Basics

In UE4, all third party software, fonts, and content are documented with an associated .tps file. This is a simple XML file that contains the following info:

- What the TPS is/does
- Which folder/file this .tps file applies to
- A link to their EULA (if available online)
- P4 path of the license we redistribute to our users if it's required by that license

Determining what Engine TPS is compiled into your product

Overview

- This script will only list TPS from Unreal Engine that's compiled into your product
- This script will not reflect any TPS additions you have made
- This script is only supported when your project folder lives alongside a UE4 source build from the Epic P4 branch or Github repo
    - If using the Github repo - be sure to follow the setup instructions in the ReadMe file first

Caveats

- You can run the script on a PC for a Mac or iOS client as long as you're set up for remote compilation. Otherwise you'll need to run this script on a Mac using RunUAT.sh instead of the bat file

Run the ListThirdPartySoftware Script

- Install all necessary SDKs, otherwise the script will throw an error
- Open command prompt under <Branch>/Engine/Build/BatchFiles
- Run the following command, replacing the game target name, platform and build config state as appropriate

  ```
  RunUAT.bat ListThirdPartySoftware "-target=UE4Game|Shipping|Win64"
  "-target=CrashReportClient|Shipping|Win64" >TPSAudit.txt
  ```

- FYI - only PC, Mac, and Linux builds use CrashReportClient. Remove that target for all other platforms.
- The ">TPSAudit.txt" portion of the command will write the output of the script to a txt file.
- This txt file will contain a row for every .tps file used by the provided targets. "Redirect" .tps files are listed, but resolved down to the actual TPS they refer to so that duplicate info isn't included in the final result

Extract info from these .tps files via LinqPad

Linqpad can be downloaded here: https://www.linqpad.net/

- Remove the debug lines at the top/end of the txt file generated by the script or LinqPad will throw an error. All you should have in this txt file are the .tps file paths.
- Run the following query in LinqPad, but replace the file URL with the full path of the txt file you just generated.

```
File
.ReadLines(@"FILEPATH")
.Select(TPSEntry => TPSEntry.Split(new[] { " (redirect" }, 2, StringSplitOptions.None)[0])
.Distinct()
.Select(TPSFile => new
{
    TPSFile,
    XmlData = XDocument.Load(TPSFile)
})
.Select(TPSData => new
{
    TPSName = TPSData.XmlData.XPathSelectElement("/TpsData/Name")?.Value,
    Function = TPSData.XmlData.XPathSelectElement("/TpsData/Function")?.Value,
    Justification = TPSData.XmlData.XPathSelectElement("/TpsData/Justification")?.Value,
    TPSFile = TPSData.TPSFile,
    EULA_URL = TPSData.XmlData.XPathSelectElement("/TpsData/Eula")?.Value,
    LicenseInUE = TPSData.XmlData.XPathSelectElement("/TpsData/LicenseFolder")?.Value,
})
```

- If you hit errors, it's probably because a “redirect” .tps file wasn't implemented correctly.
- If successful, click the "export" dropdown on the right to open in Excel
- In Excel, add a filter to all cells and remove blanks from the first column. This will remove all “redirect” .tps files.