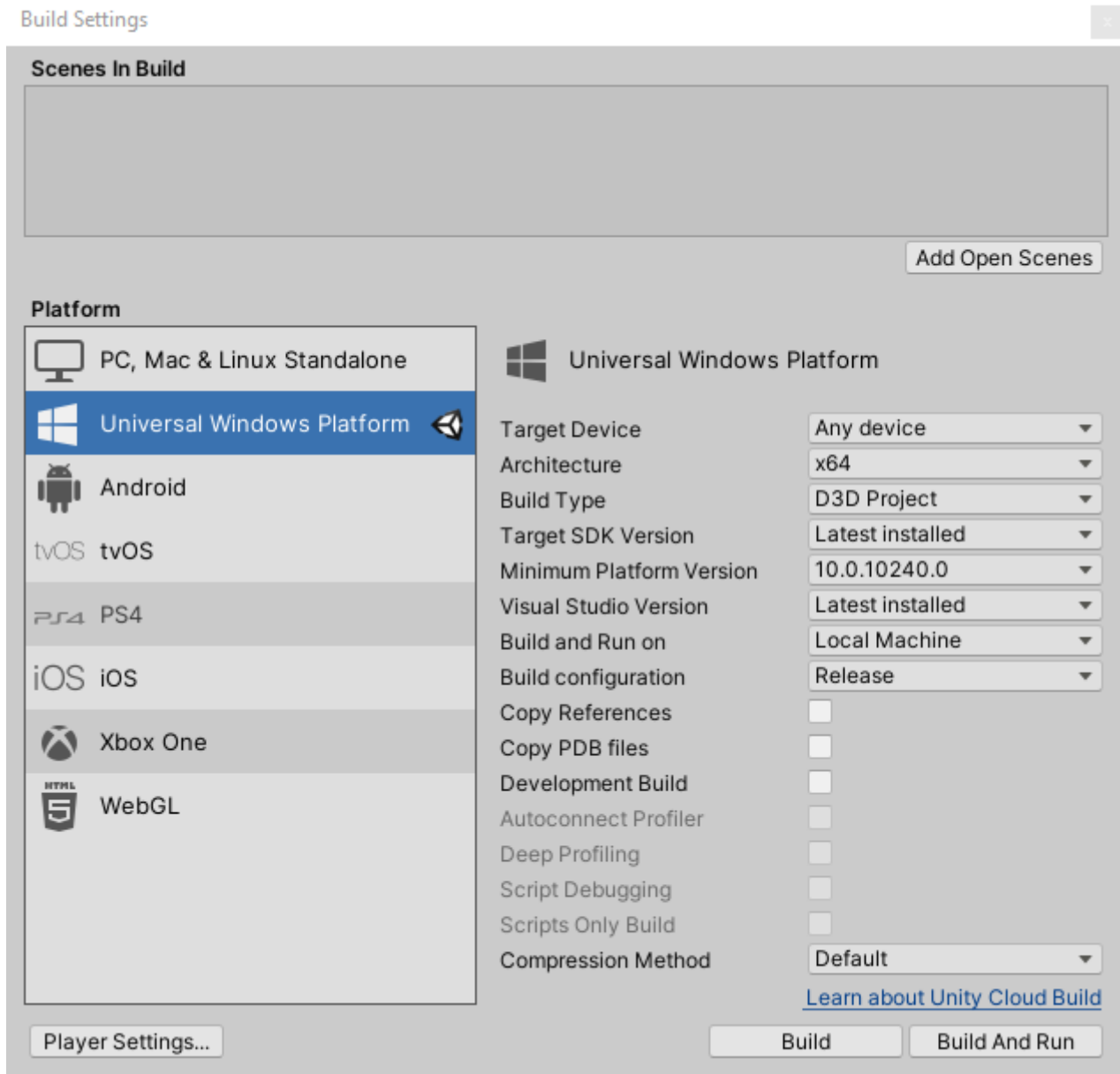synergiz

lighthouse

LightHouse
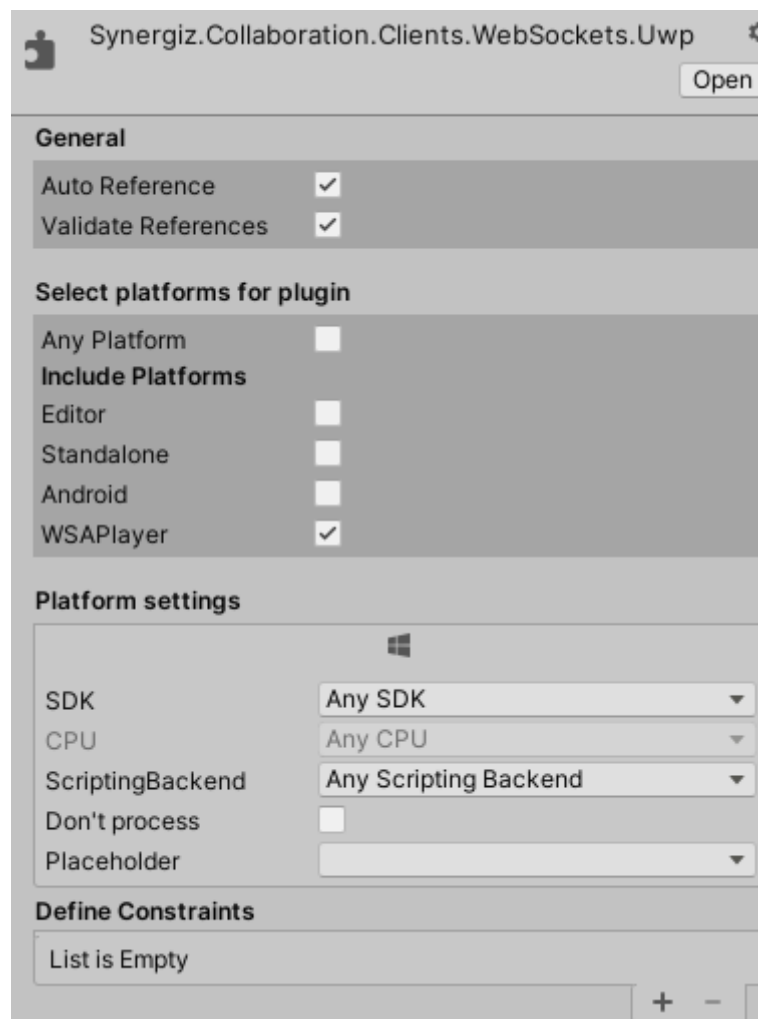
QUICK-START DOCUMENTATION

1. CREATE A NEW UWP UNITY PROJECT

2. ADD « CLIENT/LIBS/UNITY » FILES INTO THE PROJECT

a. UWP dll should have « WSAPlayer » as only target

b. Create a script named « CollaborationManager » :

- Import this :

```csharp
using Synergiz.Collaboration.Clients;
using System.Threading.Tasks;

#if UNITY_EDITOR
using Synergiz.Collaboration.Clients.WebSockets;
#else
using Synergiz.Collaboration.Clients.WebSockets.Uwp;
#endif
```

- Create two client properties : Server address and name

```csharp
public string HostAddress = "ws://localhost:5000/Collaboration";
public string clientName = "SynergizBoy";

private CollaborationClient client;
```

- Create an asynchronous function for the client connection :

```csharp
private async Task<bool> ConnectionAsync(string address, string name)
{
    try
    {
        if(client != null)
        {
            await client.DisconnectAsync();
        }

        client = new CollaborationClient(new WebSocketsConnectionClient());

        client.PropertyChanged += Client_PropertyChanged;

        var connectionResult = await client.ConnectAsync(
            new System.Uri(address),
            name);

        if (connectionResult.IsConnected)
        {
            Debug.Log($"Hello {this.clientName}, you are connected !");

            await InitProperties();
        }
        else
        {
            throw new System.Exception();
        }

        return true;
    }
    catch
    {
        return false;
    }
}
```

- Create a function to initialize the shared properties, if they already exist online it can be a good idea to retrieve them.

Example :

```
private string textHelloWorld = "Hello World";
```

```csharp
private async Task<bool> InitProperties()
{
    var textResult = await client.CreatePropertyAsync<string>("TextCollab", textHelloWorld,
        Synergiz.Collaboration.Shared.NotificationTypes.Others,
        Synergiz.Collaboration.Shared.PropertyAccess.ReadWrite);

    if(textResult.HasError)
    {
        Debug.LogError("IsTransformMode creation issue");
        return false;
    }

    if(textResult.ServerValue == Synergiz.Collaboration.Shared.CreatePropertyResult.PropertyAlreadyExists)
    {
        var getResult = await client.GetStringPropertyAsync("TextCollab");
        if(getResult.HasError)
        {
            return false;
        }

        textHelloWorld = getResult.ServerValue;
    }

    return true;
}
```

- Once the property is shared, handle the event trigger when a property is updated on the server.

Example :

```csharp
object locker = new object();
string textHelloWorldValue = null;

1 reference
private void Client_PropertyChanged(string hostName, string typeName, string propertyName, object value)
{
    switch(propertyName)
    {
        case "TextCollab":

            lock (locker)
            {
                textHelloWorld = (string)value;
            }

            break;
    }
}
```

- It is important to understand that when you get a shared property value, it is not during an application cycle (like Unity for this situation). This is why we need to pass through the main thread Update.

Example :

```
Unity Message | 0 references
void Update()
{
    if(textHelloWorldValue != null)
    {
        lock(locker)
        {
            textHelloWorld = textHelloWorldValue;
        }
        textHelloWorldValue = null;
    }
}
```