OAuth 2.0 User Access Token for Making Identity Governance REST API Requests

OSP = One SSO Provider
NAM = NetIQ Access Manager

Note: The various OAuth 2.0 endpoints described below can also be obtained from the OAuth/ OpenID Connect provider "metadata" found at the following location relative to the "issuer URI":

```
<issuer-uri>/.well-known/openid-configuration
```

Issuer URIs:

OSP: `http[s]://<server>[:port]/osp/a/idm/auth/oauth2`
NAM: `https://<server>/nidp/oauth/nam`

See Open ID Connect Discovery 1.0 for more information.

Obtaining the Initial Access Token

OAuth 2.0 Resource Owner Password Credentials Grant Request

1. Determine the OAuth 2.0 token endpoint for the authorization server:
   (a) OSP: `http[s]://<server>[:port]/osp/a/idm/auth/oauth2/token`
   (b) NAM: `https://<server>/nidp/oauth/nam/token`
2. Obtain the Identity Governance "iac" client identifier and client secret.
   (a) OSP
      i. The identifier is usually *iac* but can be changed with the configutil or configupdate utilities.
      ii. The client secret is the "service password" specified during installation but can be changed with the configutil or configupdate utilities.
   (b) NAM
      i. Open the Access Manager administrator console in a browser and navigate to the *OAuth & OpenID Connect* tab on the *IDPCluster* page. Select the *Client Applications* heading.
      ii. Click on the "View" icon under the "Actions" heading for the *Client Application* named *iac*.
      iii. Click on *Click to reveal*.
      iv. Copy the *Client ID* value and the *Client Secret* value.
      v. Ensure that *Resource Owner Credentials* appears in the *Grants Required* list. If not, edit the client definition and check the *Resource Owner Credentials* box, save the client definition, and update the IDP.
3. Obtain the user identifier and password of a user with the required privilege for the desired API endpoint.
4. Create an HTTP POST request with the following characteristics (see RFC 6749 section 4.3.1)
   (a) `Content-Type: application/x-www-form-urlencoded`
   (b) POST body:

```
grant_type=password&username=<user-id>&password=<user-password>&client_id=<iac-client-id>&client_secret=<iac-client-secret>
```

where the angle-bracket-delimited names are replaced with the client and user values obtained in the steps above.

5. Issue the request to the OAuth 2.0 token endpoint.
6. The JSON response will be similar to the following (see RFC 6749 section 4.3.3):

```
{
  "access_token":"eyJraWQiOiI0...",
  "token_type":"bearer",
  "expires_in":119,
  "refresh_token":"eyJhbGciOiJ..."
}
```

7. When issuing a REST request to an Identity Governance endpoint pass the access token value using an HTTP *Bearer* authorization header (see RFC 6750 section 2.1):

```
Authorization: Bearer <access-token>
```

Refresh Tokens

If the authorization server is configured to return an OAuth 2.0 refresh token in the JSON result of the Resource Owner Password Credential Grant request then the refresh token should be used to obtain additional access tokens after the currently-valid access token expires.

In addition, each refresh token issued on behalf of a user causes a "revocation entry" to be stored in an attribute on the user's LDAP object. Obtaining many refresh tokens without revoking previously-obtained, unexpired refresh tokens will eventually exceed the capacity of the attribute on the user's LDAP object and will result in errors.

Therefore, if a refresh token is obtained it must be revoked after it is no longer needed.

Access Token Request

1. Create an HTTP POST request with the following characteristics (see RFC 6749 section 6)
   a) `Content-Type: application/x-www-form-urlencoded`
   b) POST body

```
grant_type=refresh_token&refresh_token=<refresh-token>&client_id=<iac-client-id>&client_secret=<iac-client-secret>
```

   where the angle-bracket-delimited names are replaced with the obvious values.
2. Issue the request to the OAuth 2.0 token endpoint.
3. The JSON result will be similar to

```
{
  "access_token":"eyJraWQiO...",
```

```
  "token_type":"bearer",
  "expires_in":119
}
```

4. Use the new access token value in requests to Identity Governance REST endpoints as described above.

Refresh Token Revocation Request

1. Determine the OAuth 2.0 token revocation endpoint for the authorization server:
   a) OSP: `http[s]://<server>[:port]/osp/a/idm/auth/oauth2/revoke`
   b) NAM: `https://<server>/nidp/oauth/nam/revoke`
2. Create an HTTP POST request with the following characteristics (see RFC 7009 section 2.1)
   a) `Content-Type: application/x-www-form-urlencoded`
   b) POST body

```
token=<refresh-token>&client_id=<iac-client-id>&client_secret=<iac-client-secret>
```

3. Issue the request to the OAuth 2.0 revocation endpoint.