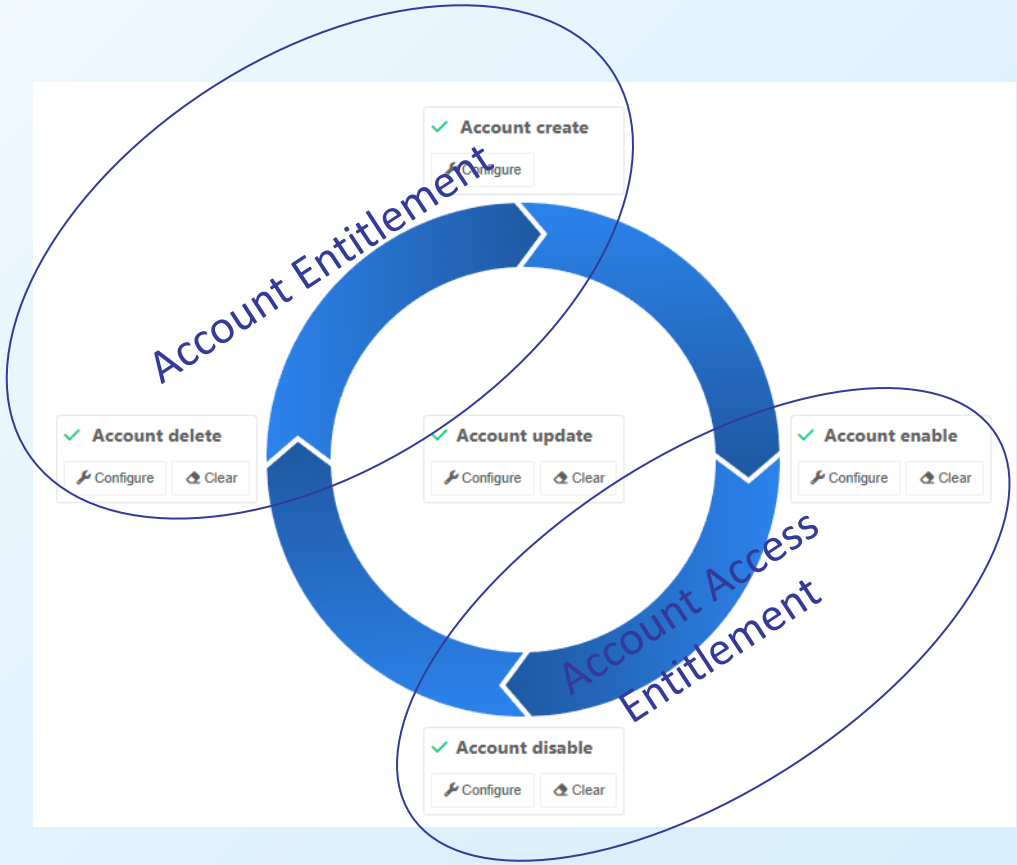
'Account Access' entitlement

- **Enable** – Het account wordt alleen geactiveerd (advies: geen verdere updates)
- **Disable** – Het account wordt alleen gedeactiveerd

Account Update

- Alleen toegekende accounts worden bijgewerkt
- Een update in de snapshot-data leidt tot updates in elk geconfigureerd doelsysteem
- Updates kunnen ook plaatsvinden wanneer een account is uitgeschakeld

Heb je een actie niet nodig: wis het script



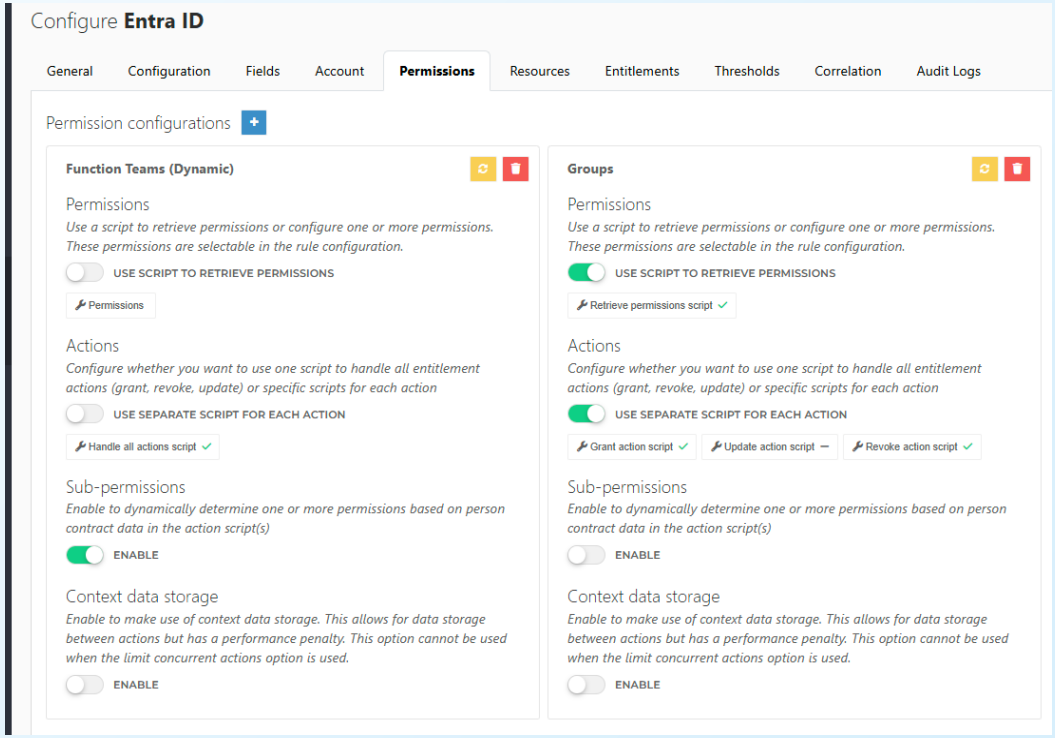
Permissies

Normale Permissie-entitlements

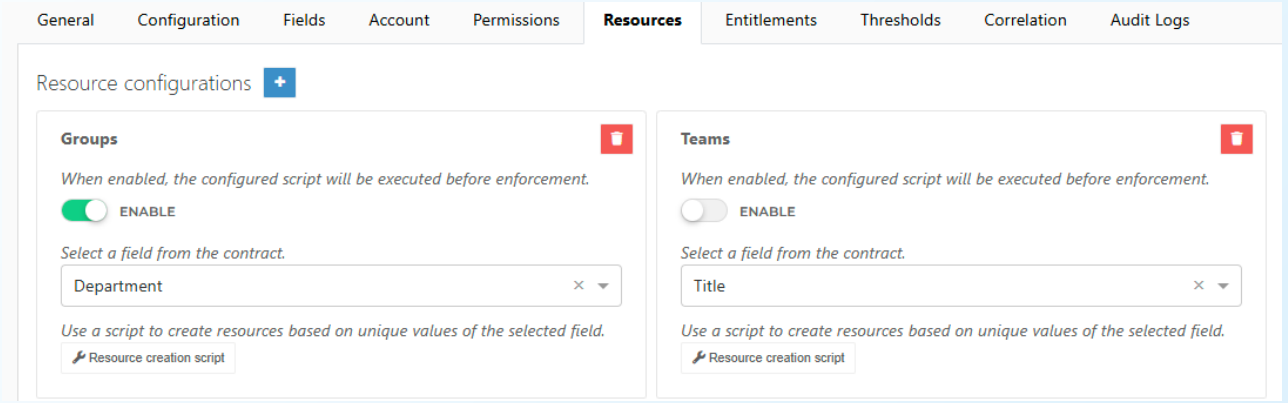
- Retrieve: haalt beschikbare permissies op
- Grant: wijst permissie aan gebruiker toe
- Revoke: neemt permissie van gebruiker af
- Update: wordt bij normale permissies niet gebruikt
- Scripts dialogen bevatten een preview-optie

Dynamische Permissie-entitlements

- Worden gebruikt voor dynamische groepstoewijzing (bijv. afdelingen of functies)
- Retrieve wordt meestal hardcoded gedaan (bijv. Dynamic – Department)
- Maakt gebruik van het ‘all-in-one’ script (subPermissions.ps1 op GitHub)
- Door gebruik sub-permissions aantal benodigde business rules en permissions in je omgeving aanzienlijk verminderen



Resources



Wat zijn Resource scripts?

- PowerShell-scripts die vóór de enforcement fase worden uitgevoerd
- Worden bijvoorbeeld gebruikt om groepen te creëren op basis van contractgegevens

Verschil met Permissions:

- Resources worden niet als 'entitlements' beheerd
- Controle op bestaande resources moet in het resource script gebeuren

Werking:

- Voert acties uit voor unieke waarden in een geselecteerd contractveld of contractobject (bv department.*)
- Let op inconsistente brongegevens om duplicaten te voorkomen
- Scripts hebben een uitvoerlimiet van 10 minuten
- Wordt uitgevoerd voor de enforcement in de schedule, of handmatig via enforcement **+**

Belangrijke aandachtspunten:

- Resources werken alleen met contractvelden, niet met persoonsvelden
- Niet opgenomen in de evaluatie, maar wel testbaar via de preview-functie (max 10 records)
  - Testen beter buiten HelloID
- Resultaten worden vastgelegd in de auditlogs van het systeem en van de enforcement

Functioneel plan Create actie

- Plan maken
- Wat doe je in de create actie ?
- Aanmaken van een account voor een nieuwe medewerker
- Wat nog meer?
- Uitzoeken of er al een account voor de medewerker bestaat
- Dat heet correlatie

Correlatie

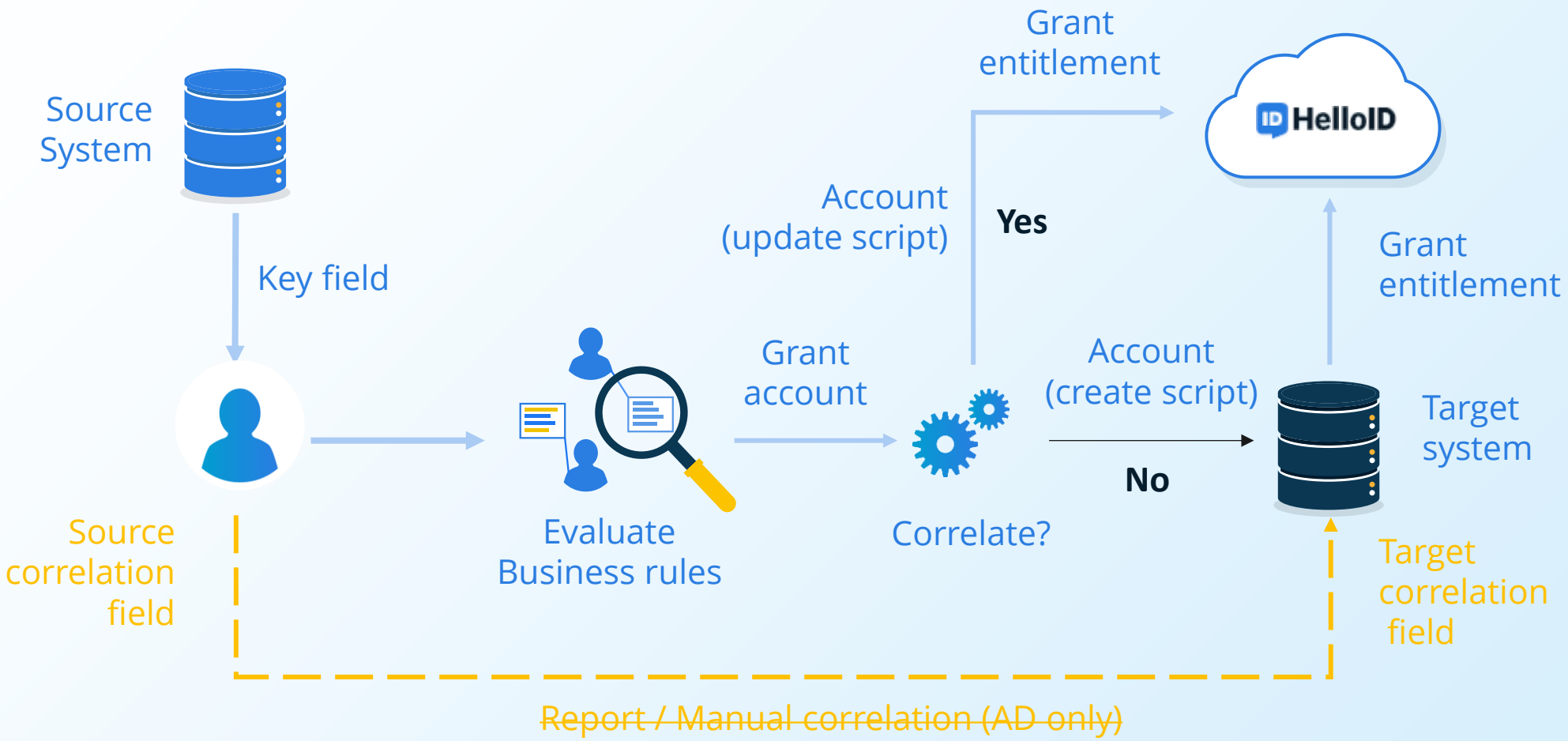
Wat is correlatie?

- HelloID herkent bestaande accounts en voorkomt het aanmaken van dubbele accounts.
- Correlatielogica configureer je in het account create-script én via het tabblad Correlatie in HelloID.

Belangrijk om te weten:

- Handmatige correlatie is niet mogelijk.
- Er is geen standaard correlatierapportage.
- Correlatie-instellingen beschikbaar in $actionContext.CorrelationConfiguration.
- Resultaatvariabelen:
- In create-scripts: $outputContext.AccountCorrelated
- In update-scripts: $actionContext.AccountCorrelated

Correlatie in een Powershell connector



Import van entitlements uit doelsysteem

- HelloID kan bestaande accounts, account access en permissies ophalen uit een doelsysteem.
- Vergelijkt resultaten met de verwachte entitlements in HelloID
- Belangrijk om te weten:

  Alleen beschikbaar voor:

  - on-premises Active Directory
  - PowerShell-doelsystemen.
- Vereist een apart importscript per connector.
- Het doelsysteem moet een geschikt koppelvlak hebben (API, database, etc.).
- Je kunt kiezen om alleen te rapporteren of direct entitlements uit te delen (en zo gebruiken als Correlatierapportage).

Account flow scripts Create (en Update)

1. Ontvang informatie via de $actionContext-variabele
2. Haal de user op uit het doelsysteem in het create script
3. Controleer of de gebruiker bestaat obv de correlatieconfiguratie:
    1. Niet bestaand → Maak gebruiker aan in het het doelsysteem
    2. Wel bestaand → Correleer gebruiker
        1. Zet $outputContext.AccountCorrelated op true → Activeert het update script
4. Stuur informatie terug naar HelloID via $outputContext:
    1. Success → true
    2. Auditlogs: Eén regel per schrijfactie
    3. AccountCorrelated: true als het account al bestond, false als het is aangemaakt
    4. AccountReference → ID van het account
        1. (verkregen uit doelsysteem)

dryRun - Testen zonder echte acties

Wat is $actionContext.dryRun?

- Variabele die bepaalt of provisioningacties echt worden uitgevoerd of alleen gesimuleerd

Standaardgedrag:

- Tijdens een Preview staat $actionContext.dryRun = $true
- Scripts draaien wel, maar voeren geen wijzigingen uit.
- Je script moet hier zelf rekening mee houden
- Waarom belangrijk?
- Beschermt tegen onbedoelde wijzigingen tijdens testen
- Maakt veilig testen mogelijk zonder impact

Interfacen met HelloID (Technisch overzicht)

De belangrijkste input en output variabelen:

$actionContext
- Configuration.*
- CorrelationConfiguration
- Data.*
- DryRun (true/false)

$outputContext
- AccountCorrelated
- AccountReference
- AuditLogs
- Data
- PreviousData
- Success

Audit logs

Audit logging houdt bij welke acties worden uitgevoerd, zodat je kunt zien welke acties door welke doelsysteemkoppeling in HelloID zijn uitgevoerd en welke velden zijn veranderd.

Opslag:

- Alle auditmeldingen worden 1 jaar bewaard in Elastic en is zichtbaar via de Elastic rapportages
- Zichtbaar in het Audit log tabblad gedurende 3 maanden

Acties:

- Vaste lijst van acties beschikbaar (Zie de link rechtsonder op deze slide)
- Actieveld mag leeg blijven; in dat geval wordt de actie van het script gebruikt

```
$outputContext.AuditLogs.Add([PSCustomObject]@{
        Action  = "UpdateAccount"
        Message = "Successfully updated UserId for AFAS user [$($currentAccount.Gebruiker)]"
        IsError = $false
    })
```

```
$outputContext.AuditLogs.Add([PSCustomObject]@{
        Action  = "UpdateAccount"
        Message = "Error updating AFAS user [$($currentAccount.Gebruiker)]. Error Message: $($errorMessage.AuditErrorMessage)"
        IsError = $true
    })
```

Configuration

Gebruik van Input Forms in PowerShell Scripts

- **Doel:** Voer en sla inputparameters op voor PowerShell scripts (API-sleutels, wachtwoorden, URL's, etc.)
- **Voordeel:** Veiliger en handiger dan het hardcoderen van waarden in scripts

Werking:

- Maak een aangepast formulier om parameters in te voeren en op te slaan
- Verkrijg de waarden binnen **$actionContext.Configuration**

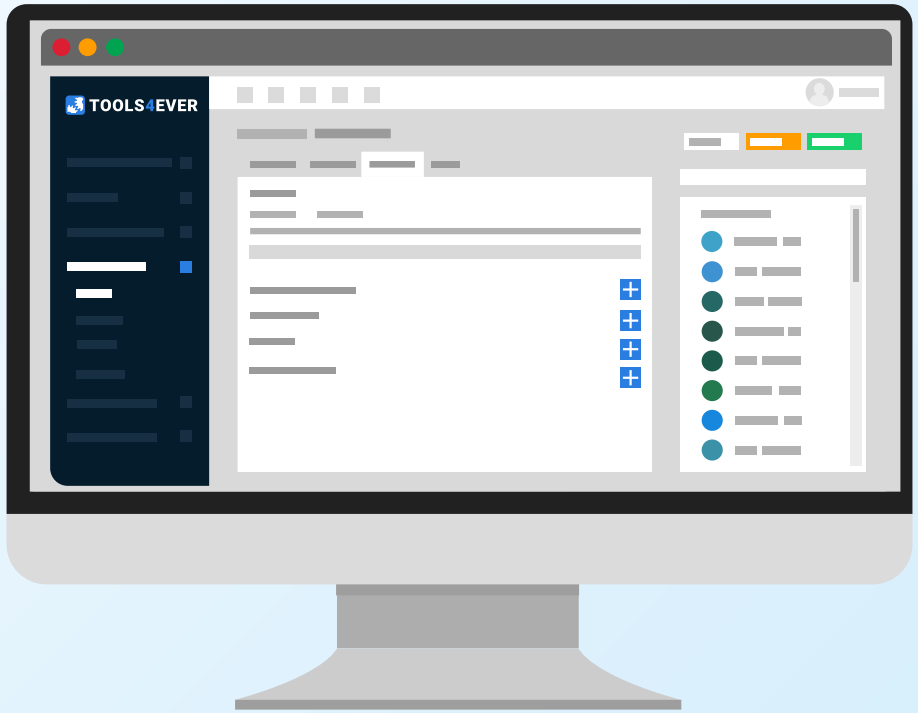
Sjabloon in JSON Editor:

- Ondersteunde formulierelementen: tekst, multi-line tekst, wachtwoord, e-mail, toggle, radioknop, dropdownmenu
- Pas aan naar de behoeften van je PowerShell-scripts

Lab 6 – Configuratieformulier

- Wat ga je doen in dit lab?
- - Je maakt een configuratieformulier aan
- - Je haalt inputparameters op in je script via $configuration

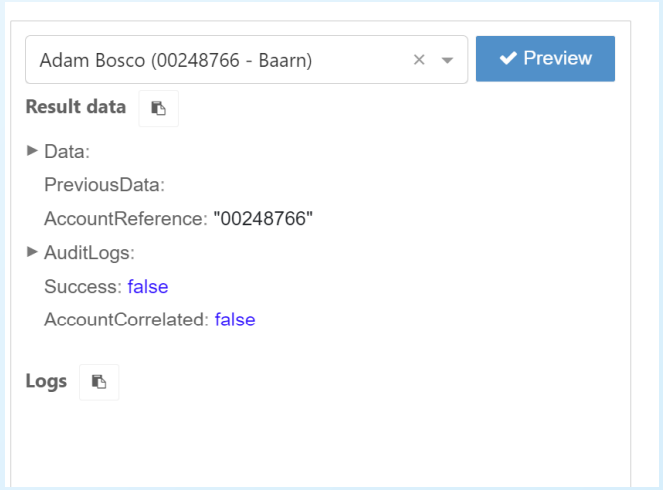
- Duur van dit lab: X minuten



Script preview

```
if (-Not($actionContext.DryRun -eq $true)) {
    #write your create logic here
}
```

- Test een script zonder het uit te voeren via een entitlement in een Business Rule.
- Preview-optie is beschikbaar in elk scriptbewerkingsdialoog
- DryRun-vlag voorkomt schrijfacties
- Zorg ervoor dat alle schrijfacties zijn ingesloten in een controle op $actionContext.DryRun:
- Uitvoerdata in $outputContext wordt rechts weergegeven:



Please check out the references on Customize an account script (PowerShell target systems)

Input- en outputvariabelen zichtbaar maken

- Handige stap tijdens het ontwerpen van een koppeling
- Laat zien wat HelloID aanlevert en wat je terugstuurt

Werkwijze

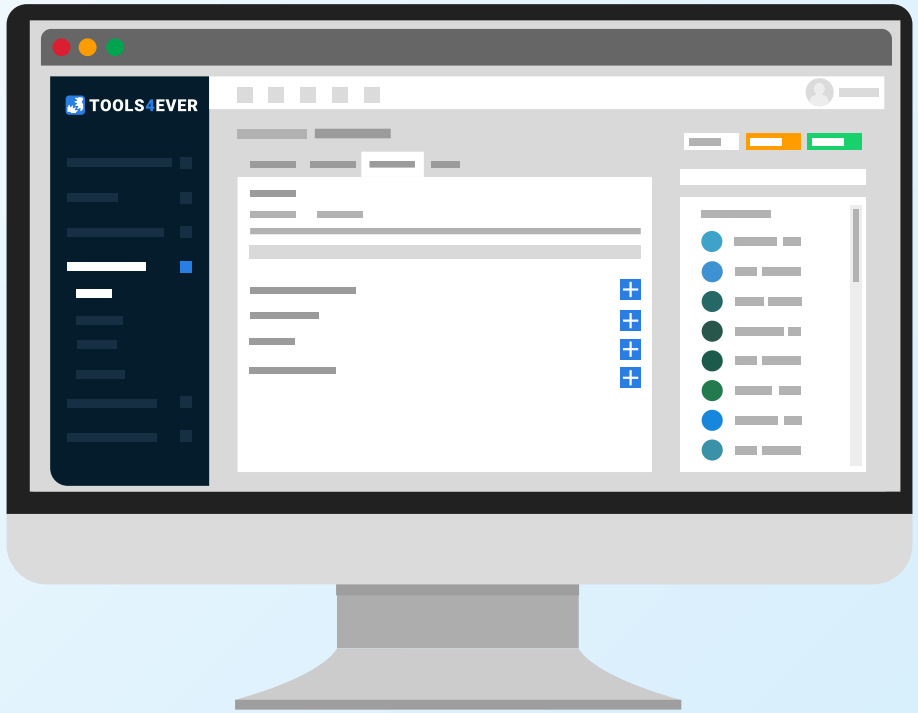
- Converteer de variabelen naar **JSON**
- Toon ze via **preview**
- Stop de scriptuitvoer met een **Throw**

- Throw ($actionContext | ConvertTo-Json)
- Throw ($outputContext | ConvertTo-Json)

Lab 7 – Create script

- Wat ga je doen in dit lab?
- - Je configureert het Create-script met correlatie en aanmaaklogica
- - Je test de preview en voert een create-actie uit

- Duur van dit lab: X minuten



Interfacen met HelloID - Update

- Update vinkje in field mapping
- Je roept nu het account aan met de account reference
- De belangrijkste input en output variabelen:

$actionContext
- References
- Configuration.*
- ~~CorrelationConfiguration~~
- Data.*
- DryRun (true/false)
- AccountCorrelated

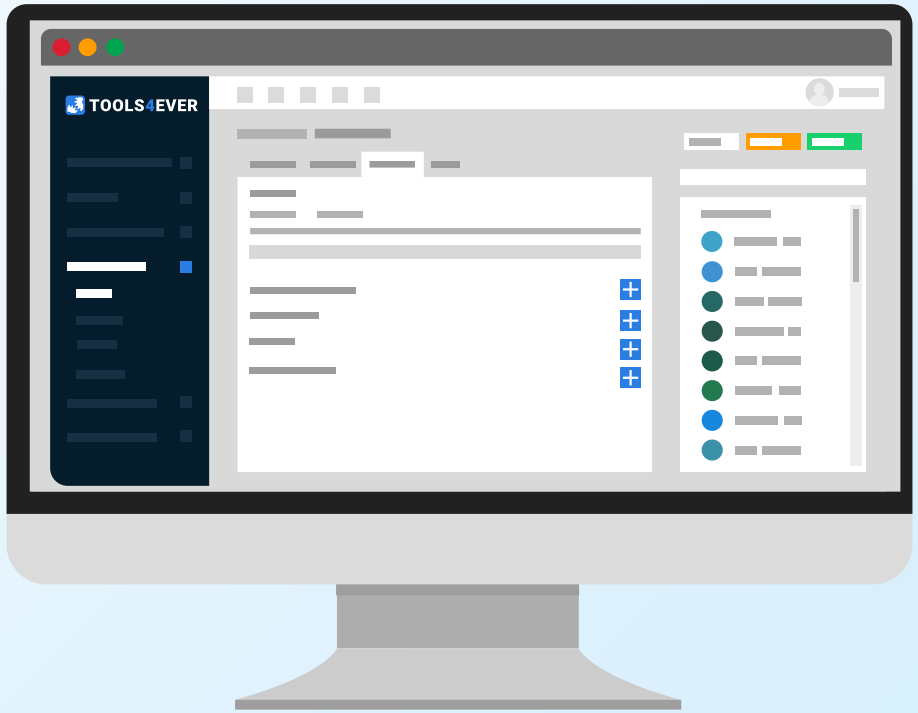
$outputContext
- ~~AccountCorrelated~~
- ~~AccountReference~~
- AuditLogs
- Data
- PreviousData
- Success

- (PowerShell target system variable reference)

Lab 8 – Update script

- Wat ga je doen in dit lab?
- - Je configureert het Update-script voor bestaande accounts
- - Je test de update zowel handmatig als via correlatie

- Duur van dit lab: X minuten



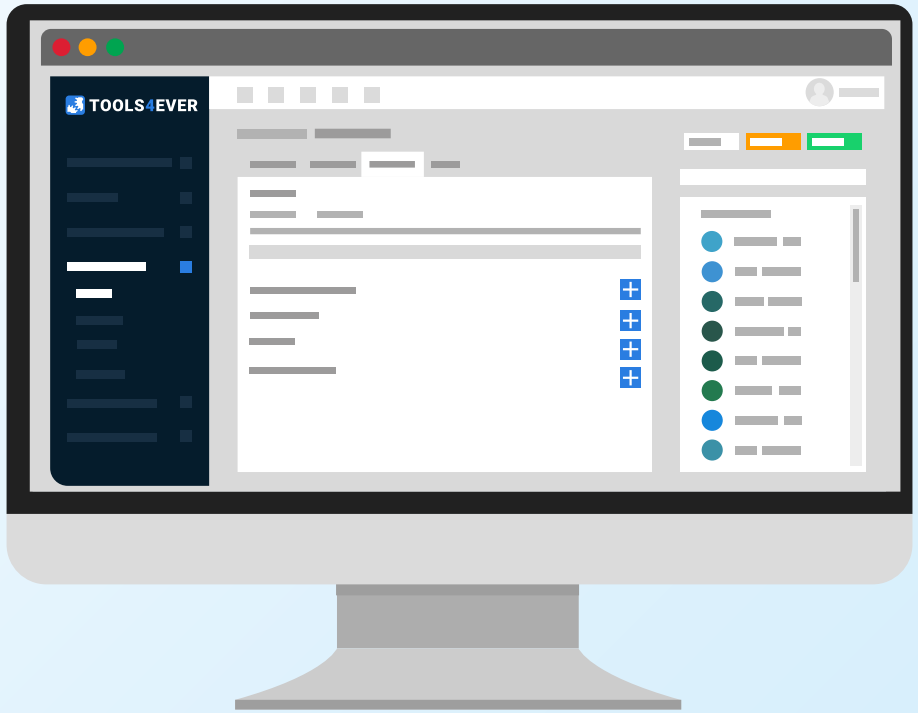
Interfacen met HelloID - Delete

- Delete vinkje in field mapping?
- De belangrijkste input en output variabelen:
- $actionContext
- Configuration.*
- ~~CorrelationConfiguration~~
- Data.*
- DryRun (true/false)
- References
- $outputContext
- ~~AccountCorrelated~~
- ~~AccountReference~~
- AuditLogs
- Data.*
- ~~PreviousData~~
- Success

Lab 9 – Delete script

- Wat ga je doen in dit lab?
- - Je configureert het Delete-script voor het verwijderen van accounts
- - Je test of accounts correct worden verwijderd uit het doelsysteem

- Duur van dit lab: X minuten



Quick reference guide

https://docs.helloid.com/

- Manuals
- Changelog
- API docs

https://feedback.helloid.com/

- Feature request

https://roadmap.helloid.com/

- Roadmap overview

https://helloid.statuspage.io/

- Status page

https://github.com/Tools4everBV

- Connector / Forms repositories

https://github.com/Tools4everBV/HelloID-Lib-ProvHelperFunctions

- Helper functions

https://github.com/Tools4everBV/HelloID-Conn-Prov-Target-V2-Template

- Target connector template

https://forum.helloid.com

- Community forum

Badges

Tools4ever gebruikt badges om certificeringen bij te houden

- Badges worden uitgegeven via het Acclaim-platform.
- Elke deelnemer ontvangt een e-mail van Acclaim om zijn badge te accepteren.
- Een Acclaim-account is vereist om de badges te accepteren.
- Badges kunnen worden toegevoegd aan sociale media (bijv. LinkedIn) zodat ontvangers hun certificering kunnen opslaan en delen.
- Volg de instructies van Acclaim om de badge te delen en selecteer de juiste uitgevende organisatie (Tools4ever B.V.).