Client Modding Quick Start Guide

In order to write client mods for SPT (or any other Unity game with BepInEx) you will need to know how to program in C#. See the resources section to get started if you are new to programming.

Resources:

- C# Learning resource: https://dotnet.microsoft.com/en-us/learn/csharp
- BepInEx docs: https://docs.bepinex.dev/
- Harmony 2 docs: https://harmony.pardeike.net/articles/intro.html
- Client mod examples repo: https://github.com/Jehree/SPTClientModExamples

Step 0, Installing needed programs:

1. Install Visual Studio: https://visualstudio.microsoft.com/
    - Click **Download Visual Studio**.
    - Once the installer is downloaded, run it. Click **Available** at the top, then click **Install** under **Visual Studio Community 2022**.
    - Scroll down under the **Workloads** tab until you see **Game development with Unity**. Check the box next to that workload, and click **Install** in the bottom right.
2. Install .NET 8.0 SDK: https://dotnet.microsoft.com/en-us/download/dotnet/8.0
3. Install 4.7.1 .NET Framework (runtime version): https://dotnet.microsoft.com/e…d/dotnet-framework/net471
4. Install dnSpy: https://github.com/dnSpyEx/dnSpy/releases
    - Choose the latest stable release (should be at the top of the page linked above)
    - Scroll to the bottom of that release to the **Assets** section, and select either **dn-spy-net-win32.zip** or **dn-spy-net-win64.zip** depending on your system.
    - Create a folder on your C:Drive called dnSpy, then drag the contents of the zip you downloaded into that folder.
    - Optionally, right click **dnSpy.exe** and create a shortcut, then place it on your desktop.

Step 1, SPT development install setup:

1. Create a fresh SPT install to use for development.

    - SPT installer page: SPT-INSTALLER
2. Create a Development folder to hold your mod projects in the root directory of your new SPT install (Ex: [your development SPT install]/Development).
    - Doing this is nice because when it is time to update your mod to a new SPT version, you can just paste the whole Development folder into that install and get to work without needing to update reference paths, etc. (thank Drakia for the idea!)
3. Install Unity Explorer, head to this link: https://github.com/sinai-dev/UnityExplorer and download the **Mono** version of **BIE 5.X**. Install it like any other client mod.
4. Navigate to **[your development SPT install]/BepInEx/config** and open **BepInEx.cfg**, set **LogChannels = all** and **Enabled = true**. This will cause the BepInEx console to launch when you launch SPT. All logging done in your mod will appear in this console.
5. Make sure to run your dev install once, all the way to the main menu and then quit. This deobfuscates the assembly.

Step 2, Export Assembly-CSharp.dll file to view Tarkov's decompiled source:

We do this so we can study the Tarkov source code to see what we may want to change, as well as where to change it. I suggest creating a Visual Studio project to contain the decompiled source. Fortunately, dnSpy has the ability to do this for us!

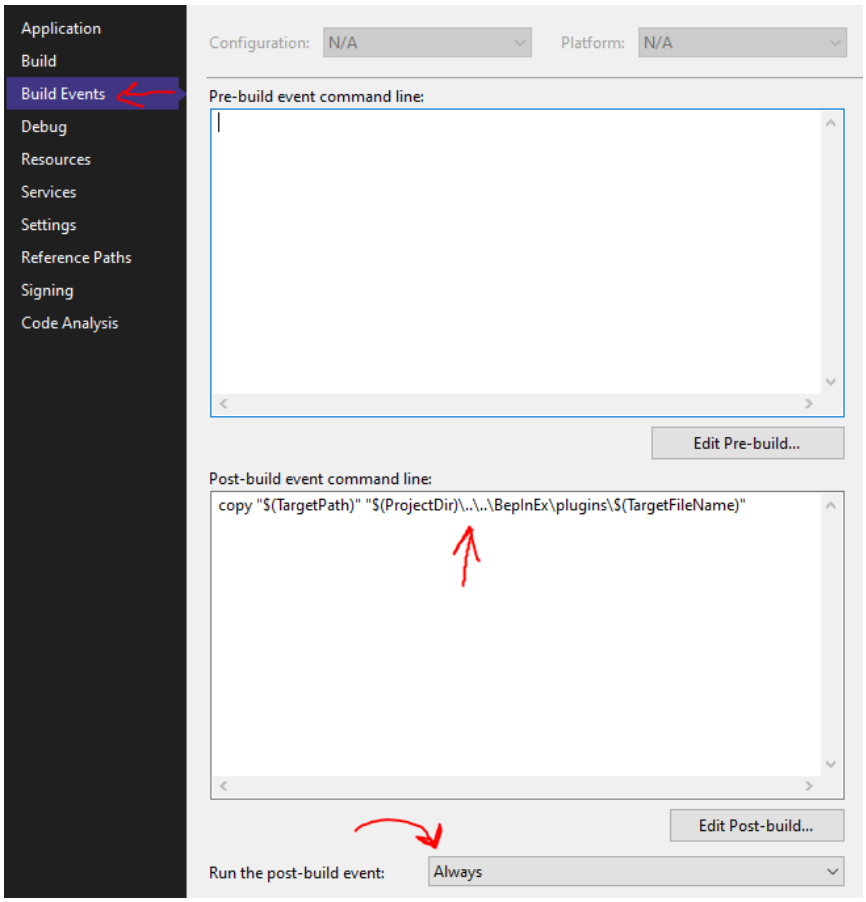
1. Create a folder to store your decompiled source. I suggest naming it after the SPT version the source is for, to differentiate it from future assemblies you decompile when SPT updates to a new EFT client version. (Ex: SPT380_assembly)
2. Open dnSpy with the shortcut we created earlier.
3. Go to **File > Open** and navigate to this path: ***[your development SPT install]/EscapeFromTarkov_Data/Managed***, select the **Assembly-CSharp.dll** file and click **Open**.
4. Once the Assembly-CSharp.dll file is open, you should see it in dnSpy. You can look through the code inside dnSpy if you prefer, but I suggest creating a Visual Studio project to hold it. You can do so by going to **File > Export to Project…** then selecting the folder we created in step 1.
5. Open the project! Run **Assembly-CSharp.sln** in the project folder dnSpy created to do so.

You can use **CTRL + Shift + F** to search the entire solution for code. Go ahead and try and search for something like "Jump" or "Door" to see what you can dig up!

Step 3, Visual Studio mod project setup:

note: If you would prefer to copy the **SPTClientModExamples** project and rename it instead of making a new one from scratch, skip step 3 and 3.1 and follow the steps in the README in the repo instead: https://github.com/Jehree/SPTClientModExamples

1. Open Visual Studio, head to **File > New > Project**.
2. Select **Class Library**. Make sure it is the one for **C# development**.
3. Set the **Project name** to the name of your mod (this can be a little annoying to change later, FYI).
4. Set the **Location** to the path to the Development folder you just created.
5. Check the **Place solution and project in same directory** box.
6. Set the **Framework** to **.NET Standard 2.1**, click **Create** in the bottom right.
7. Rename the single **.cs** file in your project to **Plugin.cs**, and create a **Patches** folder.
8. Double click **Properties** on the left in the **Solution Explorer** area, and click on **Build Events** .
9. Paste the command below into the **Post-build event command line:** box, and set **Run the post build event:** to **Always**.
    - Command: *copy "$(TargetPath)" "$(ProjectDir)\..\..\BepInEx\plugins\$(TargetFileName)"*
    - Why do this?: This will automatically copy your mod's .dll file into BepInEx/plugins when you build it with **Build > Build Solution**, so you don't have to manually place it there.
    - NOTE: If you build your mod without making any changes to the code, the copy will not happen. If you want to force it to copy, manually delete your mod's .dll file from *BepInEx/plugins*, make a small change to your code like adding a comment, then build it.



Step 3.1, Adding references to .csproj file:

We need to add some references so that classes in the EFT source code as well as some helper SPT classes can be referenced in your mod. Drakia has a convenient list of starting references, so we are going to add those. Do note that these are not *all* the refs that you can use. You can reference any .dll file in *EscapeFromTarkov_Data/Managed*, so feel free to add more if needed!

1. Make sure your mod project is closed.
2. Navigate to ***[your development SPT install]/Development/[your mod name]***, and open the **.csproj** file with a text editor (Notepad is fine).
3. Scroll to the bottom of the file.

*Replace the **</ItemGroup>** tag and contents as shown below in the screenshot with the one linked below*

1. Drakia's refs: https://github.com/Jehree/SPTC…/main/StarterRefPaste.txt

Step 4, Start coding!

1. Head to this repo https://github.com/Jehree/SPTClientModExamples and use the examples there to get your hands dirty!
2. Once you're ready to test, go to **Build > Build Solution** or press **F6** to build your mod. It should be automatically copied into *BepInEx/plugins*, so all you should have to do is build and launch the game to test.
3. Have fun!

Credit to the cool peeps:

To those who have helped me learn all this shiz and gave me feedback on this doc, you guys ROCK. Thank you so much!

- DrakiaXYZ
- Cj
- Arys
- Kiki
- mpstark
- Tyfon

Additional information

Programming skills required	1
Technical knowledge requirement	Intermediate

Prerequisites

Tutorial prerequisite list	Being able to program in C# is not necessary to follow this guide, but it *will* be necessary to write an actual client mod. I suggest getting a hang of the basics down before trying to make a client mod. This guide assumes you know how to install SPT as well as SPT mods.