

# How to Use WebAC Groups

In WebAC you can use the `acl:agentGroup` property of an [Authorization](#) to point to a resource that holds a list of usernames. This allows you to create and manage groups of users within Fedora, and to assign different permissions to different groups. This how-to will guide you through the process of creating a resource, creating an `agentGroup`, and limiting access to that resource through an ACL that references that `agentGroup`.

## Prerequisites

- a running Fedora repository
- `curl`

The commands in this guide assume that your Fedora repository is running at <http://localhost:8080/fcrepo>.

## Steps

1. Create these three files:

### foo.ttl

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<> dc:title "Hello, World!".
```

### group.ttl

```
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .

<> a vcard:Group;
    vcard:hasMember "testuser".
```

### acl.ttl

```
@prefix acl: <http://www.w3.org/ns/auth/acl#>.

<#groupRead> a acl:Authorization;
    acl:accessTo </fcrepo/rest/foo>;
    acl:agentGroup </fcrepo/rest/group>;
    acl:mode acl:Read.
```

2. Upload these resources into Fedora:

```
curl -X PUT http://localhost:8080/fcrepo/rest/foo -u fedoraAdmin:secret3 \
-H "Content-Type: text/turtle" --data-binary @foo.ttl
curl -X PUT http://localhost:8080/fcrepo/rest/group -u fedoraAdmin:secret3 \
-H "Content-Type: text/turtle" --data-binary @group.ttl
curl -X PUT http://localhost:8080/fcrepo/rest/foo/fcr:acl -u fedoraAdmin:secret3 \
-H "Content-Type: text/turtle" --data-binary @acl.ttl
```

3. Test that `testuser` can read the `foo` resource, while `adminuser` cannot:

```
curl -i http://localhost:8080/fcrepo/rest/foo -u testuser:password1
curl -i http://localhost:8080/fcrepo/rest/foo -u adminuser:password2
```

The first request should respond with **200 OK**, while the second should be **403 Forbidden**.

To allow `adminuser` to also read the `foo` resource, we can add `adminuser` to the members of the group.

1. Create `group.sparql` with the following contents:

#### group.sparql

```
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>

INSERT DATA { <> vcard:hasMember "adminuser" . }
```

2. Run this command to update the group and add adminuser to it:

```
curl -i -X PATCH http://localhost:8080/fcrepo/rest/group \
-u fedoraAdmin:secret3 \
-H "Content-Type: application/sparql-update" \
--data-binary @group.sparql
```

You should receive a **204 No Content** response on success.

1. Now you should be able to repeat the command from step 3 and successfully retrieve the `foo` resource as adminuser:

```
curl -i http://localhost:8080/fcrepo/rest/foo -u adminuser:password2
```

This time, you should get a **200 OK** response.

## Caveats for agentGroup Groups

- For it to be useful, the names listed in the `vcard:member` properties of an authorization need to be names that your authentication system will provide to Fedora. *Remember, Fedora does no authentication of its own.*
- The purpose of the `acl:agentGroup` groups is distinct from any group mechanism your existing authentication system may have (e.g., LDAP or ActiveDirectory groups). The groups provided by the authentication system would be passed to Fedora as security principals, which the WebAC module compares against the `acl:agent` property. In other words, externally defined groups are opaque to Fedora, thus it treats them as simple agents.

## Differences from 4.x

The WebAC implementation in Fedora 4.x used the `acl:agentClass` predicate to point to group listing resources, and those group listing resources were expected to have the class `foaf:Group` and identify their members using the `foaf:member` property. We have changed this implementation in Fedora 5.x to align with the ["Groups of Agents" section of the SOLID WebAC specification](#).