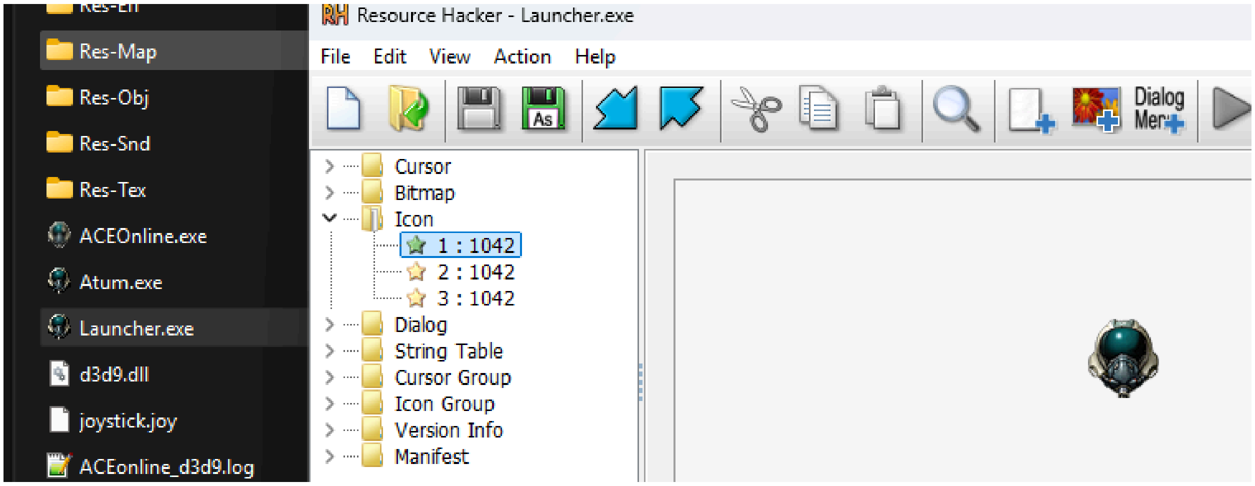
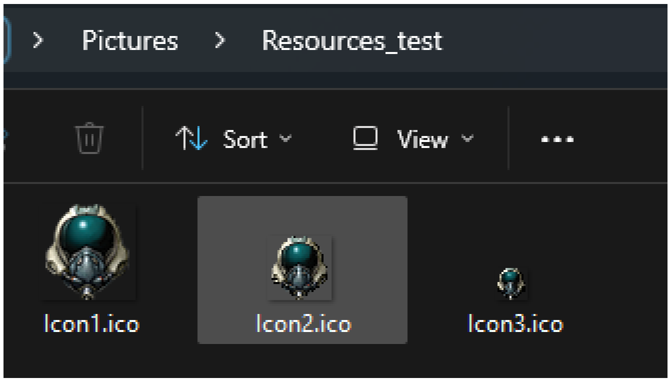
Using Resources

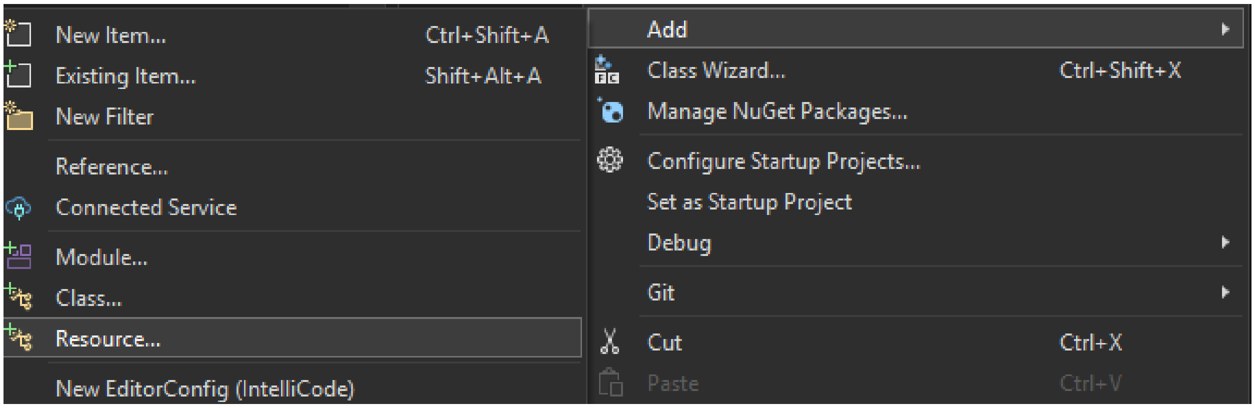
Resources is pre-defined binary data.

I tried it out by first stealing some icons from another executable using Resource Hacker

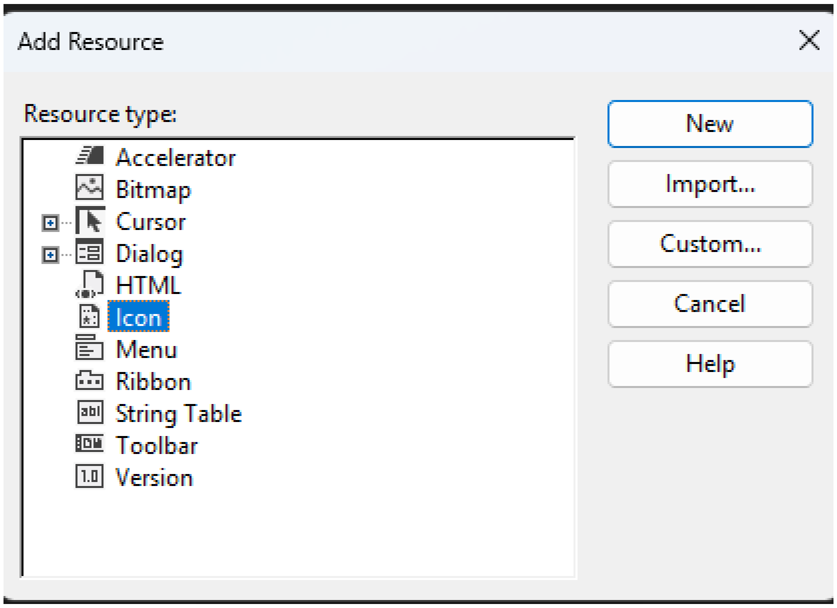




Then using VisualStudio added an `.rc` file



It opens this nice editor where I can just import stuff



It even created a `resource.h` file automatically

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by WinAPI_Project.rc
//
#define IDI_ICON1                       101
#define IDI_ICON2                       102
#define IDI_ICON3                       103

// Next default values for new objects
```

```
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        104
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1001
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

Based on Microsoft documentation About Resource Files, The steps taken to create and use Resource `.rc` files are

1. Create cursors, Icons, bitmap, dialog boxes & fonts
2. Create a `Resource-definition script` `.rc` file that describes the resources used by the application.`
3. Compile the script with the RC commandline.
4. Link the compiled `.res` file to the application executable.

The RC CommandLine can be accessed via `Visual Studio Developer Command Prompt`

```
C:\Users\Spike\Documents\Projects\WinAPI_Project>dir | find ".rc"
08/10/2025  12:34 PM             3,782 WinAPI_Project.rc

C:\Users\Spike\Documents\Projects\WinAPI_Project>rc WinAPI_Project.rc
Microsoft (R) Windows (R) Resource Compiler Version 10.0.10011.16384
Copyright (C) Microsoft Corporation.  All rights reserved.
```

```
C:\Users\Spike\Documents\Projects\WinAPI_Project>dir | find ".res"
08/11/2025  11:44 AM            14,036 WinAPI_Project.res
```

Here we can see how the `.rc` file is compiled and the resulting `.res` file is created.

A header file is needed as well when linking to the application `resource.h` this is because you want to have `#define` for the resources

```
#define IDI_ICON1                       101
#define IDI_ICON2                       102
#define IDI_ICON3                       103
```

This way we can avoid magic numbers on the .cpp code that uses the resources on the `.res` file