TOOLS4EVER

HelloID Training

Provisioning PowerShell Connectors

V1.4 / 07-03-2023

Agenda

- PowerShell Source connectors
- PowerShell Target connectors

PowerShell Connectors

Training Prerequisites

TOOLS4EVER

Before we start… make sure your Active Directory contains the following

OU structure

- HelloID Training
    - Users
    - Disabled users
    - Groups

AD Groups

- 4 random AD groups for demo purpose

Preparation

Installing and configuring the HelloID Agent

TOOLS4EVER

Preperation

Installing and configuring the HelloID Agent

30 minutes

- Please use the reference article on page 3 of the Labs document.

Please check out the references on https://docs.helloid.com
Manuals for Administrators → Agents → Install or manage an Agent

PowerShell Source Systems

TOOLS4EVER

PowerShell Source Systems

- Introduction
- Adding a PowerShell source system
- Configuration
- RAW data
- Mapping
- Imports
- Thresholds
- Snapshots
- Blocked persons

Introduction

Not every source of personnel data is a commercially available HRMS program. Some sources are built in-house and are completely unique to the organization. Other sources are just based on flat files, SQL databases, API webservices and more.

Source systems are the starting point of every provisioning process. They are where the process retrieves information about the people within your organization. The information retrieved from the source system is used to create and manage accounts, account access, and permissions.

Adding a custom PowerShell source system

In this section, we will go over the very basics of configuring the custom PowerShell source connector. We will define a static data set with only a couple of users and departments so that you can get a feeling about how system works and interacts with HelloID.

Adding a custom PowerShell source system can easily be accomplished by using the Source Template from the template list. By adding a custom connector there is also an option to define a custom name and description for the system.

A local agent is required when you need to use non-default PowerShell modules (by using Import-Module) or when you need to have a local SSL certificate on you API connection.

Adding a custom PowerShell source system

When you add a new PowerShell source system, you will see a complete list of all available PowerShell source connectors on our GitHub page. This integration is currently only available for source systems.

A local agent is required when you need to use non-default PowerShell modules (by using Import-Module) or when you need to have a local SSL certificate on you API connection.

Configuration (General)

After the new PowerShell source system is added, you will need to make some configuration changes. These changes affect how data from the source system flows into HelloID.

From this point we will cover the configuration and management options that are common to all source systems. It will lead you through how to make those changes, run a manual import, remove a system, make global configuration changes, and create a new vault snapshot.

Configuration (System)

Persons PowerShell Script

This area is where you define the PowerShell script that retrieves both **person** and **contract** information.

Each person object is represented as a hash table object within PowerShell. That is, a collection of key/value pairs. You may import as many key/value pairs as you wish, but you must ensure that each person record is unique and has at least one field that uniquely identifies them, such as an employeeID or a GUID. This is often referred to as the "External ID".

EmployeeID is preferred value to identify persons as unique. Other attributes are possible but could cause some issues depending on the target system identifier or primary contract calculation.

Configuration (System)

Departments PowerShell Script

This area is where you define the PowerShell script that retrieves departmental information for your organization.

Departments can be referred to by defining a business rule filter based on the department of a person's contract.

Just like persons and contracts, departments are defined as hash table objects within PowerShell.

Configuration (System)

Person

When returning a person object back to HelloID the following object properties are mandatory.

- ExternalID.
- DisplayName.

Department

When returning a department object back to HelloID the following object properties are mandatory.

- ExternalID.
- DisplayName.

Optionally you also have the option to return (Name, ManagerExternalId, ParentExternalId).

The department script configuration is also mandatory when using manager determination based on the department of the employees' primary contract.

Configuration (System)

Performance

Each object should be sent individually using the PowerShell Write-Output command.

For example: if you have 1,000 person objects to send, the Write-Output command need to be called 1,000 times—once for each object—instead of once for the entire batch.

The result should be returned to HelloID as JSON output model.

Max response size is limited to 1 Mb, HelloID automatically takes care when this limit is exceeded by applying padding to the incoming data.

Configuration (System)

On-premise PowerShell

By turning on the "Execute on-premises" switch you can execute the Persons and Departments PowerShell scripts you've created directly on your local HelloID provisioning agent service.

A local running agent is required to use this option.

Be aware that functionality is limited to the PowerShell version installed on the local server running the HelloID provisioning agent.

PowerShell 5.0 or higher is needed for running local PowerShell scripts.

Configuration (System)

Cloud PowerShell

By turning off the "Execute on-premises" switch you can execute the Persons and Departments PowerShell scripts you've created directly on your HelloID instance's server.

A locally running agent is not required to use this option.

Be aware that functionality is currently limited to PowerShell Core commands and importing extra modules is not supported.

RAW data

The **Raw data** tab lets you see the data that HelloID is importing from the source system for every person, without any modifications, filters, or mappings being applied to it.

Such a view is useful during troubleshooting where you can view the data in an untouched state. By selecting a person from the displayed person list makes It possible to zoom in on a specific person’s RAW data showing a detailed overview.

RAW data is available after the first import is executed successfully.

Lab 1

Adding and configuring a PowerShell Source System

TOOLS4EVER

Lab 1

Create a custom PowerShell source system

60 minutes

- Please use the reference article on page 4 of the Labs document.

Please check out the references on https://docs.helloid.com
Manuals for Administrators → Provisioning → Configure a Custom PowerShell source system

Mapping

Once a source system is added, you will likely need to make some configuration changes. These configuration affects how data from the source system flows into HelloID.

The configuration of a field mapping is required to translate the RAW data towards the internal HelloID vault database model. Make sure the mapping is in place before executing the full import.

Mapping

Person

The **Person** tab lets you configure how data from the source system is mapped onto fields within HelloID Person records.

Each field can contain a **Fixed** (static) value, a **Field** value pulled from the source data, or a **Complex** value.

Complex values are calculated by JavaScript that is run during the import.

Additional fields can be added with the **Map additional field** tool. They are appended to the end of the field list.

Mapping

Contract

The **Contract** tab allows you to configure how employment and contract information from the source system is mapped onto fields in HelloID.

Each field can contain a **Fixed** (static) value, a **Field** value pulled from the source data, or a **Complex** value.

Complex values are calculated by JavaScript that is run during the import.

Additional fields can be added with the **Map additional field** tool. They are appended to the end of the field list.

Imports

The **Import schedule** tab allows you to configure the system for automatic import according to a schedule. Here you may select an hour of the day within which HelloID will run the import. The specific time within the hour is decided by HelloID and not user-configurable. After the source system has been imported, its Person records will be run through evaluation and enforcement for all business rules for the configured target systems.

A manual import can also be performed so only raw data will flow into HelloID and will not initiate any processing or enforcement in target systems.

Thresholds

The **Thresholds** tab lets you automatically block imports when the amount of added or removed Persons exceeds a specified value. When an import occurs for a source system, the pending activity is compared to the last import for the same source system. If the amount of activity exceeds the threshold, the import is blocked until manually approved. In this way, thresholds are safety nets. They reduce the likelihood of a major mistake in the provisioning process if a source system contains bad data or no data.

Since the HelloID release (2021.01) we enable the removal of persons threshold by default when a new source system is added.

Snapshots

After you changed source system configurations, changes are not immediately applied to your current vault snapshot, and are not immediately visible on the provisioning dashboard.

For example, if you alter display name formatting, change primary contract or manager determinants, or alter complex field mappings, you will not immediately see these changes reflected in the Persons list.

To apply and see your changes, you must refresh the vault snapshot. The new Snapshot button refreshes the current vault snapshot and applies all pending changes, without performing a new import or otherwise re-fetching any data from source systems.

Snapshots

The new snapshot functionality isn’t working if the data of the source system has been changed, therefor a new full import is required.

The new Snapshot button offers a faster way to experiment with configurations and formatting. Additionally, it is useful if you wish to alter these settings but specifically avoid importing any new data from your source system(s).

Blocked persons

Each mapped field on the **Person** and **Contract** tabs has a **Require this field** toggle. When this toggle is turned on, any Persons for whom the field's mappings produce an empty value will not be imported. Persons blocked in this way are listed on the **Blocked persons** tab.

When the blocked persons threshold is configured, and the current import exceeds the limit the entire import will get blocked. Otherwise the blocked persons are skipped without affecting the rest of the import.

Lab 2

Data validation and thresholds for source systems

TOOLS4EVER

Lab 2

30 minutes

Data validation and thresholds for source systems

- Please use the reference article on page 8 of the Labs document.

Please check out the references on https://docs.helloid.com
Manuals for Administrators → Provisioning → How to configure and manage source systems

PowerShell Target Systems

TOOLS4EVER

PowerShell Target Systems

- Introduction
- Adding a custom PowerShell target system
- Account (lifecycle)
- Permissions (list / grant / revoke)
- Scripting
- Uniqueness Validation
- Correlation
- Audit Logs

Introduction

Target systems are where HelloID's provisioning processes do their work, they are responsible for granting and revoking accounts, account access, and permissions.

This section will cover the common configuration and management options that are found in multiple types of target systems.

Adding a custom PowerShell target system

Adding a custom PowerShell target system can easily be accomplished by using the target systems menu item from the general navigation pane on the left-hand side of the screen.

After you've added a target system, you will need to make some configuration changes. These changes affect how data from HelloID is written to the target system.

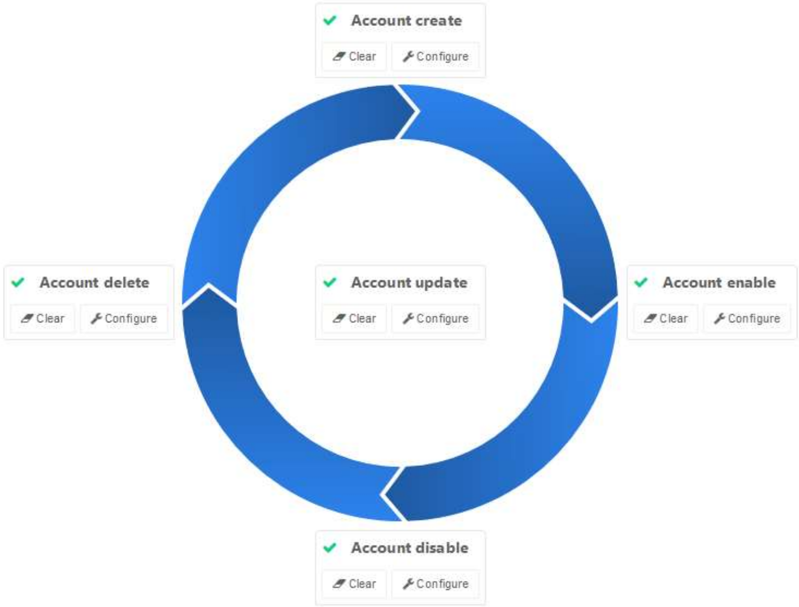
When you enable the option “Disable system”, all the actions of the target system will be excluded from the enforcement of entitlements.

Account (lifecycle)

Before we start working with a custom target system connector, it's important to have a solid understanding of the lifecycle stages of a user account.

User account

- Create
- Enable
- Update
- Disable
- Delete



Permissions

The entitlement scripting is responsible for retrieving a list of permissions, these permissions can be granted by defining these as entitlements inside a business rule. When the unique identifier for the entitlement is changed, all the granted entitlements of this type get revoked and will be re-granted afterwards using the new identifier as key.

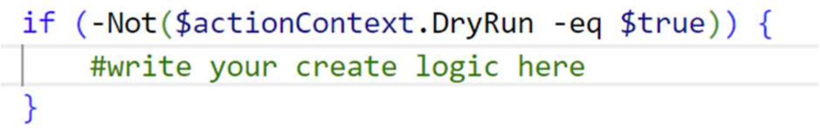
The grant / revoke scripting will be used to grant new permissions which haven’t assigned before or revoking the permissions which have been granted by HelloID in the past.

Mapping

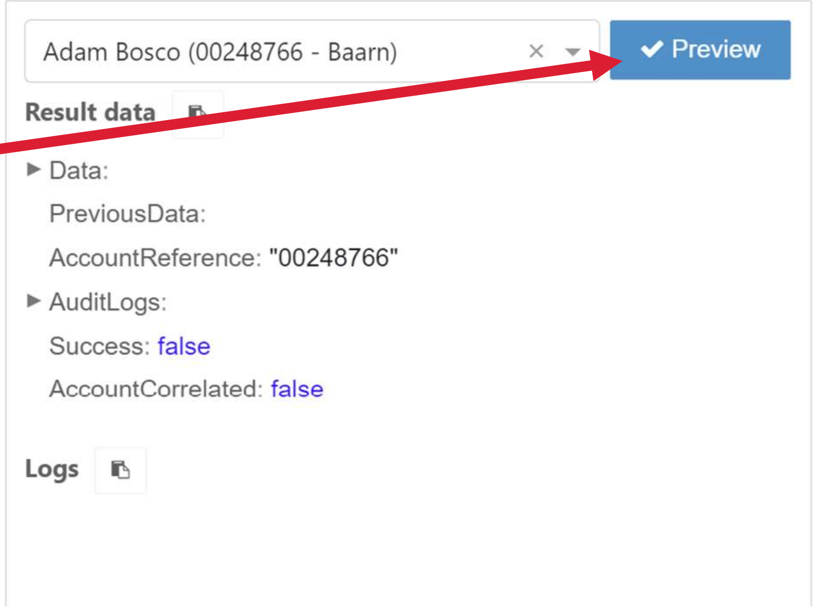
Account lifecycle events.

- Full Person model available on the fields mapping tab.

- Dry run condition



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



Adam Bosco (00248766 - Baarn)

Result data

▶ Data:
PreviousData:
AccountReference: "00248766"
▶ AuditLogs:
Success: false
AccountCorrelated: false

Logs

TOOLS4EVER

Mapping

Mapping

- Mapping can be done at the “Fields” tab within the configuration of the target system.
- Mapping can be made for specific actions (e.g. create, update, enable)
- You can use Fixed, Field and Complex Mapping (Javascript) to map the data
- The full person model is available within the mapping of the fields
- Fields that are mapped can be resolved within the PowerShell script by using the object $actionContext.Data
- Store in account data switch will store te generated value for the use in other target systems

Scripting

Mapping

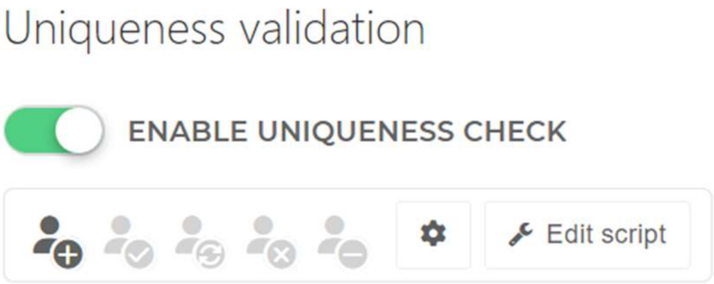
- When you want to use mapped fields in notifications, the toggle “use in notifications” must be enabled.
- It’s now possible to run a preview of the configured mapping including the configured complex mapping.

Uniqueness validation

Uniqueness validation is available to validate whether certain generated fields are unique within the target system. (e.g. username or email address)

The validation takes place on the basis of a PowerShell script to be created that carries out the check in the target system.

Once enabled, you need to enable the action



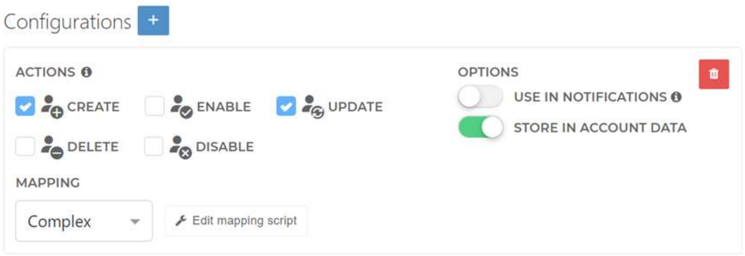
Uniqueness validation

ENABLE UNIQUENESS CHECK

Scripting

Store in account data

- Using the switch “store in account data” will store the generated field from te complex mapping into the person object.
- Use account data from systems gives the possibility to use previous stored data within the javascript of the complex mapping.



Configurations +

ACTIONS

☑ CREATE □ ENABLE ☑ UPDATE

□ DELETE □ DISABLE

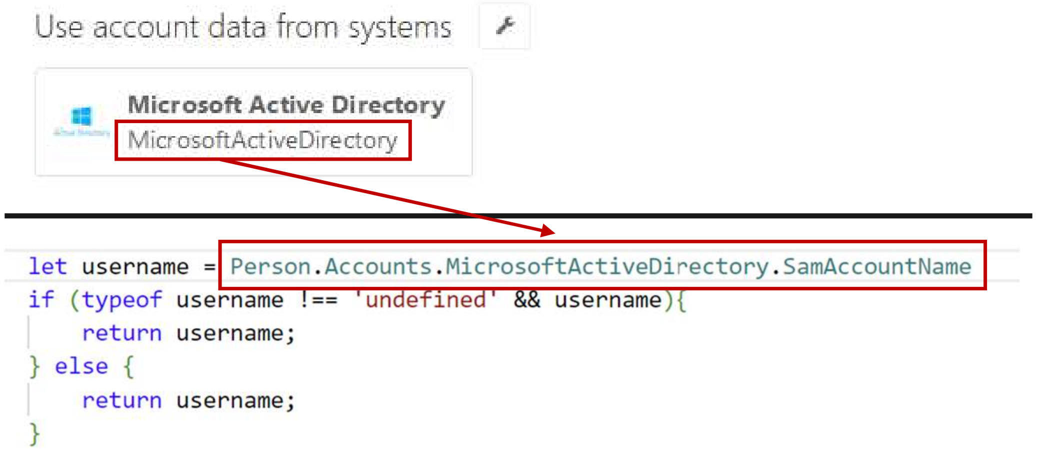
OPTIONS

○ USE IN NOTIFICATIONS

● STORE IN ACCOUNT DATA

MAPPING

Complex



Use account data from systems

Microsoft Active Directory

MicrosoftActiveDirectory

```
let username = Person.Accounts.MicrosoftActiveDirectory.SamAccountName
if (typeof username !== 'undefined' && username){
    return username;
} else {
    return username;
}
```

Scripting

Output

- The objects nested within the $outputContext.Data object could be used for notification purposes.
- Audit logging can be added to $outputContext.Auditlogs and have 3 sub objects. Within Auditlogs you can store Action, Message and IsError (True/False)
- There are 3 important objects to send within $outputContext:
  - $outputContext.Succes : Result of the script (True/False)
  - $outputContext.Data : Objects that could be used for notification
  - $outputContext.AccountReference : Reference of the created account

Lab 3

Create a PowerShell target system to export data

TOOLS4EVER

Lab 3

30 minutes

Create a PowerShell target system to export data

- Please use the reference article on page 10 of the Labs document.

Please check out the references on https://docs.helloid.com
Manuals for Administrators → Provisioning → How to configure and manage target systems

Lab 4

Add DisplayName generation on target system

Lab 4

30 minutes

Add DisplayName generation on target system

- Please use the reference article on page 12 of the Labs document.

Lab 5

Use data from another system

TOOLS4EVER

Lab 5

Use data from another system

20 minutes

- Please use the reference article on page 14 of the Labs document.

Please check out the references on https://docs.helloid.com
Manuals for Administrators → Provisioning→ How to use data from one target system in another target system

Correlation

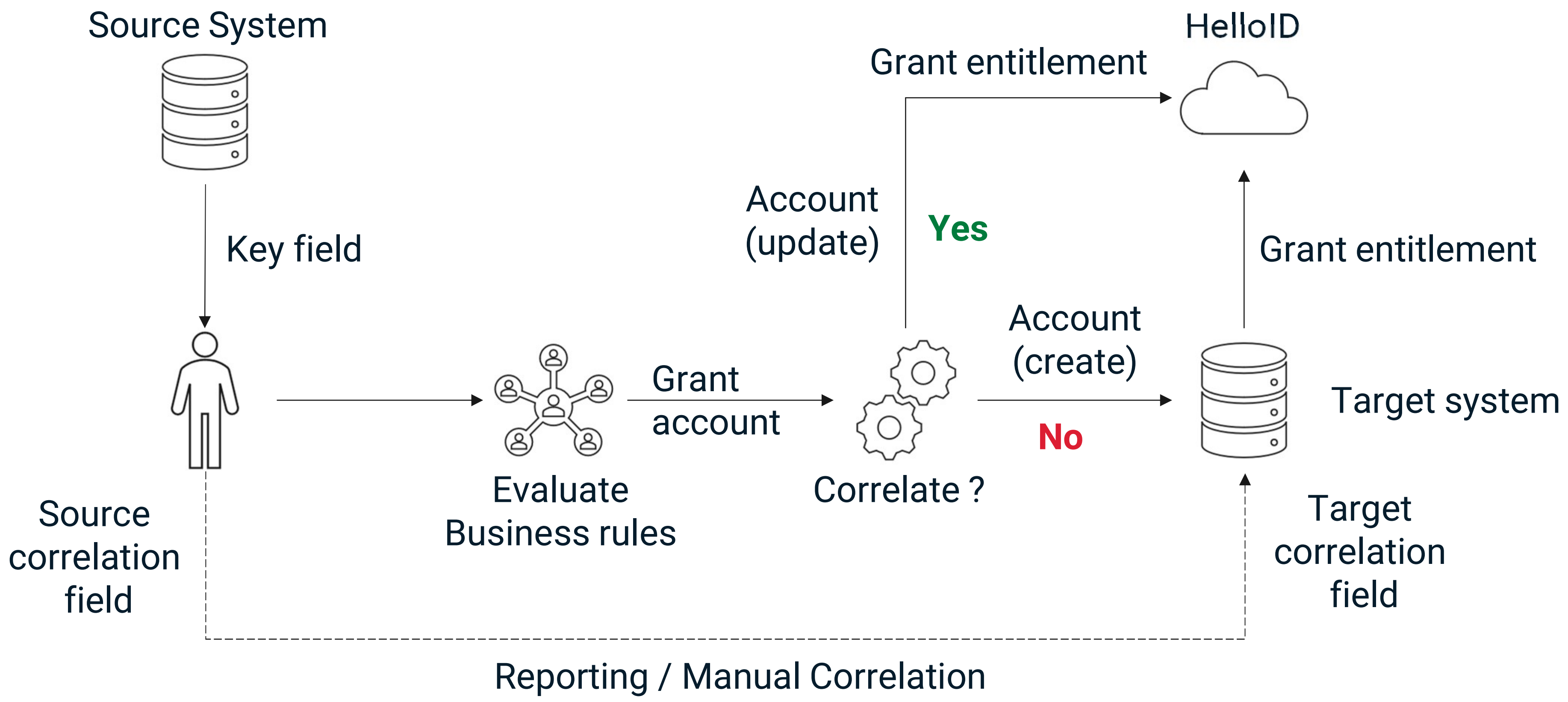
- Correlation prevents the creation of duplicate accounts.
- Correlation logic needs to be configured inside the account create event scripting.
- On the tab Correlation, we can enable correlation and set the correct person and account fields.
- Manual correlation is not possible.
- No UI reporting is available.

Correlation

- Correlation settings can be resolved within the PowerShell script from the object $actionContext.CorrelationConfiguration

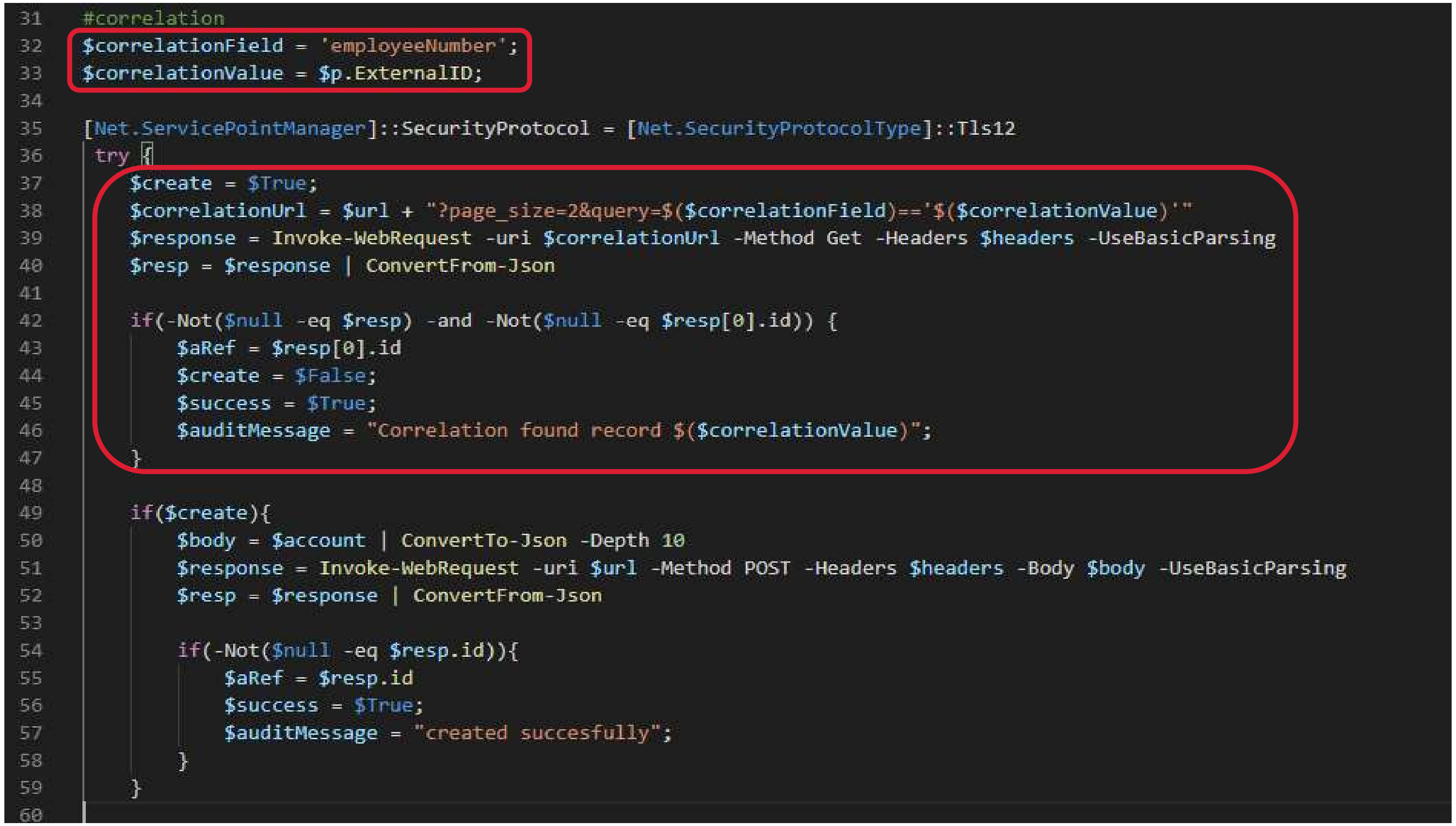
- Within this object you will have 5 values
  - $actionContext.CorrelationConfiguration.Enabled
  - $actionContext.CorrelationConfiguration.accountField
  - $actionContext.CorrelationConfiguration.accountFieldValue
  - $actionContext.CorrelationConfiguration.personField
  - $actionContext.CorrelationConfiguration.personFieldValue

Correlation (How does it work)



TOOLS4EVER

PowerShell Correlation (example)



```
#correlation
$correlationField = 'employeeNumber';
$correlationValue = $p.ExternalID;

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
 try {
    $create = $True;
    $correlationUrl = $url + "?page_size=2&query=$($correlationField)=='$($correlationValue)'"
    $response = Invoke-WebRequest -uri $correlationUrl -Method Get -Headers $headers -UseBasicParsing
    $resp = $response | ConvertFrom-Json

    if(-Not($null -eq $resp) -and -Not($null -eq $resp[0].id)) {
        $aRef = $resp[0].id
        $create = $False;
        $success = $True;
        $auditMessage = "Correlation found record $($correlationValue)";
    }

    if($create){
        $body = $account | ConvertTo-Json -Depth 10
        $response = Invoke-WebRequest -uri $url -Method POST -Headers $headers -Body $body -UseBasicParsing
        $resp = $response | ConvertFrom-Json

        if(-Not($null -eq $resp.id)){
            $aRef = $resp.id
            $success = $True;
            $auditMessage = "created succesfully";
        }
    }
```

Audit Logs

Audit logs vary between applications, devices, systems, and operating systems but are similar in that they capture events which can show “who” did “what” activity and “how” the system behaved.

A log file event will indicate what action was attempted and if it was successful or not. This is critical to check during routine activities like creating and updating accounts, and to determine when a system component is failing or incorrectly configured.

Lab 6

Correlation on the PowerShell target system

TOOLS4EVER

Lab 6

Correlation on PowerShell target system

20 minutes

- Please use the reference article on page 16 of the Labs document.

Custom events

Within HelloID it is possible to send notifications based on a custom event. By example to sent a notification when a certain value (e.g. displayName) has been changed.

At the section Business you can find the menu option “Custom events”.

The created event can be used as trigger within the notification configuration.

Lab 7

Create notification based on custom event

TOOLS4EVER

Lab 7

Create notification based on custom event

20 minutes

- Please use the reference article on page 17 of the Labs document.

Quick reference guide

- https://docs.helloid.com/
  - Manuals
  - Changelog
  - API docs
- https://feedback.helloid.com/
  - Feature request
- https://roadmap.helloid.com/
  - Roadmap overview
- https://github.com/Tools4everBV
  - Connector / Forms repositories
- https://forum.helloid.com
  - Community page
- https://helloid.statuspage.io/

TOOLS4EVER

Badges

- **Tools4ever uses Badges to track certifications**
  - Badges are issued via the Acclaim platform
  - Each participant will receive an email from Acclaim to accept their Badge
  - An Acclaim account is needed to accept the Badges
  - Badges can be added to social media (e.g. LinkedIn) to allow earners to store and share their certification
  - Follow Acclaim instructions on how to share and select the right issuing organization (Tools4ever B.V.)