Entra ID Groups – Action1 Integration Connector

Configuration Guide

2929 Allen Parkway Suite 200,
Houston, TX 77019

www.action1.com

Contents

Summary

Entra ID Groups → Action1 Integration Connector is a utility that synchronizes device properties from Microsoft Entra ID into Action1.

The primary use case is synchronizing Entra ID device group membership into Action1 endpoint custom attributes, enabling dynamic grouping and automation in Action1.

The connector:

1. Reads devices from Entra ID
2. Retrieves Entra group membership per device
3. Matches devices to Action1 endpoints by name
4. Writes data to Action1 custom attributes
5. Devices are automatically placed into the correct Action1 groups

Device Matching Logic

Entra ID	Action1
Device Display Name	Action1 Device Name

Device names must match between Entra ID and Action1.

Configuration Model

The connector configuration follows this structure:

- Each Entra tenant acts as a source.
- Each Action1 organization acts as a target.

```
Source Entra ID Tenant
 └── Scope:
      Target Action1 Organizations
             └── Custom Attribute Mapping (where Entra data is written)
```

Supported Configuration Scenarios

1. One Entra Tenant → Multiple Action1 Organizations

A single Entra tenant is used as the source, and data is synchronized to multiple Action1 organizations.

```
Entra Tenant A
 ├── Action1 Org 1
 ├── Action1 Org 2
 ├──  …
 └── Action1 Org 10
```

2. Multiple Entra Tenants → Multiple Action1 Organizations *(Enterprise / MSP scenario)*

Multiple Entra tenants are supported, each with its own set of target Action1 organizations.

```
Entra Tenant A
 ├── Action1 Org 1
 ├── Action1 Org 2
 └── Action1 Org 3
Entra Tenant B
 ├── Action1 Org 4
 ├── Action1 Org 5
 └── Action1 Org 6
```

Important: An Action1 organization must not be assigned to more than one Entra tenant.

Custom Attribute Mapping Scenarios

Option 1. One Mapping for All Organizations within a scope

The same custom attribute names are used across all target organizations.

```
Entra Tenant A
 ├── Action1 Org 1
 ├── Action1 Org 2
 └── Action1 Org 3
     Custom Attribute: "Entra Groups"
```

Option 2. Individual Mapping per Organization

Different organizations use different custom attribute names.

```
Entra Tenant A
 ├── Scope 1: Action1 Orgs 1- 4
 │           Custom Attribute: "Entra Groups"
 ├── Scope 2: Action1 Org 5 – 6
 │    Custom Attribute: "Group Membership"
 └── Scope 3: Action1 Org 7 – 8
      Custom Attribute: "Department"
```

Important: Custom attribute names in Action1 must exactly match the names specified in config.json (case-sensitive).

Prerequisites

The connector is a Node.js–based application and can be executed on Windows, Linux, and macOS. This guide focuses on Windows-based configuration, which is the most common deployment scenario.

1.1 Install Node.js

- Download from: https://nodejs.org/en/download
- Select Latest LTS
- Keep default installer options

1.2 Download and Extract Action1 Integration Connector

- Download project archive (Code → Download ZIP)
- Extract to a permanent folder, for example:

```
C:\Tools\entra-action1-connector
```

Important: Do not place the connector in Desktop or Temporary folders

1.3 Install Connector Dependencies

- Open PowerShell
- Go to the extracted folder

```
C:\Tools\entra-action1-connector\
```

Run: npm ci

```
C:\Tools\entra-action1-connector\npm ci
```

1.4 Get Action1 API Credentials

- Action1 Console → Users and API Credentials
- Create New API Credentials
- Required permission:
    - Manage Endpoints Attributes
- Copy and Save Action1 API Credentials

Important: apiBaseUrl must match your Action1 region.

1.5 Configure Microsoft Graph API (Entra ID)

1.5.1 Open Microsoft Entra Admin Center

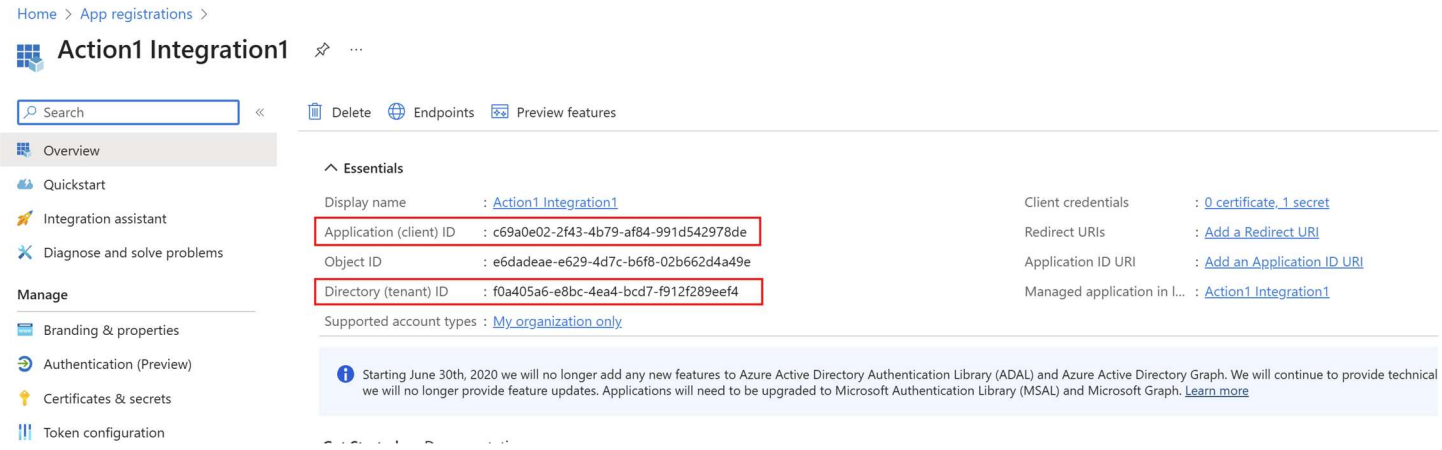
1.5.2 Go to App registrations → New registration

1.5.3 Name: Action1 Integration

1.5.4 Keep default options → Register

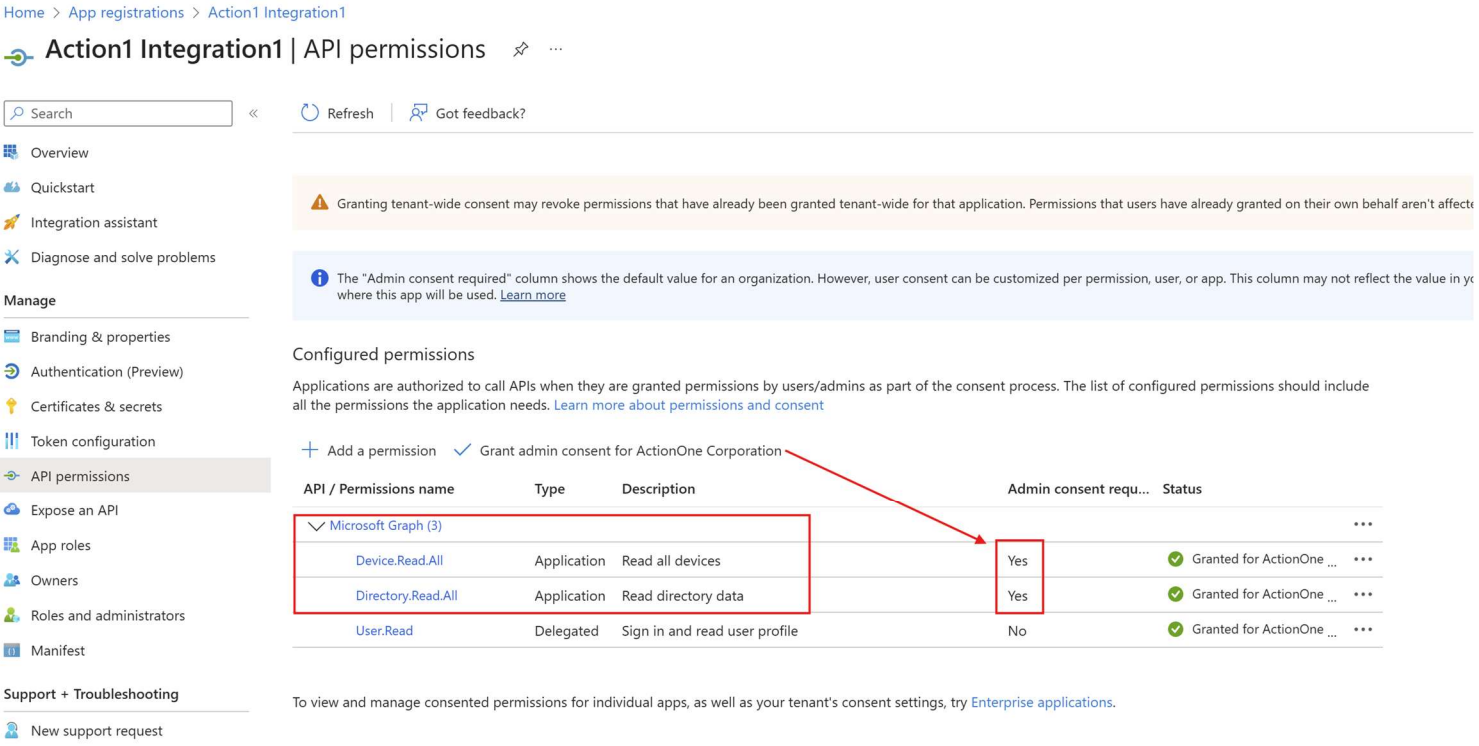
Copy and save:

- Application (client) ID
- Directory (tenant) ID



1.5.5 Set API Permissions for the App

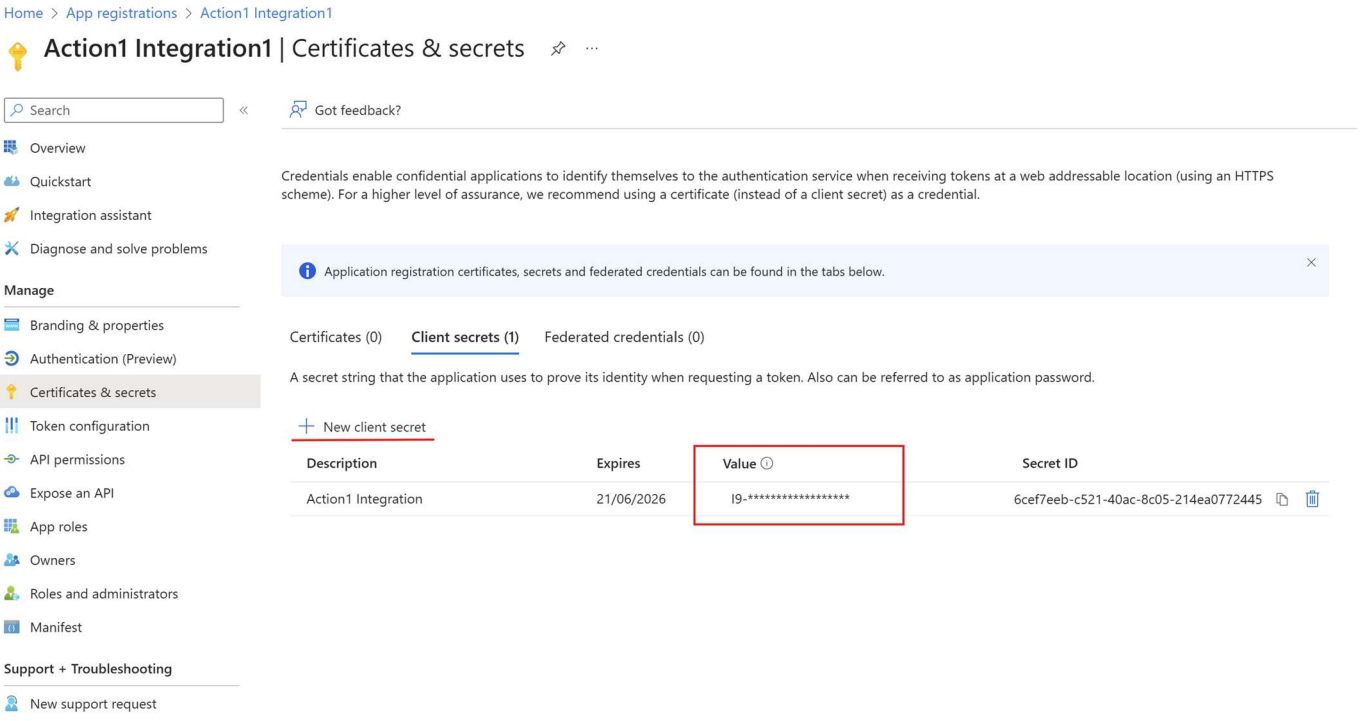
- API permissions → Add permission
- Microsoft Graph → Application permissions
- Add:
  - Device.Read.All
  - Directory.Read.All
- Grant Admin consent



1.5.6 Generate Client Secret

- Certificates & Secrets → New client secret
- Copy and Save Secret Value

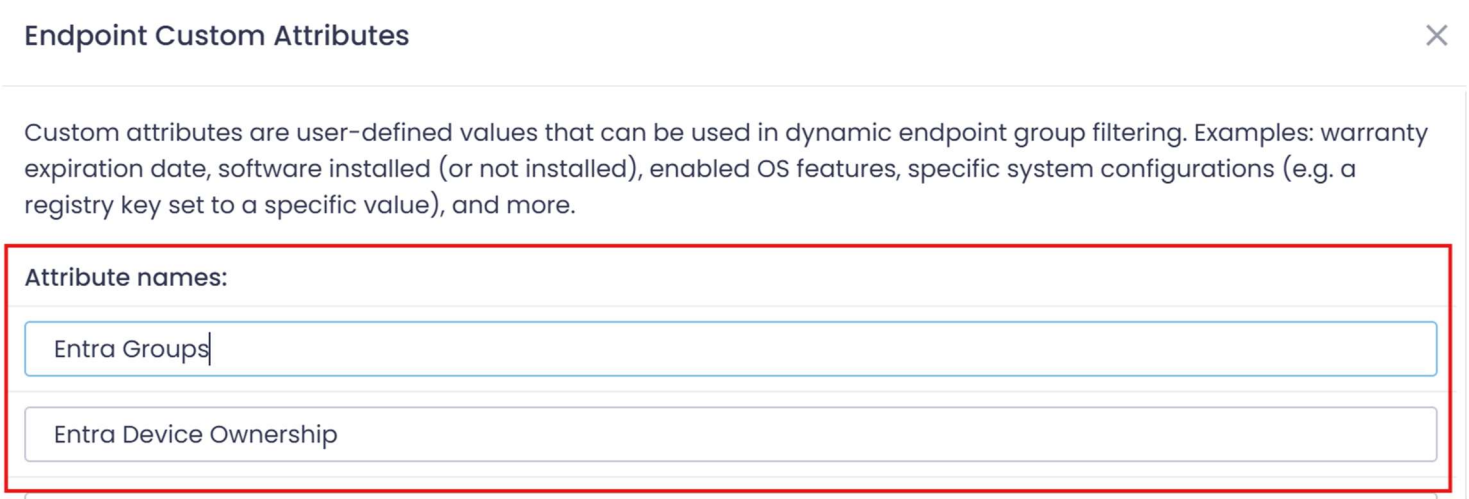
If you adding multiple Entra tenants repeat these steps for each Entra tenant you need to add.



1.5 Create Custom Attributes in Action1

- Open Action1 Console → Advanced
- For each target organization:
    - Create or rename Custom Attributes to match the following names exactly:
        - Entra Groups
        - Entra Device Ownership

Important: Attribute names are case-sensitive and must match exactly.



Configure Connector Settings

The connector is configured via a Command Line Interface (CLI).

In this example, the connector is extracted to:

```
C:\Tools\entra-action1-connector
```

Prerequisites

- Open PowerShell as Administrator
- Ensure script execution is allowed:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

2. Create Configuration File

Create an empty configuration template:

```
npm run config:init
```

This creates config.json in the connector folder.

3. Store Secrets (Windows Credential Manager)

The connector stores all secrets in Windows Credential Manager.

Important

Only the Windows account that stores the secret can later retrieve it.
If you plan to run the connector under a **different account** (for example, a scheduled task service account), **log in as that account** and store the secrets again.

3.1 Store Action1 API Secret

```
npm run secrets:set -- --ref action1:main
```

You will be prompted to enter the Action1 API secret.

Keep the reference name exactly as shown: action1:main.

3.2 Store Entra Tenant Secret

Run this command for each Entra tenant:

```
npm run secrets:set -- --ref entra:<tenant-ref>
```

Example:

```
npm run secrets:set -- --ref entra:acme-west
```

The reference name must be:

- unique
- lowercase
- no spaces

To verify stored secrets:

```
npm run secrets:list
```

4. Configure Action1 API Connection

Run

```
npm run config:action1:set -- \
  --base-url <action API URL> \
  --client-id <action1-client-id> \
  --secret-ref action1:main
```

Example:

```
npm run config:action1:set -- \
  --base-url https://app.action1.com/api/3.0 \
  --client-id api-key-xxxxx@action1.com \
  --secret-ref action1:main
```

5. Add Entra Tenant

This defines which Entra tenant is used as the source for Action1 organizations.

```
npm run config:tenant:add -- \
  --name "<Tenant Name>" \
  --tenant-id <tenant-guid> \
  --client-id <entra-app-id> \
  --secret-ref entra:<tenant-ref>
```

Example:

```
npm run config:tenant:add -- \
  --name "ACME Main" \
  --tenant-id g0a305a6-e8bc-4ea4-bcd7-f912f289eef4 \
  --client-id c23a0e02-2g43-4b79-af84-991d542978de \
  --secret-ref entra:acme-west
```

Repeat for each Entra tenant.

View configuration:

```
npm run config:show
```

Remove tenant:

```
npm run config:tenant:remove -- --tenant "<Tenant Name>"
```

6. Select Action1 Organizations

Select which Action1 organizations belong to a tenant:

```
npm run config:org:select -- --tenant "<Tenant Name>"
```

Example:

```
npm run config:org:select -- --tenant "ACME Main"
```

Select all organizations that should be synchronized.

To remove organizations from Tenant:

```
npm run config:org:unselect -- --tenant "<Tenant Name>"
```

7. Test Configuration (Recommended)

Review the resolved configuration:

```
npm run config:show:resolve -- --verbose
```

Run a dry-run sync:

```
npm run start
```

This will:

- Validate configuration
- Test API connectivity
- Not modify any data

8. Configure Scheduled Task (Windows)

8.1 Create run-connector.cmd

In the connector folder, open the run-connector.cmd file

```
@echo off
setlocal
cd /d "C:\Tools\entra-action1-connector"
node index.js --config ".\config.json" --apply
endlocal
```

Ensure the path matches the actual connector location.

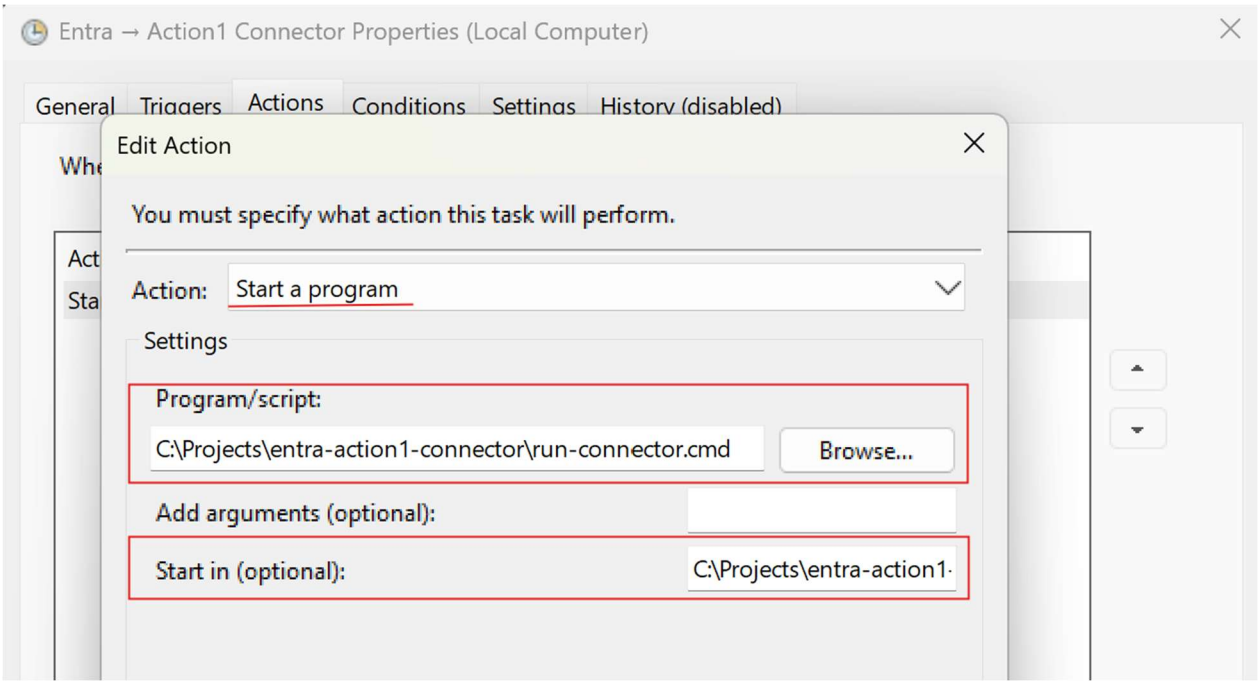
8.2 Create Scheduled Task

- Open Task Scheduler
- Click Create Task

Setting	**Value**
Task Name	Entra → Action1 Connector
Run Mode	Whether user is logged on or not
Triggers	Daily / Every X minutes / At startup

Task Actions Configuration

Field	Value
Action	Start a program
Program/Script	<Connector folder>\run-connector.cmd
Start In	<Connector folder>



8.3 Service Account

The scheduled task **must run under the same account** that stored the secrets.

Grant permissions to entra-action1-connector folder:

- Read
- Execute
- Modify
- Write

9. Full CLI Command Reference

```
docs/CLI.md
```