Install the use case's configuration files

A use case gives you two kinds of configuration files. Your Nuts node needs both, for every use case you join.

File	What it tells your node	Directory setting
Discovery Service definition	Where the use case's Discovery Service is, and which credentials a care organization must present to register on it	`discovery.definitions.directory`
Policy definition	Which credentials another party must present to get an access token from you, per scope	`policy.directory`

Do this once per use case, per environment.

Don't change these files. They contain the use case's trust rules, such as which certificate authorities are trusted. A changed file can make your node accept parties it shouldn't.

Install the files

1. Download the Discovery Service definition and the policy definition for your environment from the use case's IG.
2. Put them in the directories set in `discovery.definitions.directory` and `policy.directory`. Use any file name that ends in `.json`; the node skips other files.
3. If you use Docker, mount the directories read-only.
4. Restart the node. It reads the files at startup. If it can't read a Discovery Service definition, it doesn't start; the log says which file.

Each environment has its own files. Only install the files for the environment the node runs in. A use case typically uses the same service ID and scope in all environments; with files for two environments, the node won't start.

360-graden-clientbeeld

The files are in the nl-360-ig repository:

Environment	Discovery Service definition	Policy definition	Discovery Server
Development	discovery_360_dev.json	policy_360_dev.json	`http://localhost:9091` (a local Discovery Server)

Environment	Discovery Service definition	Policy definition	Discovery Server
Test	discovery_360_test.json	policy_360_test.json	`https://development.nedap-nuts.nl`
Acceptance	discovery_360_acc.json	policy_360_acc.json	`https://staging.nedap-nuts.nl`
Production	discovery_360_prod.json	policy_360_prod.json	`https://nedap-nuts.nl`

All environments use the service ID `360-graden`.

What's in a Discovery Service definition

Part	What it means
`id`	The service ID. You use it in the Discovery Service API calls.
`endpoint`	The Discovery Server where registrations are stored
`presentation_max_validity`	How long a registration is valid, in seconds. Your node refreshes it before then.
`presentation_definition.input_descriptors`	The credentials a care organization must present to register
Fields with a `filter`	Requirements on a credential, e.g. which certificate authorities are trusted. This is why each environment has its own file.
Fields with an `id`	Values the node takes from the credentials. They show up as `fields` in search results. A pattern with a capture group, like `(\d+)`, returns only the captured part.

360-graden-clientbeeld

The test definition, shortened: some fields, like `organization_id`, `organization_city` and `fhir_base_url_stu3`, are left out. It requires an X509Credential from a trusted certificate authority, and a DiscoveryRegistrationCredential with the endpoints. The `organization_ura` pattern returns only the URA.

```
{
  "id": "360-graden",
  "endpoint": "https://development.nedap-nuts.nl/discovery/360-graden",
  "presentation_max_validity": 259200,
  "presentation_definition": {
```

```
    "input_descriptors": [
      {
        "id": "id_uzicert_x509credential",
        "constraints": {
          "fields": [
            { "path": ["$.type"], "filter": { "const": "X509Credential" } },
            { "path": ["$.issuer"], "filter": { "pattern":
"^did:x509:0:sha256:(GwlhBZ...|szqMaT...|svKzNx...)..." } },
            { "id": "organization_ura", "path": ["$.credentialSubject.san.otherName"],
"filter": { "pattern": "^[0-9.]+-\\d+-\\d+-S-(\\d+)-00\\.000-\\d+$" } },
            { "id": "organization_name", "path": ["$.credentialSubject.subject.O"] }
          ]
        }
      },
      {
        "id": "id_registration_parameters_credential",
        "constraints": {
          "fields": [
            { "path": ["$.type"], "filter": { "const": "DiscoveryRegistrationCredential"
} },
            { "id": "authorization_server_url", "path":
["$.credentialSubject.authServerURL"] },
            { "id": "fhir_base_url_r4", "path": ["$.credentialSubject.fhirR4BaseURL"],
"optional": true }
          ]
        }
      }
    ]
  }
}
```

What's in a policy definition

A policy definition maps a scope to the credentials another party must present to get an access token for it.

- The top-level key is the scope. Another party asks for it in its token request.
- Under the scope, `organization` lists the credentials the requesting care organization must present.

- The fields with an `id` show up in the token introspection result (see "Validate the access token").

360-graden-clientbeeld

The policy, shortened. The scope is `360-graden`. The requester must present an X509Credential and a NutsEmployeeCredential.

```
{
  "360-graden": {
    "organization": {
      "input_descriptors": [
        { "id": "id_uzicert_uracredential", "...": "X509Credential from a trusted CA, as
in the Discovery Service definition" },
        { "id": "id_employee", "...": "NutsEmployeeCredential with employee_name,
employee_role and employee_identifier" }
      ]
    }
  }
}
```

Revision #10
Created 2024-09-09 12:20:44 UTC by Rein Krul
Updated 2026-10-27 15:25:33 UTC by Rein Krul