# Contributing

## Dependencies

\* [Visual Studio 2019](https://visualstudio.microsoft.com/)

\* [Unity Editor](https://unity3d.com/unity/editor) (optional)

\* .NET Core SDK and runtimes (run `init` to install)

This repo uses the .NET Core SDK and runtimes to build and test the projects.

You can install the right versions of the SDK and runtimes by running the `init.ps1` script at the root of the repo.

By default no elevation is required as these toolsets are installed in a per-user directory. Launching `devenv` from the same PowerShell window that you ran the script will lead VS to discover these per-user toolsets.

To get VS to find the toolsets when launched from the Start Menu, run `init -InstallLocality machine`, which requires elevation for each SDK or runtime installed.

## How to Build

Open `MessagePack.sln` on Visual Studio 2019.

Alternatively you may build from the command line using `msbuild.exe` or:

dotnet build /p:platform=NoVSIX

## Unity

Unity Project requires several dependency DLL's. At first, run `copy\_assets.bat` under `src\MessagePack.UnityClient`.

Then open that directory in the Unity Editor.

## Where to find our CI feed

Once a change is in a shipping branch (e.g. `v1.8`, `v2.0`, `master`), our CI will build it and push the built package

to [our CI feed](https://dev.azure.com/ils0086/MessagePack-CSharp/\_packaging?\_a=feed&feed=MessagePack-CI). To depend on

one of the packages that are on our CI feed (but not yet on nuget.org) you can add this to your nuget.config file:

```xml

<add key="MessagePack-CI" value="https://pkgs.dev.azure.com/ils0086/MessagePack-CSharp/\_packaging/MessagePack-CI/nuget/v3/index.json" />

```