

Hi there, and thank you for using the Kash Shop for your modification needs!

This documentation exists to help you through installing a patch on an already modded copy of Intersect. Let's go over the problem, and then we'll show you how to fix it!

So the problem is actually pretty simple: If you are trying to install a patch that edits the same code file as a patch that you might have already installed, you're going to run into some issues. This is because a **git patch** applies it's changes by looking for precise line numbers, and changing what needs to be changed based off of that. So if we take a patch that adds a new feature to Items, install it, and then try to install another feature that *also* adds a new feature to Items, that patch will look for a line number that has been moved from your last patch.

In short, it will fail, and unless there's a change in policy regarding Source Tutorials on AGD, it will stay like that. That's where this documentation comes along!

Here's a quick checklist for those returning to these docs:

- 1) Make a backup
- 2) Apply patch with the --reject command
- 3) Check where hunks were rejected
- 4) Open the .rej files
- 5) Open the files that need editing via either a text editor or Visual Studio
- 6) Save your edits
- 7) Build/Test
- 8) Commit

We're now going to walk through how to apply 2 patches, and how to fix any errors that crop up. So to start, we're going to be installing Mobile Npcs, link of which can be found here:

<https://www.ascensiongamedev.com/topic/5074-5-kash-shop-mobile-npcs/>

And then we're going to be installing the Map Types Editor on top of that, link of which can be found here:

<https://www.ascensiongamedev.com/topic/5075-15-kash-shop-map-type-editor/>

Let's count the steps together, shall we?

1.Create a backup!

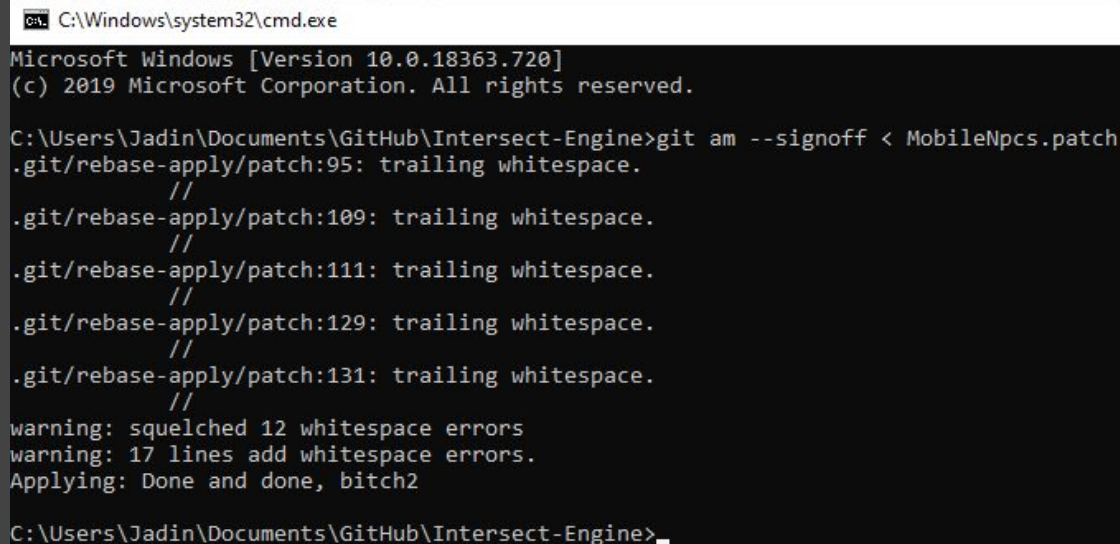
This is super crucial! Always create a backup when messing around with patches or code in general!

2.Install the first patch.

If you don't know how to do this yet, there's some documentation found on AGD. Here's the link for that:

<https://docs.freemmorpgmaker.com/dev/sharing/applying.html>

After you've finished, your command prompt should read something like this:



```
cmd. C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Jadin\Documents\GitHub\Intersect-Engine>git am --signoff < MobileNpcs.patch
.git/rebase-apply/patch:95: trailing whitespace.
//
.git/rebase-apply/patch:109: trailing whitespace.
//
.git/rebase-apply/patch:111: trailing whitespace.
//
.git/rebase-apply/patch:129: trailing whitespace.
//
.git/rebase-apply/patch:131: trailing whitespace.
//
warning: squelched 12 whitespace errors
warning: 17 lines add whitespace errors.
Applying: Done and done, bitch2

C:\Users\Jadin\Documents\GitHub\Intersect-Engine>
```

You'll also want to **Commit** at this stage.

3. Install the next patch.

This is where you need to change it up a bit. Instead of the usual `--signoff` command, we're going to type this in instead:

```
C:\Users\Jadin\Documents\GitHub\Intersect-Engine>git am --reject < MapTypesEditor.patch
```

The “`--reject`” command will allow us to install everything in the patch that *does* work so far. When you hit enter, you'll get a large wall of text. This particular patch is pretty big, so we got a large wall of text to look at. You're going to end up with something like this:

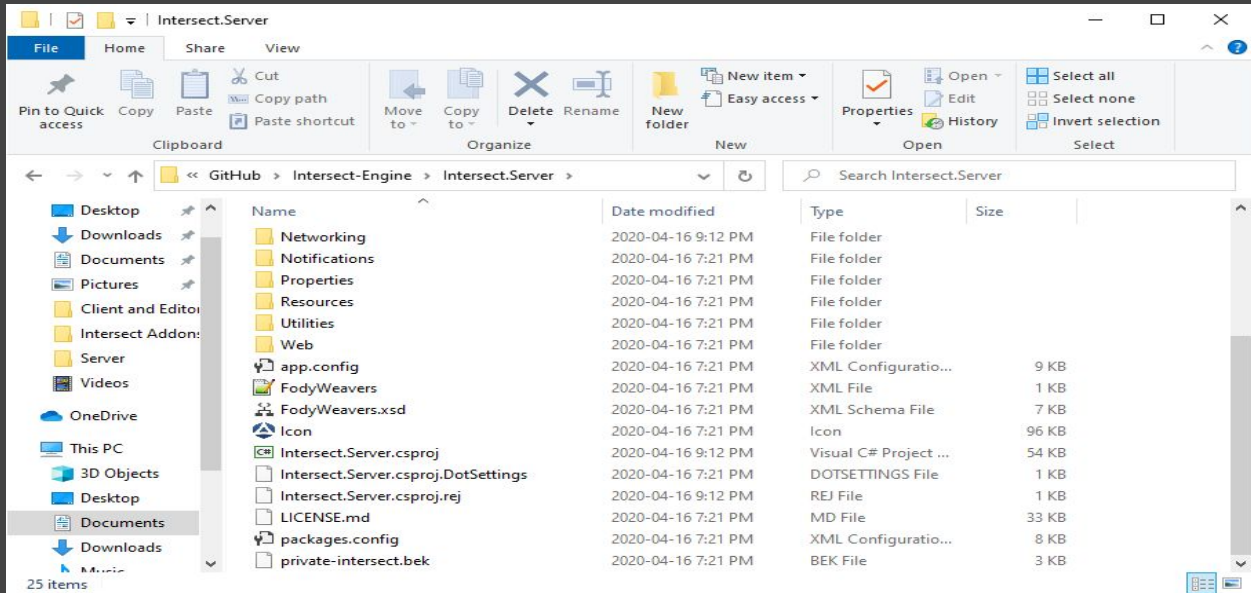
```
Applied patch Intersect.Editor/Forms/frmMain.resx cleanly.
Applied patch Intersect.Editor/Intersect.Editor.csproj cleanly.
Applied patch Intersect.Editor/Localization/Strings.cs cleanly.
Applied patch Intersect.Editor/Maps/MapProperties.cs cleanly.
Applied patch Intersect.Editor/Networking/PacketHandler.cs cleanly.
Applied patch Intersect.Server/Database/DbInterface.cs cleanly.
Applied patch Intersect.Server/Database/GameData/GameContext.cs cleanly.
Applied patch Intersect.Server/Entities/Entity.cs cleanly.
Applied patch Intersect.Server/Entities/Player.cs cleanly.
Applying patch Intersect.Server/Intersect.Server.csproj with 1 reject...
Rejected hunk #1.
Applied patch Intersect.Server/Migrations/Game/20200414162817_AddingMapTypesEditor.Designer.cs
Applied patch Intersect.Server/Migrations/Game/20200414162817_AddingMapTypesEditor.cs cleanly.
Applied patch Intersect.Server/Migrations/Game/20200415005435_AddingMoreMapTypeVars.Designer.c
Applied patch Intersect.Server/Migrations/Game/20200415005435_AddingMoreMapTypeVars.cs cleanly
Applied patch Intersect.Server/Migrations/Game/GameContextModelSnapshot.cs cleanly.
Applied patch Intersect.Server/Networking/PacketHandler.cs cleanly.
Applied patch Intersect.Server/Networking/PacketSender.cs cleanly.
hint: Use 'git am --show-current-patch=diff' to see the failed patch
Applying: Done and done, bitch
Patch failed at 0001 Done and done, bitch
When you have resolved this problem, run "git am --continue".
If you prefer to skip this patch, run "git am --skip" instead.
To restore the original branch and stop patching, run "git am --abort".

C:\Users\Jadin\Documents\GitHub\Intersect-Engine>
```

Now, you'll leave this command prompt open. What you want to look for is any “Rejected hunk”s. As you can see in this segment of text, we see that most of our patch has been applied “cleanly”. We *do* see, however, that there was a “Rejected hunk”. (This also shows you what file has rejected the code hunk!)

```
Applying patch Intersect.Server/Intersect.Server.csproj with 1 reject...
Rejected hunk #1.
```

This means that this is the part of the patch that failed. What you'll now do is navigate in your folder to "Intersect.Server", like this:



Now as you can see, we have a .rej file here:



We're going to want to open that file in a text editor. Personally I prefer Notepad++, which is a free software.



Now that it's open, you can see that this is the code that's causing the problem. I'll explain a bit about why this is failing, and then give you two options on how to fix it.

As you can see from the picture above, this is a chunk of code that causes a failure. In particular, take a look at this line at the top:

```
2 @@ -480,6 +480,14 @@
```

This is saying that on line 480 of Intersect.Server.csproj is where the change is supposed to happen. However, it can't go through with the change, because when it looks for line 480 in your file, it's also looking for the vanilla code to stick your new code between. That's what this segment is:

```
3 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
4 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
5 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
6 + <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
7 + <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
8 +   <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
9 + </Compile>
10 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
11 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
12 +   <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
13 + </Compile>
14 <Compile Include="Networking\Client.cs" />
15 <Compile Include="Entities\NPC.cs" />
16 <Compile Include="Networking\Helpers\NetDebug.cs" />
```

It's saying: At line 480, find these 3 lines of code:

```
<Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
<Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
<Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
```

And add your new code underneath that. Then make sure that your new code is followed by these 3 lines of code:

```
<Compile Include="Networking\Client.cs" />
<Compile Include="Entities\NPC.cs" />
<Compile Include="Networking\Helpers\NetDebug.cs" />
```

But if you open up your Intersect.Server.csproj file, you can see the problem:

```

473 <Compile Include="Maps\MapItemInstance.cs" />
474 <Compile Include="Maps\MapItemSpawn.cs" />
475 <Compile Include="Maps\MapNpcSpawn.cs" />
476 <Compile Include="Maps\MapResourceSpawn.cs" />
477 <Compile Include="Maps\MapTrapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
483 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
484 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs" />
485 <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
486 </Compile>
487 <Compile Include="Networking\Client.cs" />
488 <Compile Include="Entities\Npc.cs" />
489 <Compile Include="Networking\Helpers\NetDebug.cs" />
490 <Compile Include="Networking\PacketHandler.cs" />
491 <Compile Include="General\Globals.cs" />
492 <Compile Include="Networking\PacketSender.cs" />

```

There's more code there now, from our last patch! So when this current patch is looking for line 480, it finds it just fine. But when it's looking for what lines of code go beneath your new lines, it finds code that it doesn't know is there, causing the patch to fail!

I'll be going over 2 methods of fixing this. You can choose whatever solution works best for you!

4. A) Manual Editing

So you want to just put those few lines of code in yourself? This is the step for you! First thing's first, let's quit out of our patch installation. To do this, simply type in:

```
C:\Users\Jadin\Documents\GitHub\Intersect-Engine>git am --abort
```

And hit enter.

Next, we're going to open up our file that the hunk failed on. In this case, it's Intersect.Server.csproj. You can open this file in either a text editor or Visual Studio. If you're opening a csproj file in Visual Studio you'll have to open it by following these steps:

- 1) Right Click on the project in the solution explorer.
- 2) Click "Unload Project"
- 3) Right Click on the project again and click "edit"

- 4) Make your modifications
- 5) Save the file
- 6) Right Click the project file again and click “rebuild project”

Now we’re going to copy our lines that have the “+” by them in our .rej file, like so:

```

6 + <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
7 + <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
8 + <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
9 + </Compile>
10 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
11 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
12 + <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
13 + </Compile>

```

Now in your open code file, in this case Intersect.Server.csproj, we’re going to find the lines that the patch is looking for. In this case, it was looking for line 480, so we’re going to scroll down until we find line 480:

```

473 <Compile Include="Maps\MapItemInstance.cs" />
474 <Compile Include="Maps\MapItemSpawn.cs" />
475 <Compile Include="Maps\MapNpcSpawn.cs" />
476 <Compile Include="Maps\MapResourceSpawn.cs" />
477 <Compile Include="Maps\MapTrapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
483 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
484 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
485 <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
486 </Compile>
487 <Compile Include="Networking\Client.cs" />
488 <Compile Include="Entities\Npc.cs" />
489 <Compile Include="Networking\Helpers\NetDebug.cs" />
490 <Compile Include="Networking\PacketHandler.cs" />
491 <Compile Include="General\Globals.cs" />
492 <Compile Include="Networking\PacketSender.cs" />

```

We’re now going to paste our copied code into this section. Our patch wanted to put the new code starting at line 483, so we’ll make space and do just that:

```

476 <Compile Include="Maps\MapResourceSpawn.cs" />
477 <Compile Include="Maps\MapTrapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
483 <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
484 + <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
485 + <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
486 + </Compile>
487 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
488 + <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
489 + <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
490 + </Compile>
491 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
492 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
493 <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
494 </Compile>
495 <Compile Include="Networking\Client.cs" />
496 <Compile Include="Entities\Npc.cs" />

```

Now we're going to clean up by deleting those "+" signs, and making sure it's all indented in a uniform manner like the rest of this code:

```
477 <Compile Include="Maps\MapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
483 <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
484 <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
485 | <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
486 |</Compile>
487 <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
488 <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
489 | <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
490 |</Compile>
491 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
492 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
493 | <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
494 |</Compile>
495 <Compile Include="Networking\Client.cs" />
496 <Compile Include="Entities\Npc.cs" />
497 <Compile Include="Networking\Network\NetDebug.cs" />
```

Then save the file! Do this for each "rejected hunk" that was shown in your Command Prompt when you tried applying the second patch.

If all went well, your patch should now be in and working! If you're still having problems that you can't solve, drop by the Kash Shop Support discord and let us know! We'll get that working for you!

Kash Shop Support Discord Server:

<https://discord.gg/VHJ7vnW>

4. B) Continue Patch.

This method is a bit trickier, but if you like automation, it might be for you!

First thing's first, let's quit out of our patch installation. To do this, simply type in:

```
C:\Users\Jadin\Documents\GitHub\Intersect-Engine>git am --abort
```

And hit enter.

Now go open your rejected hunk (or as many as you want, if there's more than 1), like so:


```

1 diff a/Intersect.Server/Intersect.Server.csproj b/Intersect.Server/Intersect.Server.csproj (rejected hunks)
2 @@ -480,6 +480,14 @@
3     <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
4     <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
5     <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
6 +   <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
7 +   <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
8 +     <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
9 +   </Compile>
10 +   <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
11 +   <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
12 +     <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
13 +   </Compile>
14     <Compile Include="Networking\Client.cs" />
15     <Compile Include="Entities\NPC.cs" />
16     <Compile Include="Networking\Helpers\NetDebug.cs" />
17

```

Next we're going to add some lines, and fix what causes it to fail. The added lines will be added to line 2, and are formatted like this:

```

--- a/Intersect.Server/Intersect.Server.csproj
+++ b/Intersect.Server/Intersect.Server.csproj

```

And now we're going to fix the hunk. This is done by changing what the .rej file is looking for, but can't find. In this case, it's looking for these lines, all in order:

```

<Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
<Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
<Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
<Compile Include="Networking\Client.cs" />
<Compile Include="Entities\NPC.cs" />
<Compile Include="Networking\Helpers\NetDebug.cs" />

```

As we discussed earlier, however, with our last patch, its order now is like this:

```

<Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
<Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
<Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
<Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
<Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
  <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
</Compile>
<Compile Include="Networking\Client.cs" />
<Compile Include="Entities\NPC.cs" />
<Compile Include="Networking\Helpers\NetDebug.cs" />

```

Git looks for 3 lines before, then 3 lines after where your new code is going. So, as the 3 lines of code that it's looking for at the start are still there (the code that starts on line 480) we're going to edit the code it looks for at the bottom of your new code. This sounds complicated, but it's actually very simple.

Firstly, open up the file that your rejected hunk had a problem with, in this case: Intersect.Server.csproj. You're then going to scroll to the disputed area's line number. In this case, line 480:

```
473 <Compile Include="Maps\MapItemInstance.cs" />
474 <Compile Include="Maps\MapItemSpawn.cs" />
475 <Compile Include="Maps\MapNPCSpawn.cs" />
476 <Compile Include="Maps\MapResourceSpawn.cs" />
477 <Compile Include="Maps\MapTrapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
483 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
484 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
485   <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
486 </Compile>
487 <Compile Include="Networking\Client.cs" />
488 <Compile Include="Entities\NPC.cs" />
489 <Compile Include="Networking\Helpers\NetDebug.cs" />
490 <Compile Include="Networking\PacketHandler.cs" />
491 <Compile Include="General\Globals.cs" />
492 <Compile Include="Networking\PacketSender.cs" />
```

We're now going to count three lines from 480 (including 480 itself). Copy the next 3 lines of code like this:

```
473 <Compile Include="Maps\MapItemInstance.cs" />
474 <Compile Include="Maps\MapItemSpawn.cs" />
475 <Compile Include="Maps\MapNPCSpawn.cs" />
476 <Compile Include="Maps\MapResourceSpawn.cs" />
477 <Compile Include="Maps\MapTrapInstance.cs" />
478 <Compile Include="Maps\TileHelper.cs" />
479 <Compile Include="Entities\Pathfinding\Pathfinder.cs" />
480 <Compile Include="Entities\Pathfinding\SpatialAStar.cs" /> 1
481 <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" /> 2
482 <Compile Include="Entities\Pathfinding\PriorityQueue.cs" /> 3
483 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
484 <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs"> 1
485   <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon> 2
486 </Compile>
487 <Compile Include="Networking\Client.cs" />
488 <Compile Include="Entities\NPC.cs" />
489 <Compile Include="Networking\Helpers\NetDebug.cs" />
490 <Compile Include="Networking\PacketHandler.cs" />
491 <Compile Include="General\Globals.cs" />
492 <Compile Include="Networking\PacketSender.cs" />
493 <Compile Include="Entities\Player.cs" />
494 <Compile Include="Entities\Entity.cs" />
```

In that picture, the blue “1 2 3” is the top of your “code sandwich”. The black “1 2 3” is the bottom of your “code sandwich”. The patch will be looking for the top and the bottom to put the difference in between. So now that we've copied the lines shown in the screenshot, we're going to replace the bottom 3 lines of our .rej file with them, like this:

```

1 diff a/Intersect.Server/Intersect.Server.csproj b/Intersect.Server/Intersect.Server.csproj (rejected hunks)
2 @@ -480,6 +480,14 @@
3     <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
4     <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
5     <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
6 +     <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
7 +     <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
8 +     <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
9 +     </Compile>
10 +     <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
11 +     <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
12 +     <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
13 +     </Compile>
14     <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
15     <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
16     <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
17

```

After that's all done, it should look like this:

```

1 diff a/Intersect.Server/Intersect.Server.csproj b/Intersect.Server/Intersect.Server.csproj (rejected hunks)
2 --- a/Intersect.Server/Intersect.Server.csproj
3 +++ b/Intersect.Server/Intersect.Server.csproj
4 @@ -480,6 +480,14 @@
5     <Compile Include="Entities\Pathfinding\SpatialAStar.cs" />
6     <Compile Include="Entities\Pathfinding\PathfinderTarget.cs" />
7     <Compile Include="Entities\Pathfinding\PriorityQueue.cs" />
8 +     <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.cs" />
9 +     <Compile Include="Migrations\Game\20200414162817_AddingMapTypesEditor.designer.cs">
10 +     <DependentUpon>20200414162817_AddingMapTypesEditor.cs</DependentUpon>
11 +     </Compile>
12 +     <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.cs" />
13 +     <Compile Include="Migrations\Game\20200415005435_AddingMoreMapTypeVars.designer.cs">
14 +     <DependentUpon>20200415005435_AddingMoreMapTypeVars.cs</DependentUpon>
15 +     </Compile>
16     <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.cs" />
17     <Compile Include="Migrations\Game\20200414015804_AddingMoveRange.designer.cs">
18     <DependentUpon>20200414015804_AddingMoveRange.cs</DependentUpon>
19

```

Save that file, and then move it to where you put your .patch files.

Finally, you'll want to apply that .rej like you would a patch, like so:

```
git am --signoff < Intersect.Server.csproj.rej
```

If all went well, your patch should now be in and working! If you're still having problems that you can't solve, drop by the Kash Shop Support discord and let us know! We'll get that working for you!

Kash Shop Support Discord Server:

<https://discord.gg/VHJ7vnW>

At this stage, we recommend doing a **Commit**. It's needed if you're going to be applying more than 2 patches, but really you should be committing after every patch, just in case.

Thus concludes this long, exhausting documentation on how to work around lack of Source Tutorials. We hope you've got your mods/patches working, and can't wait to see what games you can now create with our Patches added into your game!