

## Azure AD B2C – Configuring Identity Providers

Azure Active Directory B2C provides business-to-customer identity as a service. Your customers use their preferred social, enterprise, or local account identities to get single sign-on access to your applications and APIs.

Azure AD B2C is a customer identity access management (CIAM) solution capable of supporting millions of users and billions of authentications per day. It takes care of the scaling and safety of the authentication platform, monitoring, and automatically handling threats like denial-of-service, password spray, or brute force attacks.

### Diagram:

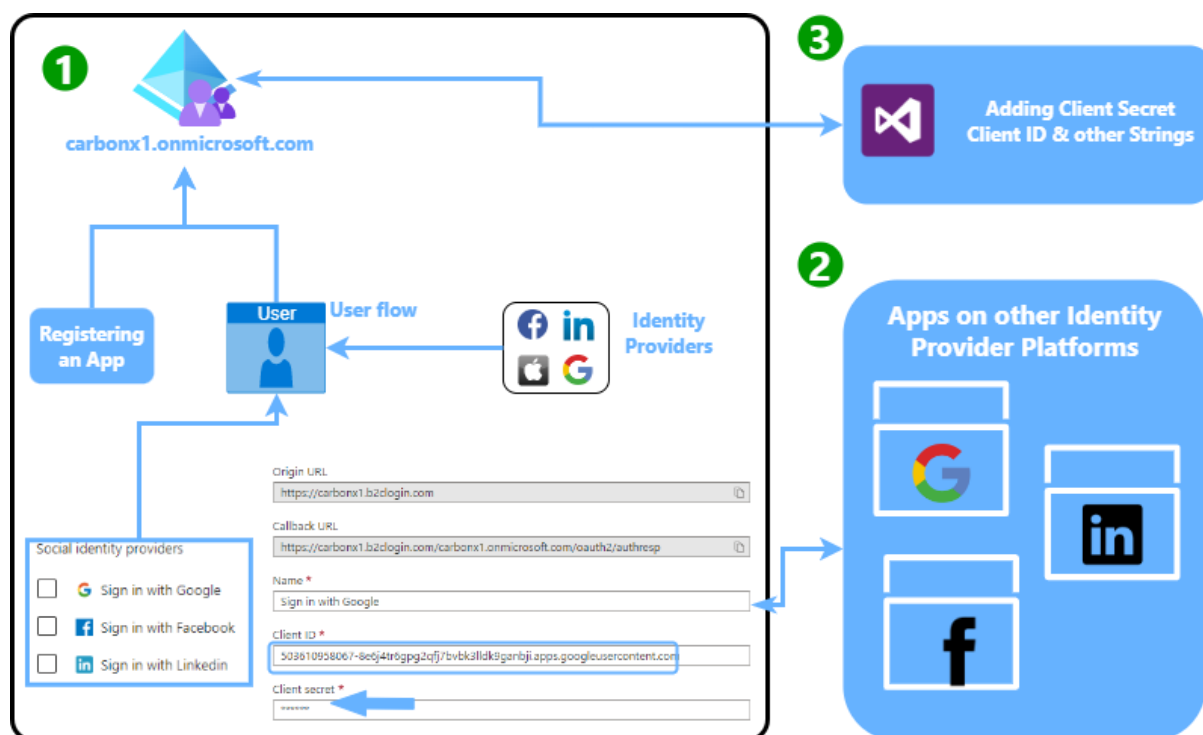
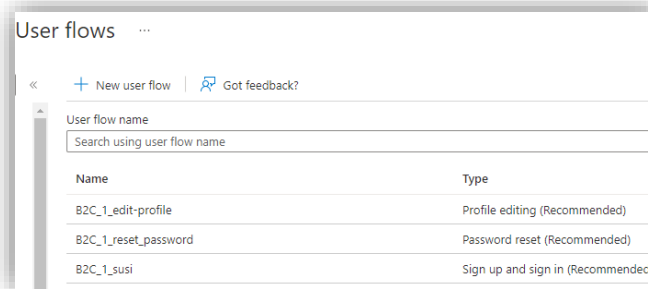
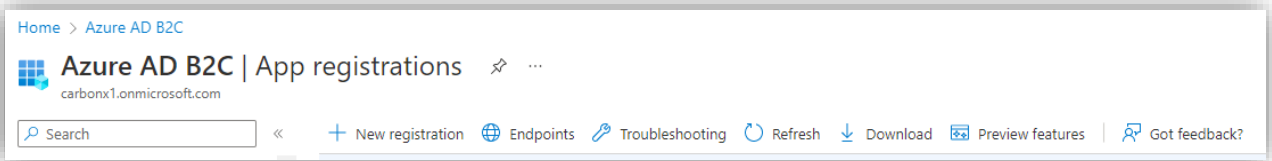


Figure 1

### Step1: Creating Azure AD B2C and doing Configuration

1. Create and Azure AD B2C
2. Add identity providers i.e Google, Facebook, LinkedIn etc
3. Adding Origin & Redirect Urls in to App of Respective Identity Provider
4. Adding client id and client secret of that App in to you B2C identity Providers
5. Configure Userflows in Policies section by selection userflow type and filling other details
  - a. In Social Identity Provider Section of Userflow select the provider you want to use
  - b. Select attributes and return claims can also add your custom attributes
6. Select Api and Expose the scopes by registering and APP

- a. In supported account type select “Account in any Identity provider or organizational directory”
7. Now use Expose and API blade to Expose the link
8. In Authentication blade select id tokens and access tokens
9. In userflow properties we can change properties according to our need i.e password complexity, MFA option etc
10. For Page layout we can select already available templates or can use our custom



*Creating Userflows 1*

## Step2: Configuring Connection in our WebApp i.e Dotnet Web Application

1. Go to source files in of WebApp in VS
2. In webclient folder of app go to appsettings.json file and add the followings information of Azure AD B2C in to that file
  - a. Instance
  - b. Domain
  - c. Client ID
  - d. Client Secret
  - e. Base-address
  - f. URL
  - g. ReadScope
  - h. Userflow Name
3. Make the same require change in appsetting.json file in Api folder

```
{
  "AzureAdB2C": {
    "Instance": "https://carbonx1.b2clogin.com", //Replace 'tenant-name' with Azure AD B2C tenant name
    "Domain": "carbonx1.onmicrosoft.com", //Replace 'tenant-name' with Azure AD B2C tenant name
    "ClientId": "a56db7cd-ad4c-4da1-a70a-fd24a8430a57", //Azure AD B2C Products.Api app registration Application (client) ID
    "Audience": "a56db7cd-ad4c-4da1-a70a-fd24a8430a57", //Same as Client Id
    "SignUpSignInPolicyId": "B2C_1_susi",
    "EditProfilePolicyId": "B2C_1_edit-profile",
    "ResetPasswordPolicyId": "B2C_1_reset_password"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

*WebApp API setting 1*

```
{
  "AzureAdB2C": {
    "Instance": "https://carbonx1.b2clogin.com", //Replace 'tenant-name' with Azure AD B2C tenant name
    "Domain": "carbonx1.onmicrosoft.com", //Replace 'tenant-name' with Azure AD B2C tenant name
    "ClientId": "a56db7cd-ad4c-4da1-a70a-fd24a8430a57", //Azure AD B2C Products.WebClient App registration client id
    "ClientSecret": "n5j8Q~RtpR4gcTgEDbsfTHSyJjV15K~12AjG6c_B", //Generate Client Secret for Azure AD B2C Product.WebClient
    "SignedOutCallbackPath": "/signout/B2C_1_susi",
    "CallbackPath": "/signin",
    "SignUpSignInPolicyId": "B2C_1_susi",
    "EditProfilePolicyId": "B2C_1_edit-profile",
    "ResetPasswordPolicyId": "B2C_1_reset_password",
    "RedirectUrl": "https://localhost:44337/signin"
  },
  "ProductApi": {
    "BaseAddress": "https://localhost:44352/",
    "Url": "https://localhost:44352/api/products/get",
    "ReadScope": "https://carbonx1.onmicrosoft.com/productsapi/products.view" //Replace 'tenant-name' with Azure AD B2C tenant
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  }
}
```

*Web client Settings 1*

### Step3: Creating and Configuring Apps on Third Party Identity Provider

1. Create and App on the provider (social platform you want to use) developer portal
2. Select Authentication option on that App
3. Add Origin URI and Redirect URI of Azure AD B2C in it
4. Add Client ID and Client Secret from App to Identity provider section in Azure AD B2C

Below are given example of App created on Google, Facebook and LinkedIn

#### 1. For Google Identity

**Name \***  
signin-demo

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

**The domains of the URIs you add below will be automatically added to your OAuth consent screen as [authorized domains](#).**

**Authorized JavaScript origins** ⓘ

For use with requests from a browser

**URIs 1 \***  
https://carbonx1.b2clogin.com

[+ ADD URI](#)

**Authorized redirect URIs** ⓘ

For use with requests from a web server

**URIs 1 \***  
https://carbonx1.b2clogin.com/carbonx1.onmicrosoft.com/oauth2/authresp

[+ ADD URI](#)

|                      |                                                                         |
|----------------------|-------------------------------------------------------------------------|
| <b>Client ID</b>     | 503610958067-8e6j4tr6pgg2qfj7bvb3lIdk9ganbji.apps.googleusercontent.com |
| <b>Client secret</b> | <del>g0PSPX-49MT7YkSn5Mmnn6H3vUPFndBdS</del>                            |
| <b>Creation date</b> | September 28, 2022 at 10:07:52 AM GMT+5                                 |

## 2. For Facebook Identity

|                  |                   |             |
|------------------|-------------------|-------------|
| <b>App ID</b>    | <b>App secret</b> | <b>Show</b> |
| 1332983383774701 | ••••••••          |             |


**Valid OAuth Redirect URIs**

A manually specified redirect\_uri used with Login on the web must exactly match one of the URIs listed here. This list is also used by the JavaScript SDK for in-app browsers that suppress popups. [?]

https://carbonx1.b2clogin.com/carbonx1.onmicrosoft.com/oauth2/authresp ✕

[Copy to clipboard](#)

### 3. For LinkedIn Identity


 **webapp demo**  
Client ID: 77i7uzxgbs5jny | Created: Sep 27, 2022

Settings **Auth** Products Analytics

**Application credentials**

Authentication keys


Client ID:  
77i7uzxgbs5jny

Client Secret:  
..... 

OAuth 2.0 settings

Token time to live duration

Access token: 2 months (5184000 seconds)

Authorized redirect URLs for your app 

https://carbonx1.b2clogin.com/carbonx1.onmicrosoft.com/oauth2/authresp

Reference:

GitHub Repository URL:

[Sharp-Programmer/SharpProg.Tutorials.AzureB2C \(github.com\)](https://github.com/Sharp-Programmer/SharpProg.Tutorials.AzureB2C)

GitHub Repository URL:

[Azure Active Directory B2C documentation | Microsoft Learn](#)