

Preprocessor Symbol Definition Files

Preprocessor Symbol Definition Files offer a simple way to manage Scripting Define Symbols in a Unity project for multiple build target groups.

Links

- [GitHub Repository](#)
- [Download Unity Package](#)
- [Asset Store](#)
- [Online Documentation](#)
- [Donate](#)

Contents

- [Technical Information](#)
- [Contact & Legal Information](#)

- [Preprocessor-Symbol-Definition-File](#)
- [Preprocessor-Definition-Settings](#)
- [Globally Defined Symbols](#)
- [Install Guide](#)
- [Uninstall Guide](#)

Technical Information

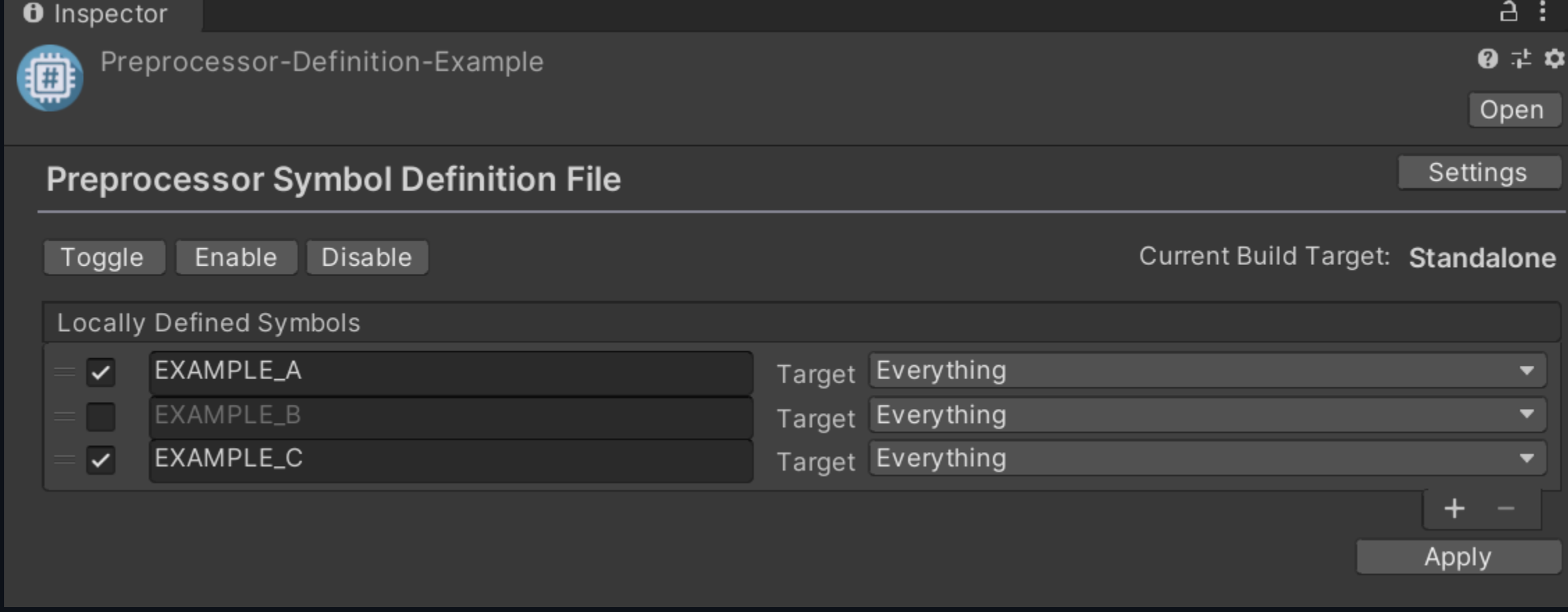
- Unity Version: [Unity 2019.1.0f2](#) or newer
- Api Compatibility Level: .NET Standard 2.0 or .NET 4.x
- Scripting Backend: Mono or IL2CPP
- Last Update: 04.09.2021

Contact & Legal Information

- Author: © 2021 Jonathan Lang
- Contact: johnbaracudagames@gmail.com
- License: MIT License

Preprocessor Symbol Definition Files

If you want to create a new file, first locate and select the folder in your **Project** window, where you would like to create the file and then navigate to (menu: **Assets > Create > Preprocessor Definition**) or (rightclick: **Create > Preprocessor Definition**). Add a new symbol by making a new entry to the **Locally Defined Symbols** list, then press apply to confirm your changes. Both adding and removing symbols must be confirmed manually. Confirmation will update unitys list of Scripting Define Symbols which will cause assemblies to reload. Entries contained in the list can be enabled and disabled manually. In addition, the **Build Target Group** of every symbol can also be set individually. Symbols will only be active if their build target group corresponds with the active build target group of the project.



Preprocessor Definition Settings

The settings file is located as an asset in your project.

To select this file manually, navigate to (menu: **Tools > Preprocessor-Symbol-Definition-File > Settings**)

Ensure that there is only one instance of this type at any time.

Remove Symbols on Delete

Removes the content of a Preprocessor Symbol Definition File when it is deleted. If this option is not enabled the symbols of a deleted file will be elevated and must be removed manually.

Log Messages

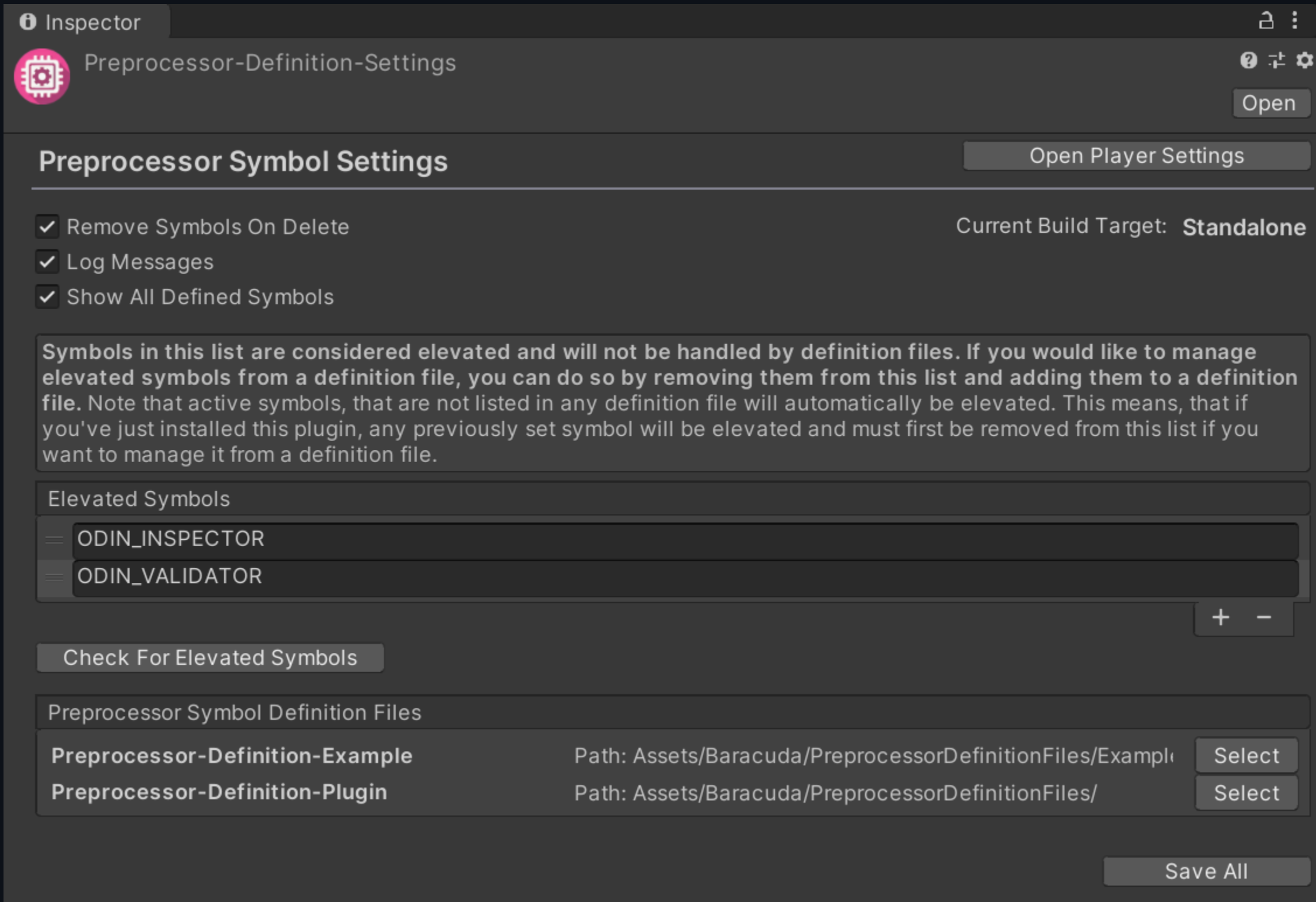
When enabled, messages will be logged when symbols are removed, added or elevated.

Save And Apply on Load

This option is only available in [Unity 2020.2](#) or newer. When enabled, unsaved changes will be applied when scripts begin to recompile.

Elevated Symbols

Symbols in this list are considered elevated and will not be handled by definition files. If you would like to manage elevated symbols from a definition file, you can do so by removing them from this list and adding them to a definition file. Note that active symbols, that are not listed in any definition file will automatically be elevated. This means, that if you've just installed this plugin, any previously set symbol will be elevated and must first be removed from this list if you want to manage it from a definition file.



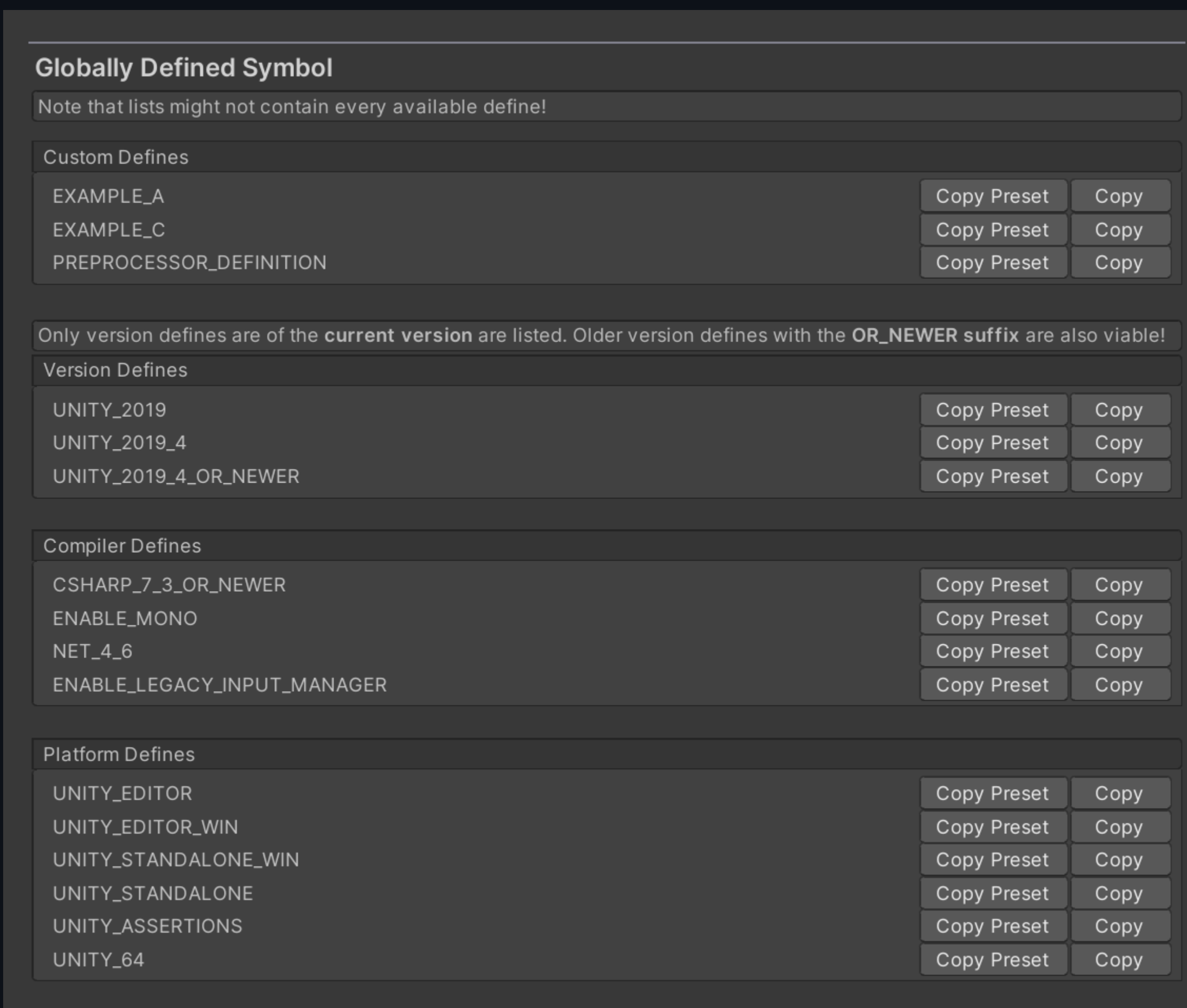
The settings file also offers a quick way to access every Definition File, located anywhere in the project.

Globally Defined Symbols

Both, definition files and the settings file can display multiple categories of defined symbols. The contents of those lists, except for the custom defines, might not contain every available symbol. Every entry offers two quick ways to copy its content to the clipboard. Use Copy for the unchanged symbol and Copy Preset to copy the symbol with the following format:

```
#if SYMBOL
```

```
#endif
```



Install Guide

Before importing this asset, please make sure that the symbol **PREPROCESSOR_DEFINITION** is not active in your project. (This could only happen if you've used this asset previously in your project)

Right now there is a known issue, that has happens in unity 2020.3.15f2 where unity will not register the assembly-definition-file of this asset correctly when re-importing. This issue can be fixed simply by forcing unity to reload the project (e.g. by restarting the project.) I was unable to reproduce this bug in another project. Please let me know if you encounter anything that could be related to this issue.

Uninstall Guide

If you are deleting individual files and **Remove Symbols on Delete** is enabled, the symbols managed by each individual file will be removed automatically. However, if you are deleting the whole package, individual symbols will not be removed. If you plan to remove this asset from your project you should proceed according to the following points.

1. Select the settings file and scan the whole project for Preprocessor-Definition-Files (press: **Validate Files**).
2. Enable or disable **Remove Symbols on Delete**, depending on whether you want to keep or remove this symbols of a file.
3. Select every definition file individually and delete it.
4. Delete the contents of this asset. (default location: **Assets / Baracuda / PreprocessorDefinitionFiles**)
5. Finally, make sure that the custom define **PREPROCESSOR_DEFINITION** is removed.

Be aware that if you delete the folder containing this asset, individual definition files that are located in your project will not be deleted, but will become zombie files that can be resurrected when this asset is re-imported. Not removing the **PREPROCESSOR_DEFINITION** symbol could also cause issues when re-importing this asset.