**Set up sign-in with an Oracle SAML provider by using SAML protocol in Azure Active Directory B2C**

Top of Form

Bottom of Form

***Before you begin***, use the selector above to choose the type of policy you’re configuring. Azure AD B2C offers two methods of defining how users interact with your applications: through predefined [user flows](https://docs.microsoft.com/en-us/azure/active-directory-b2c/user-flow-overview), or through fully configurable [custom policies](https://docs.microsoft.com/en-us/azure/active-directory-b2c/custom-policy-overview). The steps required in this article are different for each method.

Note

In Azure Active Directory B2C, [custom policies](https://docs.microsoft.com/en-us/azure/active-directory-b2c/user-flow-overview) are designed primarily to address complex scenarios. For most scenarios, we recommend that you use built-in [user flows](https://docs.microsoft.com/en-us/azure/active-directory-b2c/user-flow-overview).

This article shows you how to enable sign-in for users from a Oracle organization using [custom policies](https://docs.microsoft.com/en-us/azure/active-directory-b2c/custom-policy-overview) in Azure Active Directory B2C (Azure AD B2C). You enable sign-in by adding a [SAML identity provider](https://docs.microsoft.com/en-us/azure/active-directory-b2c/identity-provider-generic-saml) to a custom policy.

**Prerequisites**

* Complete the steps in [Get started with custom policies in Active Directory B2C](https://docs.microsoft.com/en-us/azure/active-directory-b2c/tutorial-create-user-flows?pivots=b2c-custom-policy).
* If you haven't already, [register a web application](https://docs.microsoft.com/en-us/azure/active-directory-b2c/tutorial-register-applications) and [enable the ID token implicit grant](https://docs.microsoft.com/en-us/azure/active-directory-b2c/tutorial-register-applications#enable-id-token-implicit-grant).
* If you haven't already done so, sign up for a [free Developer Edition account](https://www.oracle.com/cloud/sign-in.html).

**Set up Oracle as an identity provider**

1. [Sign in to Oracle](https://www.oracle.com/cloud/sign-in.html) and refer :- <https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/federatingADFSazure.htm>
2. For the store front - https://docs.oracle.com/en/cloud/saas/cx-commerce/20c/ccdev/configure-storefront-sso.html
3. On the left menu, under **Settings**, expand **Identity**, and then select **Identity Provider**.
4. Select **Enable Identity Provider**.
5. Under **Select the certificate**, select the certificate you want Oracle to use to communicate with Azure AD B2C. You can use the default certificate.
6. Click **Save**.

**Create a connected app in Oracle**

1. On the **Identity Provider** page, select **Service Providers are now created via Connected Apps. Click here.**
2. Under **Basic Information**, enter the required values for your connected app.
3. Under **Web App Settings**, check the **Enable SAML** box.
4. In the **Entity ID** field, enter the following URL. Make sure that you replace the value for your-tenant with the name of your Azure AD B2C tenant.

https://your-tenant.b2clogin.com/your-tenant.onmicrosoft.com/B2C\_1A\_TrustFrameworkBase

When using a [custom domain](https://docs.microsoft.com/en-us/azure/active-directory-b2c/custom-domain), use the following format:

 https://your-domain-name/your-tenant.onmicrosoft.com/B2C\_1A\_TrustFrameworkBase

 In the **ACS URL** field, enter the following URL. Make sure that you replace the value for your-tenant with the name of your Azure AD B2C tenant.

https://your-tenant.b2clogin.com/your-tenant.onmicrosoft.com/B2C\_1A\_TrustFrameworkBase/samlp/sso/assertionconsumer

When using a [custom domain](https://docs.microsoft.com/en-us/azure/active-directory-b2c/custom-domain), use the following format:

1. https://your-domain-name/your-tenant.onmicrosoft.com/B2C\_1A\_TrustFrameworkBase/samlp/sso/assertionconsumer
2. Scroll to the bottom of the list, and then click **Save**.

**Get the metadata URL**

1. On the overview page of your connected app, click **Manage**.
2. Copy the value for **Metadata Discovery Endpoint**, and then save it. You'll use it later in this article.

**Set up Oracle users to federate**

1. On the **Manage** page of your connected app, click **Manage Profiles**.
2. Select the profiles (or groups of users) that you want to federate with Azure AD B2C. As a system administrator, select the **System Administrator** check box, so that you can federate by using your Oracle account.

**Create a self-signed certificate.**

If you don't already have a certificate, you can use a self-signed certificate. A self-signed certificate is a security certificate that is not signed by a certificate authority (CA) and doesn't provide the security guarantees of a certificate signed by a CA.

* [Windows](https://docs.microsoft.com/en-us/azure/active-directory-b2c/identity-provider-salesforce-saml?tabs=windows&pivots=b2c-custom-policy#tabpanel_1_windows)
* [macOS](https://docs.microsoft.com/en-us/azure/active-directory-b2c/identity-provider-salesforce-saml?tabs=windows&pivots=b2c-custom-policy#tabpanel_1_macos)

On Windows, use PowerShell's [New-SelfSignedCertificate](https://docs.microsoft.com/en-us/powershell/module/pki/new-selfsignedcertificate) cmdlet to generate a certificate.

1. Execute this PowerShell command to generate a self-signed certificate. Modify the -Subject argument as appropriate for your application and Azure AD B2C tenant name. You can also adjust the -NotAfter date to specify a different expiration for the certificate.

PowerShell

1. New-SelfSignedCertificate `
2. -KeyExportPolicy Exportable `
3. -Subject "CN=yourappname.yourtenant.onmicrosoft.com" `
4. -KeyAlgorithm RSA `
5. -KeyLength 2048 `
6. -KeyUsage DigitalSignature `
7. -NotAfter (Get-Date).AddMonths(12) `
8. -CertStoreLocation "Cert:\CurrentUser\My"
9. Open **Manage user certificates** > **Current User** > **Personal** > **Certificates** > *yourappname.yourtenant.onmicrosoft.com*.
10. Select the certificate, and then select **Action** > **All Tasks** > **Export**.
11. Select **Yes** > **Next** > **Yes, export the private key** > **Next**.
12. Accept the defaults for **Export File Format**.
13. Provide a password for the certificate.

For Azure AD B2C to accept the .pfx file password, the password must be encrypted with the TripleDES-SHA1 option in Windows Certificate Store Export utility as opposed to AES256-SHA256.

**Create a policy key**

You need to store the certificate that you created in your Azure AD B2C tenant.

1. Sign in to the [Azure portal](https://portal.azure.com/).
2. Make sure you're using the directory that contains your Azure AD B2C tenant by selecting the **Directory + subscription** filter in the top menu and choosing the directory that contains your tenant.
3. Choose **All services** in the top-left corner of the Azure portal, and then search for and select **Azure AD B2C**.
4. On the Overview page, select **Identity Experience Framework**.
5. Select **Policy Keys** and then select **Add**.
6. For **Options**, choose Upload.
7. Enter a **Name** for the policy. For example, SAMLSigningCert. The prefix B2C\_1A\_ is automatically added to the name of your key.
8. Browse to and select the B2CSigningCert.pfx certificate that you created.
9. Enter the **Password** for the certificate.
10. Click **Create**.

**Add a claims provider**

If you want users to sign in using a Oracle account, you need to define the account as a claims provider that Azure AD B2C can communicate with through an endpoint. The endpoint provides a set of claims that are used by Azure AD B2C to verify that a specific user has authenticated.

You can define a Oracle account as a claims provider by adding it to the **ClaimsProviders** element in the extension file of your policy. For more information, see [define a SAML identity provider](https://docs.microsoft.com/en-us/azure/active-directory-b2c/identity-provider-generic-saml).

1. Open the *TrustFrameworkExtensions.xml*.
2. Find the **ClaimsProviders** element. If it does not exist, add it under the root element.
3. Add a new **ClaimsProvider** as follows:

XML

 <ClaimsProvider>

<Domain>Oracle.com</Domain>

<DisplayName>Oracle</DisplayName>

<TechnicalProfiles>

<TechnicalProfile Id="Oracle-SAML2">

<DisplayName>Oracle</DisplayName>

<Description>Login with your Oracle account</Description>

<Protocol Name="SAML2"/>

<Metadata>

<Item Key="WantsEncryptedAssertions">false</Item>

<Item Key="WantsSignedAssertions">false</Item>

<Item Key="PartnerEntity">https://contoso-dev-ed.my.Oracle.com/.well-known/samlidp.xml</Item>

</Metadata>

<CryptographicKeys>

<Key Id="SamlMessageSigning" StorageReferenceId="B2C\_1A\_SAMLSigningCert"/>

</CryptographicKeys>

<OutputClaims>

<OutputClaim ClaimTypeReferenceId="issuerUserId" PartnerClaimType="userId"/>

<OutputClaim ClaimTypeReferenceId="givenName" PartnerClaimType="given\_name"/>

<OutputClaim ClaimTypeReferenceId="surname" PartnerClaimType="family\_name"/>

<OutputClaim ClaimTypeReferenceId="email" PartnerClaimType="email"/>

<OutputClaim ClaimTypeReferenceId="displayName" PartnerClaimType="username"/>

<OutputClaim ClaimTypeReferenceId="authenticationSource" DefaultValue="socialIdpAuthentication"/>

<OutputClaim ClaimTypeReferenceId="identityProvider" DefaultValue="Oracle.com" />

</OutputClaims>

<OutputClaimsTransformations>

<OutputClaimsTransformation ReferenceId="CreateRandomUPNUserName"/>

<OutputClaimsTransformation ReferenceId="CreateUserPrincipalName"/>

<OutputClaimsTransformation ReferenceId="CreateAlternativeSecurityId"/>

<OutputClaimsTransformation ReferenceId="CreateSubjectClaimFromAlternativeSecurityId"/>

</OutputClaimsTransformations>

<UseTechnicalProfileForSessionManagement ReferenceId="SM-Saml-idp"/>

</TechnicalProfile>

</TechnicalProfiles>

</ClaimsProvider>

 Update the value of **PartnerEntity** with the Oracle metadata URL you copied earlier.

 Update the value of both instances of **StorageReferenceId** to the name of the key of your signing certificate. For example, B2C\_1A\_SAMLSigningCert.

 Locate the <ClaimsProviders> section and add the following XML snippet. If your policy already contains the SM-Saml-idp technical profile, skip to the next step. For more information, see [single sign-on session management](https://docs.microsoft.com/en-us/azure/active-directory-b2c/custom-policy-reference-sso).

XML

1. <ClaimsProvider>
2. <DisplayName>Session Management</DisplayName>
3. <TechnicalProfiles>
4. <TechnicalProfile Id="SM-Saml-idp">
5. <DisplayName>Session Management Provider</DisplayName>
6. <Protocol Name="Proprietary" Handler="Web.TPEngine.SSO.SamlSSOSessionProvider, Web.TPEngine, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
7. <Metadata>
8. <Item Key="IncludeSessionIndex">false</Item>
9. <Item Key="RegisterServiceProviders">false</Item>
10. </Metadata>
11. </TechnicalProfile>
12. </TechnicalProfiles>
13. </ClaimsProvider>
14. Save the file.

**Add a user journey**

At this point, the identity provider has been set up, but it's not yet available in any of the sign-in pages. If you don't have your own custom user journey, create a duplicate of an existing template user journey, otherwise continue to the next step.

1. Open the *TrustFrameworkBase.xml* file from the starter pack.
2. Find and copy the entire contents of the **UserJourney** element that includes Id="SignUpOrSignIn".
3. Open the *TrustFrameworkExtensions.xml* and find the **UserJourneys** element. If the element doesn't exist, add one.
4. Paste the entire content of the **UserJourney** element that you copied as a child of the **UserJourneys** element.
5. Rename the Id of the user journey. For example, Id="CustomSignUpSignIn".

**Add the identity provider to a user journey**

Now that you have a user journey, add the new identity provider to the user journey. You first add a sign-in button, then link the button to an action. The action is the technical profile you created earlier.

1. Find the orchestration step element that includes Type="CombinedSignInAndSignUp", or Type="ClaimsProviderSelection" in the user journey. It's usually the first orchestration step. The **ClaimsProviderSelections** element contains a list of identity providers that a user can sign in with. The order of the elements controls the order of the sign-in buttons presented to the user. Add a **ClaimsProviderSelection** XML element. Set the value of **TargetClaimsExchangeId** to a friendly name.
2. In the next orchestration step, add a **ClaimsExchange** element. Set the **Id** to the value of the target claims exchange Id. Update the value of **TechnicalProfileReferenceId** to the Id of the technical profile you created earlier.

The following XML demonstrates the first two orchestration steps of a user journey with the identity provider:

XML

<OrchestrationStep Order="1" Type="CombinedSignInAndSignUp" ContentDefinitionReferenceId="api.signuporsignin">

<ClaimsProviderSelections>

...

<ClaimsProviderSelection TargetClaimsExchangeId="OracleExchange" />

</ClaimsProviderSelections>

...

</OrchestrationStep>

<OrchestrationStep Order="2" Type="ClaimsExchange">

...

<ClaimsExchanges>

<ClaimsExchange Id="OracleExchange" TechnicalProfileReferenceId="Oracle-SAML2" />

</ClaimsExchanges>

</OrchestrationStep>

**Configure the relying party policy**

The relying party policy, for example [SignUpSignIn.xml](https://github.com/Azure-Samples/active-directory-b2c-custom-policy-starterpack/blob/master/SocialAndLocalAccounts/SignUpOrSignin.xml), specifies the user journey which Azure AD B2C will execute. Find the **DefaultUserJourney** element within [relying party](https://docs.microsoft.com/en-us/azure/active-directory-b2c/relyingparty). Update the **ReferenceId** to match the user journey ID, in which you added the identity provider.

In the following example, for the CustomSignUpOrSignIn user journey, the **ReferenceId** is set to CustomSignUpOrSignIn:

XML

<RelyingParty>

<DefaultUserJourney ReferenceId="CustomSignUpSignIn" />

...

</RelyingParty>

**Upload the custom policy**

1. Sign in to the [Azure portal](https://portal.azure.com).
2. Select the **Directory + Subscription** icon in the portal toolbar, and then select the directory that contains your Azure AD B2C tenant.
3. In the Azure portal, search for and select **Azure AD B2C**.
4. Under **Policies**, select **Identity Experience Framework**.
5. Select **Upload Custom Policy**, and then upload the two policy files that you changed, in the following order: the extension policy, for example TrustFrameworkExtensions.xml, then the relying party policy, such as SignUpSignIn.xml.

**Test your custom policy**

1. Select your relying party policy, for example B2C\_1A\_signup\_signin.
2. For **Application**, select a web application that you [previously registered](https://docs.microsoft.com/en-us/azure/active-directory-b2c/tutorial-register-applications). The **Reply URL** should show https://jwt.ms.
3. Select the **Run now** button.
4. From the sign-up or sign-in page, select **Oracle** to sign in with Oracle account.

If the sign-in process is successful, your browser is redirected to https://jwt.ms, which displays the contents of the token returned by Azure AD B2C.