Configuring ClearQuest for Open ID Connect SSO

OpenID Connect 1.0 is a simple identity layer built on top of OpenID 2.0. An Identity Provider (IDP) serves for information about identity, authentication, and authorization. A service provider (SP) is a service that makes use of the identity services the IDP provides. As an example, the Jazz Authorization Server (JAS) functions as an IDP while the ClearQuest Web is the SP. You must configure the SP to use the IDP for identity services, and then configure IDP to allow the SP to use it (SP registration).

OpenID Connect 1.0 uses Lightweight Third-Party Authentication for storing identity information on the browser, but OpenID Connect 1.0 defines a full method of interaction between the IDP and SP. For example, when the SP (for example, ClearQuest Web) is accessed and the user's identity has not yet been determined, the SP will redirect to the authentication UI of the IDP (for example, JAS). Once authentication is complete, the IDP (JAS) will redirect back to the SP (CQ Web) and continue the login process for that service.

Configuring ClearQuest to use OpenID Connect for single sign-on (SSO) is a three-step process. The first step is to configure the ClearQuest Web WebSphere profile and the Open ID Connect ID provider interactively perform identity operations. The second step is to tell the ClearQuest database and web server that it should use OpenID Connect for SSO.

Configuring WebSphere for OpenID Connect

This document assumes that you already have ClearQuest® installed in IBM® WebSphere® and you also have an OpenID connect ID provider in the form of a Jazz™ Authorization Server or a basic WAS Liberty Profile with OpenID Connect "openidConnectServer-1.0" feature enabled.

The ClearQuest Web WebSphere profile MUST have administrative security enabled. It is easiest to do this if you create the profile with administrative security enabled. It is more of a challenge to add administrative security later. Administrative security protects the administrative functions of WebSphere.

1) Read, but do not execute the procedure in the following help topic link for setting up WAS as an OpenID Connect Relying Party (RP). Execute the steps below. The relying party runs in the same WebSphere profile as the service provider (for example, ClearQuest Web). You will be following these steps with any variations described below.
   https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/tsec_oidconfigure.html?cp=SSAW57_8.5.5%2F1-8-2-33-2-6

2) Add the interceptor class (step 1-4)

3) For custom TAI properties, reference this link (step 5-6)
https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/csec_oidprop.html?cp=SSAW57_8.5.5%2F1-8-2-33-2-6-0

Example settings where JAS is on host adamsweb port 9447.

provider_1.identifier = libop
provider_1.interceptedPathFilter = /snoop,/cqweb.*
provider_1.clientId = CQWEB_ADAMSWEB
provider_1.clientSecret = bighit
provider_1.authorizeEndpointUrl = https://adamsweb:9447/oidc/endpoint/OP/authorize
provider_1.tokenEndpointUrl = https://adamsweb:9447/oidc/endpoint/OP/token
provider_1.introspectEndpoint = https://adamsweb:9447/oidc/endpoint/OP/introspect
provider_1.scope = openid general
provider_1.signatureAlgorithm = HS256

4) Add the com.ibm.websphere.security.InvokeTAIbeforeSSO (step 7-8)

5) Import the OpenID Connect providers (Liberty server, or JAS servers) certificate (step 9). To get the certificate, go to one of the sites used in the provider set up (step 3), for example, https://adamsweb:9447/oidc/endpoint/OP/authorize
then click on the lock to get certificate info. Click **More Information**, then **View Certificate**. Select **Details**, then **Export.** Export as X.509 Certificate with chain (PEM) (*.crt).
When importing the certificate, you have to go to **SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates**, and select **Binary DER data** for the Date Type. For Alias, give it a name. For **File name**, you must provide the full path to the certificate you exported.

Alternate method to get the certificate imported: Follow “step 9” on link given in step 1 of this document. In the IBM console go here to **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates**. Click **Retrieve from port**, then enter your OpenID Connect providers host and port (full URL not required). Enter a name for the certificate. Websphere will then fetch the certificate and import it.

6) Install the OpenID Connect RP app (step 10). First, change directory (cd) into the WebSphere AppServer bin directory (not the profile bin directory). For an AppServer with one plain profile created for development the commands are similar to

```
cd <AppServerRoot>/bin
wsadmin -f installOIDCRP.py install yourNodeName server1
```

When it completes, one the last lines will be:
ADMA5013I: Application WebSphereOIDCRP installed successfully.

7) Restart WebSphere Application Server.

ClearQuest SSO Setup

ClearQuest configuration for SSO follows almost the same directions as in the help topic Configuring strong authentication with smart cards.

The steps to set up SSO with LTPA are

1) Configure ClearQuest database for SSO.
2) Configure ClearQuest Web server for SSO.
3) Map OpenID Connect users to CQ Web application.

Configure ClearQuest database for SSO

1) You must set an SSO password in the database. See the procedure in help topic Configuring ClearQuest databases for container authentication.
2) Create an sso.properties file using cqrpc/cqrpc.exe and the password you used in step 1. See the procedure in help topic Configuring ClearQuest Web server for container authentication.
   Configure the sso.properties file for OpenID Connect. See the procedure in the help topic Configuring the ClearQuest Web client for container authentication.
   For the sso.properties, make these additional changes to the SSO_LOGIN_MODE and SSO_USES_JAS_OIDC.
   SSO_LOGIN_MODE=OIDC
   SSO_USES_JAS_OIDC=true
   The first line tells CQ Web that we will be using OpenID Connect. The second line says that CQ Web will integrate with a Jazz server that also uses OpenID Connect. The last line is only needed for CQ/Jazz integrations.
3) Follow most of the steps for setting up CQ for smart card authentication, but leave out steps 4 and 5.

Configure ClearQuest Web server for SSO

Reference the help topic Configuring client certificate authentication for ClearQuest Web to modify the web.xml descriptor, but for the security constraint and other clauses in web.xml use this:

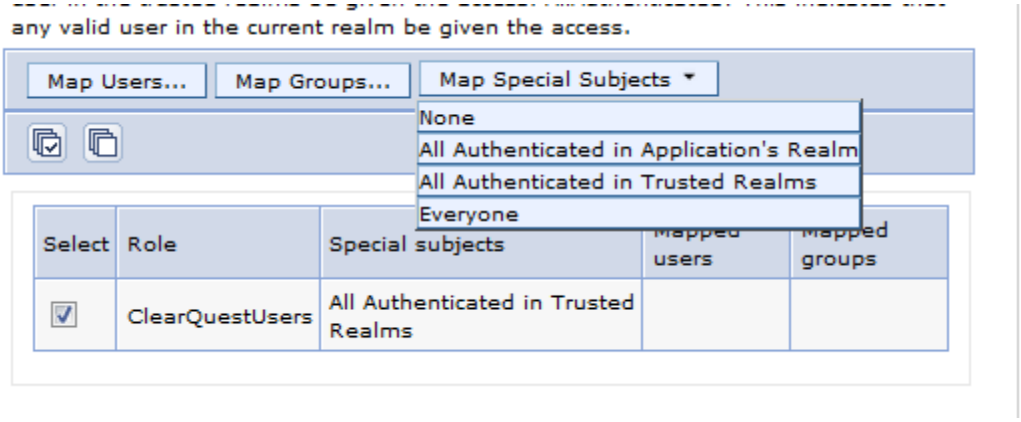
```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>secure</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>ClearQuestUsers</role-name>
    </auth-constraint>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>

<security-constraint>
    <web-resource-collection>
        <web-resource-name>CQBridge</web-resource-name>
        <!-- <url-pattern>/oslc/*</url-pattern> -->
        <url-pattern>/oslc/repo/sso2/discovery</url-pattern>
        <url-pattern>/oauth-request-consumer/*</url-pattern>
        <url-pattern>/oauth-authorize-consumers/*</url-pattern>
        <url-pattern>/oauth-request-token/*</url-pattern>
        <url-pattern>/oauth-authorization/*</url-pattern>
        <url-pattern>/oauth-access-token/*</url-pattern>
        <url-pattern>/scripts/*</url-pattern>
        <url-pattern>/images/*</url-pattern>
        <url-pattern>/stylesheets/*</url-pattern>
        <url-pattern>/gadgets/*</url-pattern>
        <url-pattern>/cqquerywizard.cq</url-pattern>
        <url-pattern>/cqartifactdetails.cq</url-pattern>
        <url-pattern>/cqqueryresults.cq</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>

<security-role>
    <role-name>ClearQuestUsers</role-name>
</security-role>
```

Map OpenID Connect users to CQ Web application

The WebSphere profile hosting ClearQuest Web application must be told which users can access it. For now, we have chosen to allow all OpenID Connect authenticated users to access the CQ Web login page. Just accessing the application does not mean they can actually log in to ClearQuest Web, it only means that they can load the CQ Web resources in their browser. Whether they can actually log in to a ClearQuest database depends on whether the user is already subscribed to the ClearQuest database. For example, if the user "tom" is allowed access to the CQ Web application but does not have any access to a CQ database, they will probably only see the database selection dialog but will not be able to actually connect to the database.

Any user who has not yet logged in to the OpenID Connect SSO will be redirected to the IDP and asked to login. Once they have successfully logged in, they will be taken back to the CQ link they were trying to load. Follow the steps below to grant all OpenID Connect authenticated users access to the CQ Web resources.

1) Using the WebSphere Application Server administrative console, click **Applications > Application types > WebSphere enterprise applications** in the navigation pane. The Enterprise Applications page opens.
2) In the **Enterprise Applications** table, click **TeamEar**. The Configuration page opens.
3) In the **Detail Properties** section, click **Security role to user/group mapping**.
4) In the table row in which the ClearQuestUsers role displays, select the check box and click on **Map Special Subjects** and select **All Authenticated in Trusted Realms**.
5) Restart the WebSphere Application Server.



Register ClearQuest Web Relying Party on the Identity Provider

For an OpenID Connect RP to make use of the IDP services, it must be registered on the IDP. How the RP is registered depends on which IDP you are using. Liberty-based IDPs support two ways for registering RP clients. You can add them to the server.xml in a localStore, or you can use a RESTful API for registering clients in a databaseStore. See the following help topic to learn about how to register clients:

- http://www.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/twlp_config_oidc_op.html
- http://www.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/twlp_client_registration.html

If you use Jazz Authorization Server (JAS), which is based on Liberty, then you can use the RESTful API described above, or command lines tools for registering clients. To use the command line, ensure that JAS is running first and then follow these steps

1) Create a JSON file with the client registration JSON in it. Name the file cqwebreg.json. You can use the same JSON that you would use for the RESTful registration described above: for example,

```
{
   "client_secret":"mysecret",
   "client_id":"CQWEB",
   "client_name":"CQWEB",
   "token_endpoint_auth_method":"client_secret_basic",
   "preauthorized_scope" : "openid profile email general phone address",
   "scope" : "openid profile scope1 email general phone address",
   "grant_types":[
      "authorization_code",
      "client_credentials",
      "implicit",
      "refresh_token",
      "urn:ietf:params:oauth:grant-type:jwt-bearer"
   ],
   "response_types":[
      "code",
      "token",
      "id_token token"
   ],
   "application_type":"web",
   "subject_type":"public",
   "introspect_tokens":true,
   "redirect_uris":[
      "https://[CQWeb Host]:[CQWeb Port]/oidcclient/[CQ RP Identifier]"
   ]
}
```

Note: For parts in brackets [ ], you should supply values for your CQ Web server. The CQ RP Identifier is the provider identifier you gave in the custom properties for the RP TAI. In the example above, we used provider_1.identifier = libop, so here you would put “libop”. For example, https://host.mydomain.com:12443/oidcclient/libop

2) Open a shell and change to the cli directory in JAS, for example,

```
cd C:\IBM\JazzAuthServer\cli
```

3) Run the following command to register your CQ WEB RP, substituting the JAS admin user and password and using the full path to the cqwebreg.json you created in step 1.

```
ldclient.bat -u "user:password" cqwebreg.json
```

In response, you will see JSON output with the client information you just registered.

Conclusion

Once restarted, the ClearQuest Web application will use OpenID Connect for single sign-on.