

Tutorial: How to debug the game client with dnSpy

Prerequisites

- dnSpy ([download](#))
- SPT 3.11.0 or newer (guide was tested and verified with SPT 3.11.2)
- At least 2 monitors highly recommended (see the **Notes and Tips** section 3 for an explanation)

Chapter 1 - Preparing the client

1. Download this prepared archive - [download](#) (if you do not trust the download, then see **Chapter 4** for how to prepare your own)
2. Backup these game files:
 1. \BepInEx\config\BepInEx.cfg
 2. \EscapeFromTarkov_Data\boot.config
 3. UnityPlayer.dll
 4. WinPixEventRuntime.dll (if it exists)
3. Overwrite all game files with the ones from the previously downloaded archive
4. Make sure the \EscapeFromTarkov_Data\boot.config file is set to Read-Only. Otherwise, the changes to it might get overwritten on game start
5. Go to the \user\launcher\config.json file and open it in a text editor ([Notepad++](#) recommended)
6. Add "WinPixEventRuntime.dll" to the string array setting "ExcludeFromCleanup". Ex.:
"ExcludeFromCleanup": ["WinPixEventRuntime.dll"],
7. Start the SPT Server & Launcher. If you've done everything correctly, the game will launch without issues and the text "**Development Build**" will be visible in the bottom-right of the screen
8. Start dnSpy. Make sure your Assembly Explorer is clear (optional, but highly recommended - see **Notes and Tips** section 5)
9. In dnSpy, click Debug in the top bar, and then Attach to Process (Unity)... Then, select the EscapeFromTarkov.exe process from the list
10. After the game process has been attached, open the assembly you want to debug in one of two ways:
 1. (Recommended) Open the loaded module view Debug -> Windows -> Modules -OR- Ctrl+Alt+U and search for the assembly you need
 2. Open the assembly file you want to debug File -> Open... -OR- Ctrl+O (see **Notes and Tips** section 6)
11. That's it! Keep reading for some notes on debugging oddities and how to work around them

Chapter 2 - Common Issues

1. If you're getting an error about a missing "WinPixEventRuntime.dll" when starting the game, please make sure you properly set the SPT Launcher's config to not delete the necessary DLL file on game start.
2. If the game is not appearing in the Unity process list in dnSpy, please double-check and make sure that your "boot.config" file has the Read-Only flag set. If it doesn't, then the game will have overwritten it's contents and you'll need to replace the file from the downloaded archive again.
3. If the game fails to load properly, such as by getting stuck in the loading screen - please check the BepInEx logs and see if any of the SPT plugins failed to load. The Unity development build has some extra checks in place that might produce errors which are not present when running the regular version of EFT. If such a scenario occurs, then please contact me and let me know about the issue!

4. If the game crashes with strange errors when attempting to start it, then it's possible that the prepared archive's DLLs are outdated and EFT has had an engine update since this guide was written. Please go to **Chapter 4** and try to create your own Development Build files.

Chapter 3 - Notes and Tips

1. With the development build enabled, if any client-side errors occur, a Unity debugging console will appear. These errors can usually be ignored, and the console itself can be closed safely. Some client mods might generate a constant stream of errors, preventing the console from being dismissed - in that case I recommend temporarily removing such mods.
2. Currently, Mono debugging might have an issue with the "Step-Over" function, which can cause it to behave in unexpected ways. I would recommend avoiding it and instead adding breakpoints to upcoming code lines that you want to inspect, instead of stepping over.
3. When a breakpoint is hit, all game threads are frozen, including the renderer. This causes the game window to freeze and will prevent minimizing the game in any way. This is easiest to mitigate by having more than one monitor, so you can keep the dnSpy window on a monitor separate from the game to avoid issues.
4. Game performance and load times are expected to be slower than usual, due to debugging builds being more performance-intensive. I would not recommend for anyone to play the game casually with the debugging build active.
5. If an assembly was added to the Assembly Explorer in dnSpy before the game process was attached, no debugging symbols may be loaded and breakpoints may not work. That's why I generally recommend clearing the workspace before starting.
6. Any DLLs that are patched with preloader patches in BepInEx will be dumped to `\BepInEx\DumpedAssemblies\EscapeFromTarkov` when using the recommended BepInEx config (included in the prepared archive). Keep this in mind if you're trying to manually add assemblies into the workspace.
7. To return the game to a regular, non-development build, simply restore the files you had backed up previously.

Chapter 4 - Preparing your own development build files

1. Go to the [Unity Download Archive](#) and download the Unity Editor version that matches EFT's current engine version. You can find out which exact version is needed by checking the game's `UnityPlayer.dll` file version in the Details tab. **You do not need any additional build support installers.** Alternatively, you can use the Unity Hub to download the correct editor version
2. Go the directory where you installed the Unity Editor and navigate to this folder: `\Editor\Data\PlaybackEngines\windowsstandalonesupport\Variations\win64_development_mon`
3. Copy `UnityPlayer.dll` and `WinPixEventRuntime.dll` over to your game directory. Please backup any files before overwriting!
4. Open the `\EscapeFromTarkov_Data\boot.config` file and add the following line: `player-connection-debug=1`. **Make sure to set the file to Read-Only mode after editing it!**
5. Open the `\BepInEx\config\BepInEx.cfg` file and apply the following changes:
 1. Change `HarmonyBackend` value to `cecil`
 2. Change `DumpAssemblies` value to `true`
 3. Change `LoadDumpedAssemblies` value to `true`

Sources

Initial guide on how to convert EFT into a debug build and debug it: [dnspy Wiki on GitHub](#)

Additional information for loading game assemblies in dnSpy with BepInEx: [BepInEx documentation](#)

Archive for Unity Editor versions, from which the prepared archive of 2022.3.43f1debugging DLLs was created: [Unity Download Archive](#)

Additional information	
Programming skills required	1
Technical knowledge requirement	High

Prerequisites	
Tutorial prerequisite list	dnSpy (download) SPT 3.0.0 or newer