HelloID Training

- Provisioning Connectoren

Doel van HelloID Provisioning training – Dag 2

- Deze tweede trainingsdag helpt IT-professionals om de HelloID Provisioning-module zelfstandig en volgens best practices in te richten, beheren en uitbreiden, met de nadruk op het bouwen van eigen provisioning-connectoren met PowerShell.

- Na afloop van de training kun je:
  - De volledige provisioningflow van bron tot doel configureren
  - Bronsystemen ontsluiten via PowerShell (zoals csv-bestanden of API's)
  - Doelsystemen aansturen met eigen PowerShell-scripts (Create, Update, Delete)
  - Werken met field mappings, inclusief complexe JavaScript-logica voor dataconversie en naamconventies
  - Correlatie instellen en gebruiken om bestaande accounts te herkennen
  - Slimme configuratieformulieren maken voor connectorparameters
  - Snapshots en raw data gebruiken om imports te analyseren
  - Problemen oplossen via preview en audit logging

De training is praktisch opgezet: leren door doen, met veel focus op zelf bouwen en testen.

Voor wie is deze training bedoeld?

- Deze training is voor jou als je:
  - HelloID beheert of gaat beheren binnen je organisatie
  - Provisioningkoppelingen wilt leren bouwen of beheren
  - Technisch verantwoordelijk bent voor automatisering van accountbeheer
  - Consultant of implementatiepartner bent voor HelloID

- Je beschikt over:
  - Ervaring met HelloID (zoals opgedaan in Dag 1 van de training)
  - Basiskennis van PowerShell (vereist voor Dag 2)
  - Enige ervaring met JavaScript (pre, voor complexe field mappings)

Kortom: je wilt zelf provisioningconnectoren kunnen ontwerpen, testen en beheren binnen HelloID.

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 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

Inleiding

Bronsystemen en PowerShell-connectoren in HelloID

- Bronsystemen leveren informatie over medewerkers en contracten voor het aanmaken en beheren van accounts, toegang en rechten
- Niet alle personeelsgegevens komen uit commerciële HR-systemen; sommige zijn intern of gebruiken platte bestanden, SQL-databases of API-webservices
- We behandelen de configuratie van een PowerShell-connector voor HelloID. Deze connector maakt gebruik van een eenvoudige csv-dataset die gebruikers en afdelingen bevat

Source connector - Toevoegen

- Het toevoegen van een PowerShell-source kan via de **Source Template** connector uit de connectorencatalogus
- De catalogus wordt gevoed vanuit onze GitHub-repositories
- Na het toevoegen kun je een **eigen naam en beschrijving** instellen
- Een **lokale agent** is vereist wanneer je niet-standaard PowerShell-modules gebruikt (*Import-Module*) of toegang nodig hebt tot interne netwerkresources, zoals een lokaal SSL-certificaat voor je API-verbinding
- Een **cloud agent** gebruik je in alle andere situaties

Bronconnectoren - on-premise en cloud

- **On-premise PowerShell agent**
  - Zet de “Execute on-premises” switch aan om scripts lokaal uit te voeren via de HelloID agent
  - Vereist een lokaal draaiende agent
  - Afhankelijk van de geïnstalleerde PowerShell versie (minimaal 5.1)
- **Cloud PowerShell agent**
  - Zet de “Execute on-premises” switch uit om scripts via de HelloID cloud uit te voeren
  - Geen lokale agent vereist
  - Beperkt tot PowerShell Core 7 en 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 –** De opgehaalde gegevens worden weergegeven op de *Raw Data*-tab.
- **Mapping –** De ruwe data wordt gemapt op de juiste persoons- en contractvelden in HelloID
- **Snapshot –** Een momentopname wordt gemaakt van de persoonsgegevens en de bronconfiguratie, inclusief 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

- Persons-script levert persoonsgegevens aan HelloID
- Script retourneert:
  - een hashtable waarin elke sleutel een persoon is,
  - met bijbehorende contracten
- Structuur:

```
$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}
)
```

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

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)

Persons import script

Person import script - vereiste velden (Contract)

Contract:

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

Optioneel:

- Afdelingscode
- Functiecode / omschrijving
- FTE

Departments import script

- Haalt afdelingsinformatie op (geen mapping van toepassing)
- Vult de afdelingsselectie in de conditie van de business rules
- **Specifiek formaat vereist:** Hashtable met afdelingens-objecten
- **Vereiste informatie voor een afdeling:**
  - **ExternalId:** Unieke sleutelwaarde (afdelingscode)
  - **DisplayName:** Weergavenaam van de afdeling
- **Optionele velden voor een afdeling:**
  - **ManagerExternalId:** ExternalId van de manager van de afdeling
  - **ParentExternalId:** Bovenliggende afdeling
  - Deze velden worden gebruikt wanneer “Primary Manager” op “From department of primary contract” staat ingesteld
  - Er kan maar één manager per afdeling worden doorgegeven

Department import script

- Het departments-script levert een lijst van afdelingen aan HelloID.
- Deze afdelingen 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 worden alleen gebruikt als “Primary Manager” op “From department of primary contract” staat ingesteld.
- Er kan maar één manager per afdeling worden doorgegeven.

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

Gebruik van een configuratieformulier in PowerShell Scripts

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

Werking:

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

Raw data

- De tab **Raw data** toont de gegevens die HelloID importeert uit het bronsysteem voor elke persoon, zonder enige modificaties, filters of mappings
- Deze weergave is handig voor troubleshooting, waar je de data in onbewerkte staat kunt bekijken
- Door een persoon uit de weergegeven lijst te selecteren, kun je de **Raw data** van die persoon in detail 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

- HelloID bepaalt het exacte tijdstip binnen het geselecteerde 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

Handmatige import

- Alleen de raw data wordt gemapt en geïmporteerd en verwerkt in een nieuw snapshot, zonder verwerking of toepassing van business rules in de 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

- Het threshold-systeem voorkomt ongewenste wijzigingen door imports automatisch te blokkeren wanneer bepaalde limieten worden overschreden.

Hoe werkt het?

- HelloID vergelijkt nieuwe gegevens met de vorige import van hetzelfde bronsysteem.
- Controle op:
  - Aantal toegevoegde/verwijderde personen
  - Lege verplichte velden (‘required this field’ staat aan)

Source system thresholds

Thresholds en blocked persons

Blokkering bij overschrijding:

- Bij overschrijding van de threshold wordt de import geblokkeerd tot handmatige goedkeuring
- Als de threshold niet wordt overschreden, worden de geblokkeerde personen overgeslagen en worden alle andere personen gewoon geimporteerd

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

- Om wijzigingen toe te passen en zichtbaar te maken, moet je de vault-snapshot vernieuwen
- 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: Als de gegevens van het bronsysteem zijn veranderd, of de PowerShell scripts zijn veranderd, is een nieuwe volledige import vereist
- Nuttig als je de configuratie wilt aanpassen zonder nieuwe gegevens van het bronsysteem te importeren
  - **Tip**: Gebruik daarna de snapshot history om het resultaat te valideren

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

- Via het menu-item **Target -> Systems** in het navigatievenster
- Pas de configuratie aan om te bepalen hoe gegevens van HelloID naar het doelsysteem worden geschreven
- Bij het toevoegen van een nieuw doelsysteem:
  - "**Disable target system**" wordt standaard ingeschakeld, waardoor alle acties uitgesloten worden van de enforcement van de business rules
  - Alle **Thresholds** worden standaard op 1 ingesteld

PowerShell target systems

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:

- In het primaire doelsysteem zet je bij de gewenste mapping aan:
  - → “Store in account data”
- In het secundaire doelsysteem stel je in:
  - → “Use account data from systems”
- 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

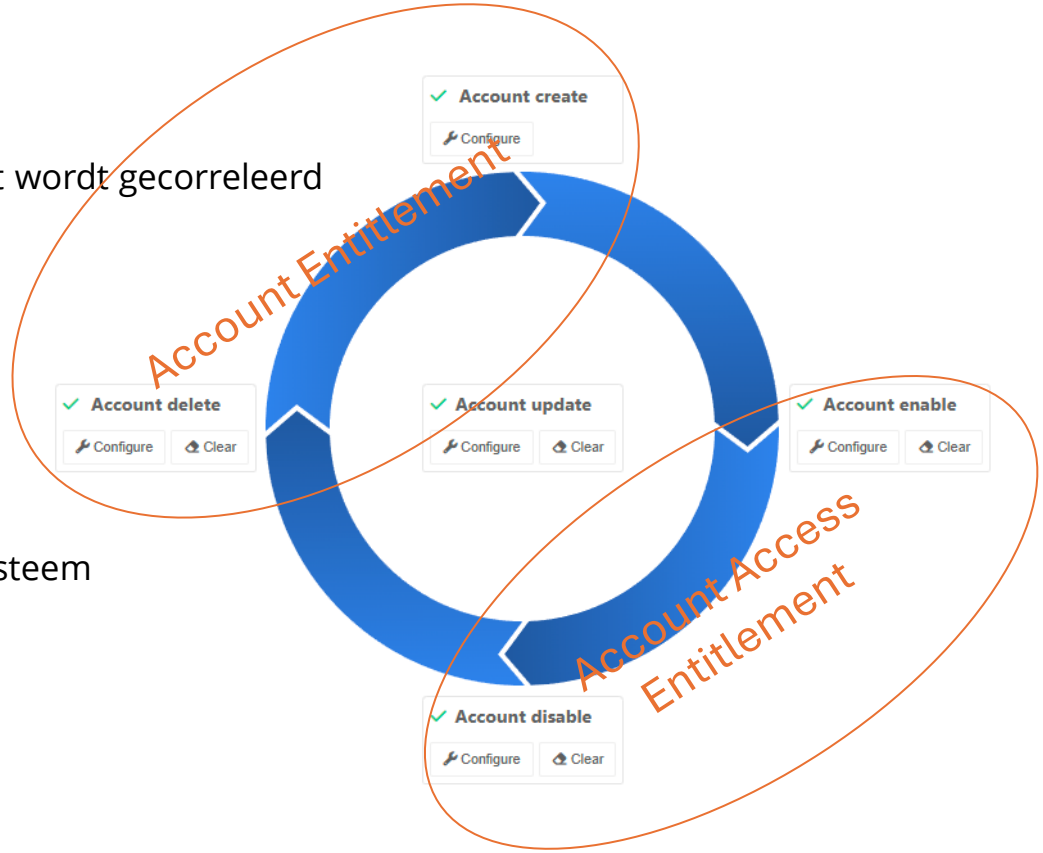
‘Account Access’ entitlement

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

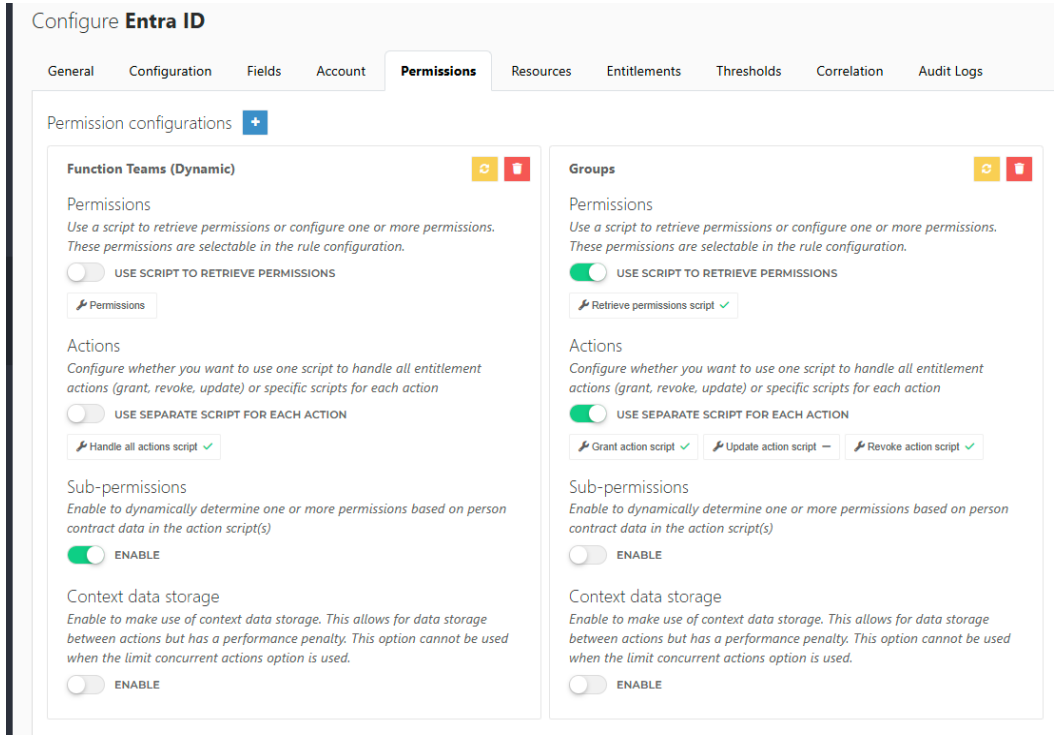
Account Update

- 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

Tip: Heb je een actie niet nodig: wis het script



Permissies



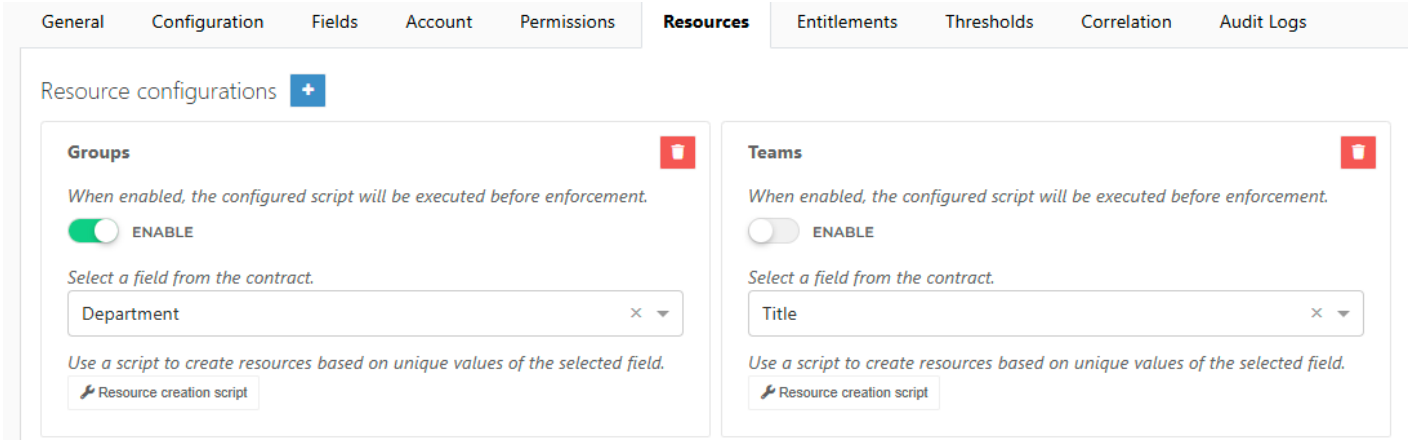
Normale Permissie-entitlements

- 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 te maken van sub-permissions kun je het 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

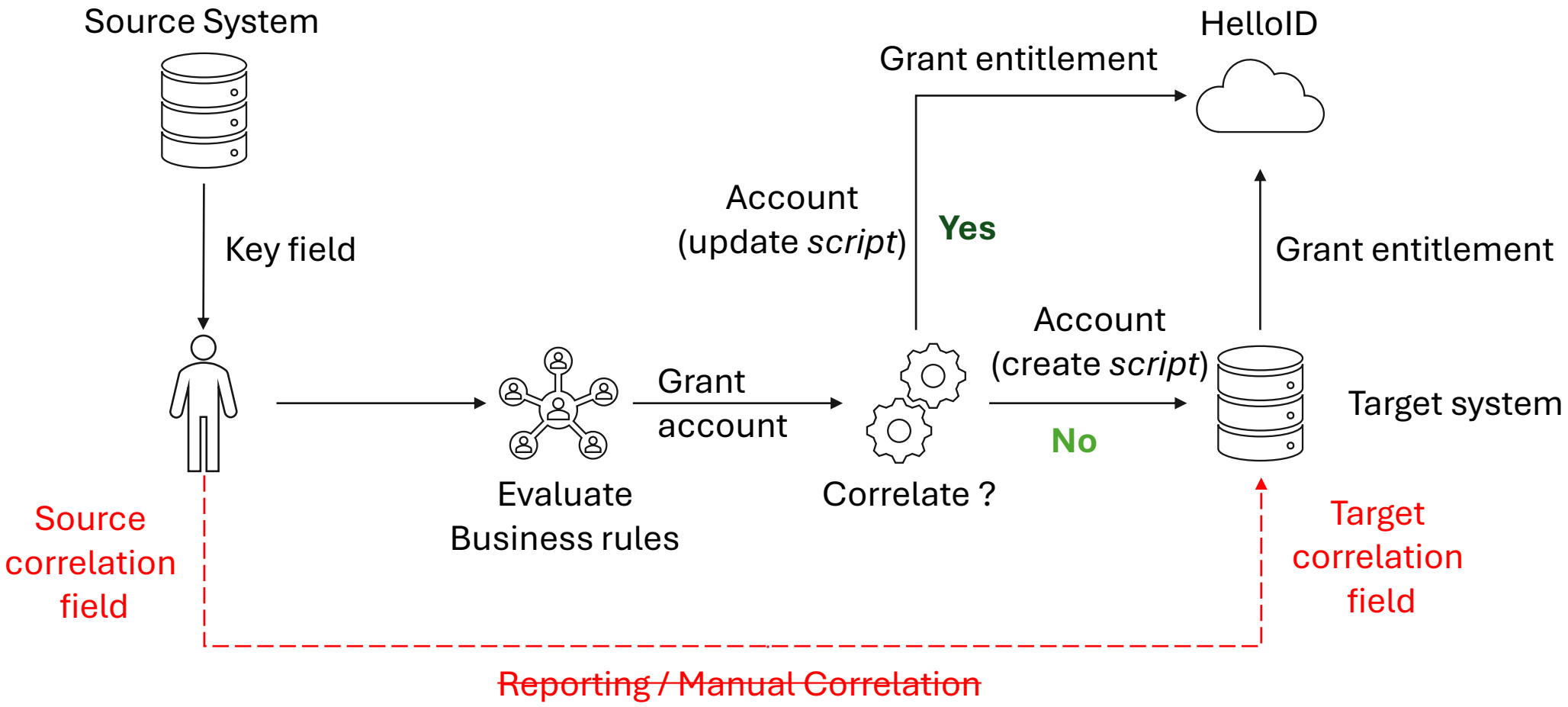
Wat is correlatie?

- 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.
- De resultaten worden vergeleken met de verwachte entitlements in HelloID.

Belangrijk om te weten:

- Alleen beschikbaar voor on-premises Active Directory en 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?

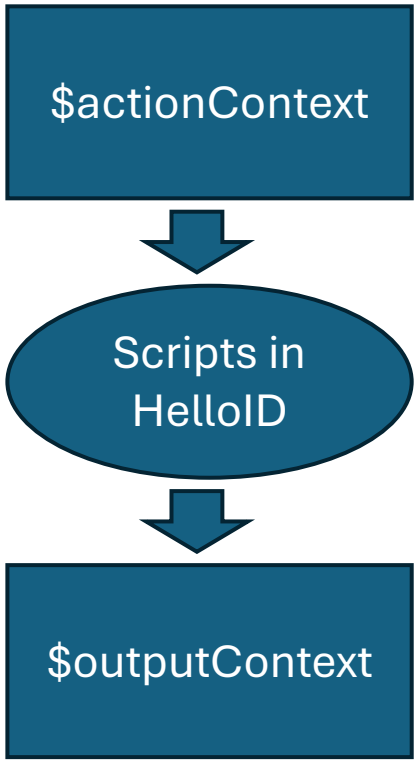
- Variabele die bepaalt of provisioningacties echt worden uitgevoerd of alleen gesimuleerd

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



PowerShell target system variable reference

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
    })
```

Audit logs (target systems)

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

Configure an input form (PowerShell target systems)

Lab 6 – Configuratieformulier

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

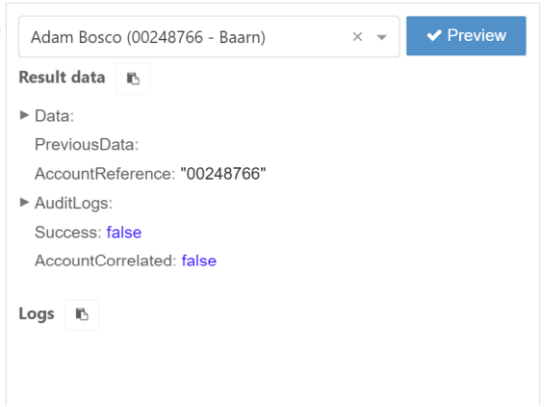
- Duur van dit lab: X minuten

Script preview

- 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:

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

- Uitvoerdata in $outputContext wordt rechts weergegeven:



Customize an account script (PowerShell target systems)

Input- en outputvariabelen zichtbaar maken

- Nadat de field mapping is opgezet en je hebt een ontwerp van de koppeling gemaakt, is het in het begin best handig om zichtbaar te maken hoe de input- en output variabelen van HelloID eruitzien.

- Je kan dit doen door de variabelen te converteren naar JSON en dan via preview te tonen en daarna meteen een error te generen, zodat de scriptuitvoer wordt gestopt via Throw:
  - Throw ($actionContext | ConvertTo-Json)
  - Throw ($outputContext | ConvertTo-Json)

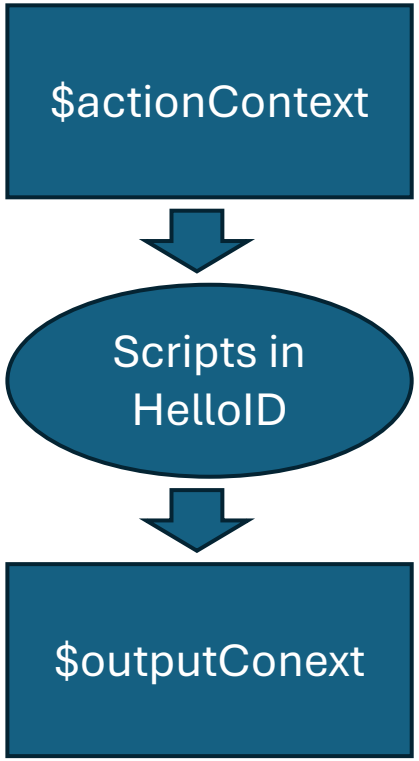
Lab 7 – Create script

- 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)



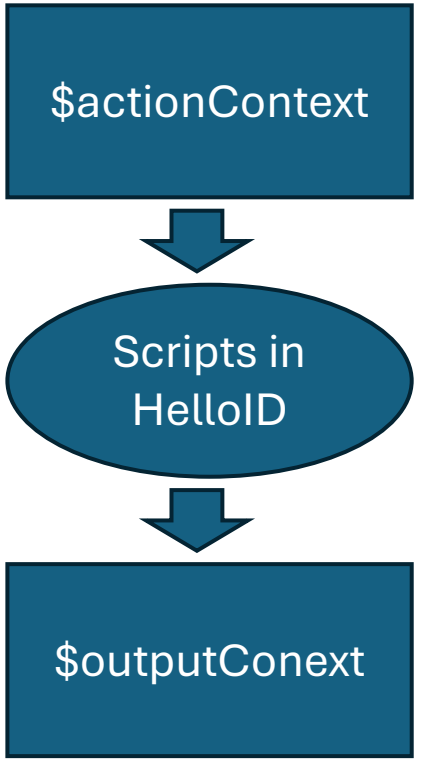
Lab 8 – Update script

- 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



- (PowerShell target system variable reference)

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-Prov-HelperFunctions
  - 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.).