

**Description:** This document describes the various new features for working with external identity providers using Open ID Connect (OIDC), SAML 2.0, and Azure Active Directory.

**General Requirements:** Basic knowledge of ICONICS security and Workbench. OpenID Connect (OIDC) or SAML 2.0 identity provider for web login functionality. Azure Active Directory for Azure Active Directory sync.

## Introduction

Version 10.96 adds compatibility with external identity providers and clients. There are three separate parts to this support.

### Log into ICONICS Security with an External Identity Provider

When configured to do so, users can log into ICONICS security via an external OpenID Connect (OIDC) or SAML 2.0 identity provider. Instead of logging into the ICONICS security login dialog, users will be directed to a custom login page or the login page of a compatible identity provider, log in with their credentials for that provider, and then ICONICS security will recognize them as an ICONICS user and grant them the appropriate permissions. This feature is referred to as "web login".

See the "An identity provider client, also known as an OIDC relying party, is an application that requires end-user authentication or wants to get access to something using the user's credentials. In the case of ICONICS, the client most often needs permission to access data points.

Third party clients that can connect to ICONICS security as an OIDC relying party include:

- Alexa
- Cortana
- Google Home
- Apple Watch

The connection to clients is configured under **Workbench > Security > OIDC Relying Parties**. Create a new OIDC relying party for your device type and fill in the configured information according to the client you are connecting.

For more information on the configuration for these devices and services, see one of these application notes:

- *GENESIS64 - Voice Machine Interface*
- *KPIWorX - Support for Apple Watch*

## Configuring Public Origin and Web Server Endpoints

Before configuring a web login, Azure Active Directory sync, or an OIDC Relying Party, we have to verify the configuration of the web server endpoints and the public origin. **This step is very important.** Having the wrong settings here could open your system to a security vulnerability or prevent a proper connection with external identity providers or clients.

You can reach these settings with these steps:

1. Open **Platform Services Configuration**, either from **Workbench > Tools** or from the **ICONICS Tools** folder in the start menu.
2. Go to the **Basic** tab and find the **Web server endpoints** section.

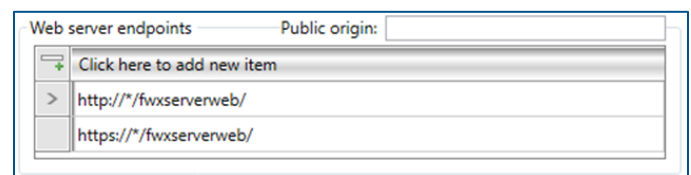


Figure 1 – Web Server Endpoints dialog

If you want your clients to use HTTPS instead of HTTP, or if you are running behind a reverse proxy server or your host name may not otherwise be able to be correctly determined automatically by ICONICS security, you may want to set the **Public origin**. The **Public origin** field is how ICONICS security advertises itself to identity providers and clients. It must contain only the protocol and host name, without the trailing slash. An example of a properly formatted **Public origin** is: **https://yourserver.com**

The **Web server endpoints** table lists all the URLs that the security service listens on. To properly secure your system, change the asterisk in the URLs to a specific DNS name, as an asterisk here creates security vulnerabilities. You may also wish to remove the HTTP endpoint for production environments.

It is not advisable to change the path in the URL from the default **/fwxserverweb/**.

An example of a properly secured endpoint is: **https://yourserver.com/fwserverweb/**

Security with an External OIDC or SAML Identity Provider" section for information on how to configure web login.

# Security - Working with Third Party Identity Providers and Clients



Application Note

November 2019

## Azure Active Directory Support

In addition to synchronizing with an in-house Active Directory, ICONICS security can now synchronize with Azure Active Directory. Web login support can also be enabled for Azure Active Directory, allowing users to log in via Microsoft's login screen. Using Microsoft's login screen allows you to use single-sign-on between Azure Active Directory and ICONICS Security and gives ICONICS security any additional features of the Azure Active Directory login, such as multi-factor authentication.

See the "Azure Active Directory Security Mode " section for more information on how to configure Azure Active Directory support.

## Provide Identity/Security to Third Party Clients

ICONICS Security can be an OIDC identity provider for third party clients.

An identity provider client, also known as an OIDC relying party, is an application that requires end-user authentication or wants to get access to something using the user's credentials. In the case of ICONICS, the client most often needs permission to access data points.

Third party clients that can connect to ICONICS security as an OIDC relying party include:

- Alexa
- Cortana
- Google Home
- Apple Watch

The connection to clients is configured under **Workbench > Security > OIDC Relying Parties**. Create a new OIDC relying party for your device type and fill in the configured information according to the client you are connecting.

For more information on the configuration for these devices and services, see one of these application notes:

- *GENESIS64 - Voice Machine Interface*
- *KPIWorX - Support for Apple Watch*

## Configuring Public Origin and Web Server Endpoints

Before configuring a web login, Azure Active Directory sync, or an OIDC Relying Party, we have to verify the configuration of the web server endpoints and the public origin. **This step is very important.** Having the wrong settings here could open

your system to a security vulnerability or prevent a proper connection with external identity providers or clients.

You can reach these settings with these steps:

3. Open **Platform Services Configuration**, either from **Workbench > Tools** or from the **ICONICS Tools** folder in the start menu.
4. Go to the **Basic** tab and find the **Web server endpoints** section.

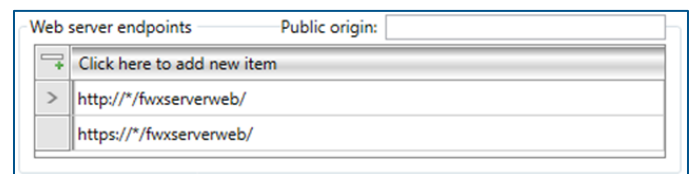


Figure 1 – Web Server Endpoints dialog

If you want your clients to use HTTPS instead of HTTP, or if you are running behind a reverse proxy server or your host name may not otherwise be able to be correctly determined automatically by ICONICS security, you may want to set the **Public origin**. The **Public origin** field is how ICONICS security advertises itself to identity providers and clients. It must contain only the protocol and host name, without the trailing slash. An example of a properly formatted **Public origin** is: **https://yourserver.com**

The **Web server endpoints** table lists all the URLs that the security service listens on. To properly secure your system, change the asterisk in the URLs to a specific DNS name, as an asterisk here creates security vulnerabilities. You may also wish to remove the HTTP endpoint for production environments.

It is not advisable to change the path in the URL from the default **/fwxserverweb/**.

An example of a properly secured endpoint is: **https://yourserver.com/fwxserverweb/**

## Security with an External OIDC or SAML Identity Provider

**Note:** Before configuring External OIDC or SAML identity provider, make sure you have followed the steps in the "Configuring Public Origin and Web Server Endpoints" section.

When using an external OIDC or SAML identity provider, the security login is redirected to the identity provider. The user

# Security - Working with Third Party Identity Providers and Clients



Application Note

November 2019

logs into the identity provider, which returns a token to ICONICS that identifies the successfully authenticated user.

The token contains information about the authenticated user. This information is referred to as a "claim". The claim might be the user's name, email address, a unique GUID, or other identifying detail.

Once ICONICS security receives the token, it matches the claim to an existing user in the ICONICS security configuration and logs in that user. The claim can be configured to match any of these attributes of an ICONICS security user:

- **Display Name**
- **Unique Name** – For Active Directory and Azure Active Directory modes, this is the user's GUID in string form.

**Note:** For database mode, users also have a unique GUID assigned, but since this GUID is not visible in Workbench, the GUID is not practical to use as the matching attribute. ICONICS is considering improvements in a future version that may make this attribute more feasible to use with database mode.

- **Active Directory SID** – This attribute is only available in Active Directory mode, or in Azure Active Directory mode as long as Azure Active Directory is synchronizing with an on-premise Active Directory.
- **User Lookup Identifier** – A custom text field that can be used for claims that don't match an existing field, such as email address.

It's important to note that the external identity provider does not tell ICONICS what the user can and cannot do. It only tells ICONICS security who the user is, and it is up to ICONICS security to then apply the appropriate permissions granted to that user.

In order to make the connection between the claim (identifying information from the identity provider) and the permissions the user should be granted, ICONICS security needs to be configured with the list of users, identified by their claims. Permissions are then assigned to those users or their groups via the standard ICONICS security configuration.

ICONICS security does not automatically pull in the list of users from an external identity provider. The list of users can be synchronized with Active Directory or Azure Active Directory, populated with a CSV import, or manually entered in Workbench.

## Preparing the External Identity Provider

The first thing you need when connecting to an external identity provider is that identity provider! An example identity provider is Okta (<https://developer.okta.com/>).

This document will not describe in detail how to configure an identity provider, as this varies from provider to provider and may change over time as the provider makes updates, but here is a quick description of how to get started with Okta and what information you need to use Okta as the identity provider for ICONICS:

**Note:** These instructions may change over time as Okta releases updates. Please consult Okta's documentation if you are having trouble finding these settings.

1. Register on the Okta website and take a look at the sample configuration that is generated for new users.
2. If you do not have a user named John Doe, create one and give him a password you will remember.
3. Under **API > Authorization Servers**, take note of your **Issuer URI**, which might look something like this:

<https://dev-randomizedid.okta.com/oauth2/default>

4. Go to **Applications**.
5. Create a **Web Application Platform** with these details:
  - a. **Login URL:**

<https://yourserver.com/fwserverweb/security/signin-oidc>

- b. **Logout URL:**

<https://yourserver.com/fwserverweb/security/signout-callback-oidc>

6. Take note of your **Client ID** and **Client secret**.

## Configuring Web Login

This section will walk through configuring the security Web Login settings with information about our external identity provider and configuring a user to match the claims from that identity provider.

This example assumes we are using the Okta sample configuration, or another identity provider that contains a user named "John Doe". We will be using the full name of the Okta users as our claim and matching it against the display name of our ICONICS users.

# Security - Working with Third Party Identity Providers and Clients



Application Note

November 2019

Some of these settings may vary for different identity providers, or if Okta updates their system or default configuration.

1. Open **Workbench**.
2. Go to **Security > Users** and create a new user.
3. Name it, **John Doe** and apply the changes.
4. Go to **Security > Global Settings**.
5. On the **General** tab, set **Security active** to **Testing**.

**Note:** Testing mode applies permissions when a user is logged in, but when no user is logged in all permissions are allowed, similar to when security is inactive. This mode is for *testing only*. Remember to set **Security active** to **Active** before putting a production system live!

6. Uncheck **Allow simultaneous login**.
7. Go to the **Web Login** tab.
8. Check **Enabled** in the **General Settings** section.

**Note:** When Web Login is set to **Disabled**, your HTTP endpoints for OIDC/OAuth would remain functional but they would not allow any authentications. They will simply return error codes or display error messages (depending on endpoint) when used.

9. Set **Signing credentials type** to **Auto-generated temporary key**.
10. Change **Authentication Type** to **OpenID Connect**.
11. Fill the **Issuer URL**, **Client ID**, and **Client Secret** with the information you noted from your identity provider.
12. Check **Use PKCE**.
13. For **Prompt**, check only **Default**. **Prompt** is an OIDC parameter that tells the identity provider what information it should prompt the user for, and **Default** means to use the identity provider's default settings.
14. For **OIDC scope to request**, choose **profile**.
15. For **Use this OIDC claim**, enter **name**.
16. For **Find existing users by**, choose **Display name**.
17. Apply your changes.
18. Wait a moment for the changes to propagate, then test your connection by visiting the hyperlink next to **Show list of claims**. If this list populates, you have a good connection and can proceed to the next step.
19. Go to the **General Settings** section and check **In-house applications use web login**. This will forward any login requests from the built-in ICONICS dialog to the identity provider's login page in a browser.
20. Apply your changes.
21. Open a login dialog. The dialog should automatically forward you to the Okta login page where you can log in as **John Doe**.

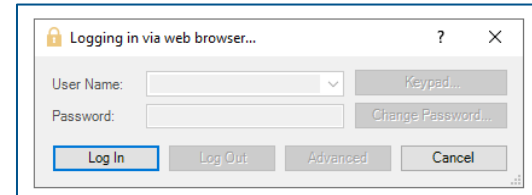


Figure 2 – Login dialog forwarding to web login page

## Azure Active Directory Security Mode

**Note:** Before configuring Azure Active Directory security mode, make sure you have followed the steps in the "Configuring Public Origin and Web Server Endpoints" section.

Version 10.96 adds a new security mode. In addition to Database and Active Directory modes, users can now use Azure Active Directory mode to synchronize with an Azure Active Directory.

Before ICONICS security can sync with Azure Active Directory, Azure Active Directory needs to be configured properly.

**Note:** These steps may be subject to change as Microsoft releases updates. Please consult the Azure documentation if you are having trouble finding these settings.

1. Create an Active Directory.
2. Register your application on the Azure portal. More info about the registration process can be found here: <https://docs.microsoft.com/en-us/graph/auth-v2-service>
3. Under **API permission**, add a new permission: **Microsoft Graph > Application Permission > Directory > Directory.Read.All**.
4. Grant the application at least one **Delegated API permission**, either **User.Read** or **openid**.
5. Still on the **API permission** page, click the button **Grant admin consent for directory name**. This step is necessary and requires administrative privileges to the Azure Active Directory.
6. If you want users to be able to use the standard login dialog in GENESIS64, go to **Authentication** and enable **Treat application as a public client**.
7. Now we need to collect some data from Azure to use when configuring Workbench. Write down this information, which we will use in a moment.

# Security - Working with Third Party Identity Providers and Clients



Application Note

November 2019

## Settings

## Location in Azure

Application (client) ID

**App Registration > your app > Overview** under **Active Directory**

Directory (tenant) ID

**App Registration > your app > Overview** under **Active Directory**

Client secret

Created under **App Registration > your app > Certificates & Secrets > New client secret**

Initial administrator account

Name under **Owners > Username\***

**\* Note:** The initial administrator account name should be in the format of `user@domainname.onmicrosoft.com` for internal domain users, and in the format of `user_domainname.com#EXT#@domainname.onmicrosoft.com` for external users.

8. Open **Workbench** and browse to **Security > Global Settings**.
9. Go to the **General** tab.
10. For **Security active**, choose **Testing**.

**Note:** Testing mode applies permissions when a user is logged in, but when no user is logged in all permissions are allowed, similar to when security is inactive. This mode is for *testing only*. Remember to set **Security active** to **Active** before putting a production system live!

11. For **Security mode**, select **Azure Active Directory**.
12. In the **Azure Active Directory Settings** section, fill in the information you noted above in step 7.
13. Select **Synchronize now**. After a moment you should see an **Operation Successful** message next to the button.

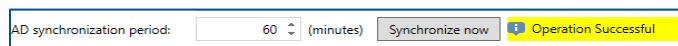


Figure 3 – Azure Active Directory successful synchronization

14. Open the login dialog and test logging into security with an Azure Active Directory user. If this test succeeds, we can proceed with configuring the web login.

**Note:** Web login is an optional step. Without configuring it, you will still be able to log into the standard ICONICS login dialog with your Azure email and password, however you will not get the benefits of the Azure login dialog, such as single-sign-on and multi-factor authentication.

15. Go back to **Security > Global Settings**. Web login requires simultaneous login to be turned off, so uncheck **Allow simultaneous login** on the **General** tab.
16. Go to the **Web Login** tab.
17. Check the **Enabled** box.
18. Under **Authentication Type**, select **Azure Active Directory**.
19. If you would like to use Microsoft's page to log in, check **In-house application use web login**.
20. If you checked **In-house application use web login**, it is highly recommended to go to the **General** tab and uncheck **Automatic login**.
21. Apply your changes.
22. Go back to the **Azure portal > Active Directory > App Registration > your app > Authentication** and fill in your OIDC URI:  
`https://servername/fwxserverweb/security/signin-oidc`

Test your login again. If everything is configured correctly you should be able to login using Microsoft's login page.