

# ShaderlabVS Pro

---

## ShaderlabVS Pro

[Introduction](#)

[Installation](#)

[Main Features](#)

1. [Syntax Highlighting](#)
2. [Code Completion and Basic Intellisense](#)
3. [Hover Information](#)
4. [Signature Help](#)
5. [Auto Format](#)
  - [Format Document](#)
  - [Format Selection](#)
  - [Place Open Brace On New Line](#)
  - [Macros Alignment Modes](#)
    - [Indentation with hierachy](#)
    - [Indentation without hierachy](#)
    - [No Indentation but with hierachy](#)
    - [No Indentation and without hierachy](#)
6. [Go To Definition](#)
7. [Support SRP](#)
8. [Code Snippets](#)

[CHANGELOG](#)

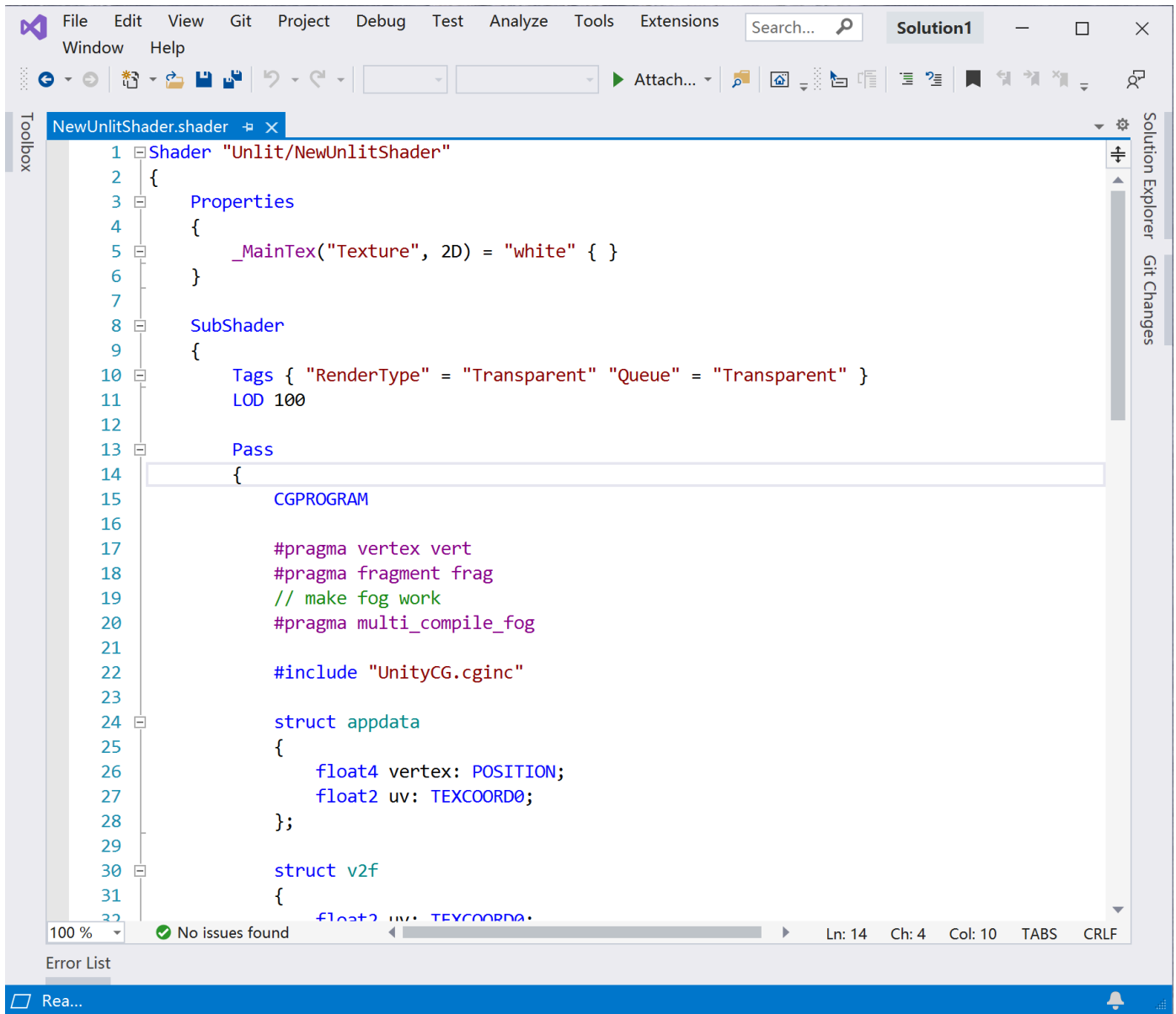
[For more information](#)

## Introduction

---

ShaderlabVS Pro is a Visual Studio Plugin for Unity Shaderlab programming.

[Documents](#) | [Forum](#) | [Email](#) | [Asset Store](#)

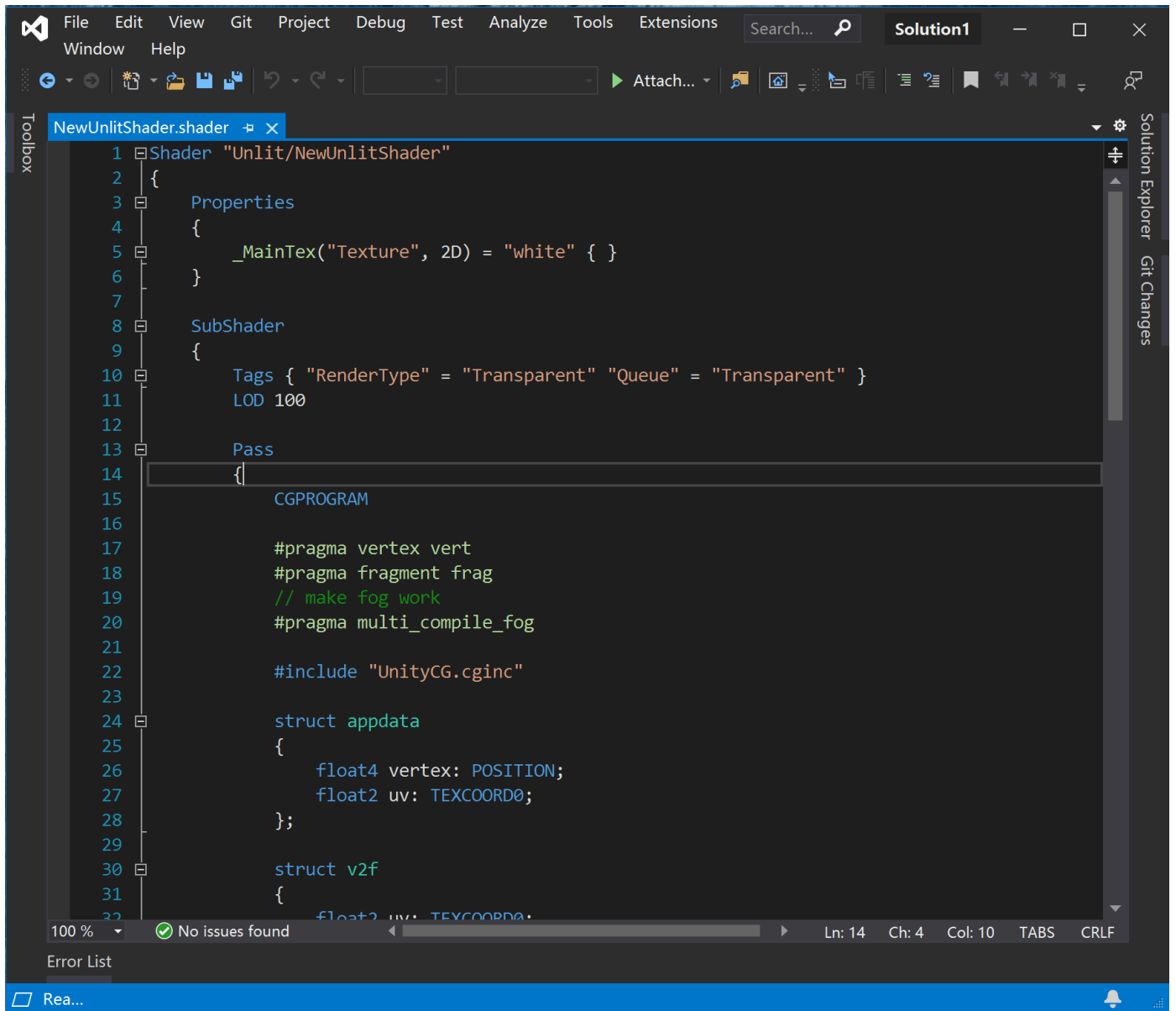


## Installation

1. Get .vsxi file under path `Assets/ShaderlabVS/VSPPlugin`
2. Double click the .vsxi file, a Visual Studio extension install window will pop up
3. Click `Install` button on install the window
4. After install successfully, restart Visual Studio

## Main Features

### 1. Syntax Highlighting



## 2. Code Completion and Basic Intellisense

```
float2 uv: TEXCOORD;
UNITY_FOG_COORDS(1)
float4 vertex: SV_POSITION;
};

sampler2D _MainTex;
float4 _MainTex_ST;

v2f vert(appdata v)
{
    v2f o;

    o.vertex = UnityObjectToClipPos(v.);
    o.uv = TRANSFORM_TEX(v.uv, _MainTex);
    UNITY_TRANSFER_FOG(o, o.vertex);
    return o;
}

fixed4 frag(v2f i): SV_TARGET
{
    // sample the texture
    fixed4 col = tex2D(_MainTex, i.uv);
}
```

### 3. Hover Information

```
texture", 2D) = "white" { }
```

---

```
derType" = "Transparent" "Queue" = "Background" }
```

Determine in which order your objects are drawn using the Queue tag.

```
AM

vertex vert
fragment frag
fog work
multi_compile_fog

e "UnityCG.cginc"

appdata

float4 vertex: POSITION;
```

### 4. Signature Help

```

44 |         o.vertex = UnityObjectToClipPos(v.vertex);
45 |         o.uv = TRANSFORM_TEX(v.uv, _MainTex);
46 |         UNITY_TRANSFER_FOG(o, o.vertex);
47 |         return o;
48 |     }
49 |
50 |     fixed4 frag(v2f i): SV_TARGET
51 |     {
52 |         // sample the texture
53 |         fixed4 col = tex2D(
54 |
55 |         // apply fog
56 |         UNITY_TRANSFER_FOG(o, o.vertex);
57 |         return col;
58 |     }
59 |     ENDCG
60 |
61 | }
62 |

```

▲ 1 of 16 ▼ float4 tex2D(sampler2D samp, float2 s)  
performs a texture lookup in a given 2D sampler and, in some cases, a shadow

## 5. Auto Format

### Format Document

via menu `Editor -> Advanced -> Format Document`

### Format Selection

### Place Open Brace On New Line

We can set brace formatting style in **Tools -> Options -> Text Editors -> Shaderlab -> Formatting** page.

If `Place Open Brace On New Line` is `true`, after formatting, open brace will be in new line, like below

```

if (true)
{

}

```

if `Place Open Brace On New Line` is `false`, after formatting, open brace will not in new line.

```

if (true) {

}

```

## Macros Alignment Modes

In Settings, there is an item under `ShaderlabVScode` section named `Formatting: Style`, change the `Macros alignment modes` to set formatting mode for macros.

### Indentation with hierachy

```
Subshader
{
    Pass
    {
        CGPROGRAM
        void MacroTest()
        {
            float c;
            #if 0
                c = 0;
            #if 1
                c = 1;
            #endif
            #endif
        }
        ENDCG
    }
}
```

### Indentation without hierachy

```
Subshader
{
    Pass
    {
        CGPROGRAM
        void MacroTest()
        {
            float c;
            #if 0
                c = 0;
            #if 1
                c = 1;
            #endif
            #endif
        }
        ENDCG
    }
}
```

## No Indentation but with hierachy

```
Subshader
{
    Pass
    {
        CGPROGRAM
        void MacroTest()
        {
            float c;

#if 0
            c = 0;
#endif
#if 1
            c = 1;
#endif
        }
        ENDCG
    }
}
```

## No Indentation and without hierachy

```
Subshader
{
    Pass
    {
        CGPROGRAM
        void MacroTest()
        {
            float c;

#if 0
            c = 0;
#endif
#if 1
            c = 1;
#endif
        }
        ENDCG
    }
}
```

## 6. Go To Definition

```

v2f o;

o.vertex = UnityObjectToClipPos(v.vertex);
o.uv = TRANSFORM_TEX(v.uv, MainTex);
UNITY_SETUP_STEREO_EYE_VRT(v.vv2);

return o;
}

float3 GetColor(v2f i)
{
    // sample the texture
    fixed4 col = tex2D(MainTex, i);
}
```

## 7. Support SRP

```

#pragma vertex vert
#pragma fragment frag
// make fog work
#pragma multi_compile_fog

#include "UnityCG.cginc"
#include "Packages/com.unity.render-pipelines.core/"

struct appdata
{
    float4 vertex: POSITION;
    float2 uv: TEXCOORD0;
};

struct v2f
{
    float2 uv: TEXCOORD0;
    UNITY_FOG_COORDS(1)
    float4 vertex: SV_POSITION;
}
```

- CHANGELOG.md
- LICENSE.md
- package.json
- ValidationConfig.json
- Documentation~
- Editor
- Runtime
- ShaderLibrary
- Tests

## 8. Code Snippets

Below are the snippets:

Snippets	Description



blend1-dc_1	Blend OneMinusDstColor One
blend1_1-sa	Blend One OneMinusSrcAlpha
blend1_1	Blend One One
blenddc_0	Blend DstColor Zero
blenddc_sc	Blend DstColor SrcColor
blendsa_1-sa	Blend SrcAlpha OneMinusSrcAlpha
cgp	CGPROGRAM...ENCG
for	<code>for</code> loop
forr	Reverse <code>for</code> loop
glp	GLSLPROGRAM...ENCGLSL
hlp	HLSLPROGRAM...ENDHLSL
if	if { ... }
ifelse	if {...} else {...}
inc	<code>#include ""</code>
incpkg	<code>#include "Packages"</code>
incucg	<code>#include "UnityCG.cginc"</code>
prop2d	<code>2D</code> type property
propc	<code>Color</code> type property
propcube	<code>Cube</code> type property
propf	<code>Float</code> type property
proprange	<code>Range</code> type proprety
props	Properties { ... }
propv	<code>vector</code> type property
region	<code>//region ... //endregion</code>
region2	<code>//#region ... //#endregion</code>
shader	Shader { ... }
struct	struct { ... };
subshader	SubShader { ... }

tags	Tags { ... }
tagtt	Tags with both of RenderType and Queue is Transparent

## CHANGELOG

---

### v1.1.2

- Add supports for shaders inside Packages folder
- Add supports for Visual Studio 2020 Preview
- Fix bug that some fields of struct are missing in code completion when struct contains macros
- Fix minor bugs in auto format

### v1.1.1

- Improve macros formatting in AutoFormat
- Fix exception when `place brace in new line` is `false` in Auto Format
- Fix bug that code completion of struct with method members is not correct

### v1.1.0

- Add `Format Selection` feature
- Better HLSL Supports that add more keywords and builtin methods
- Better Outlining that support regions and program blocks
- Better macros formatting. We can change different styles in `Macros Alignment Modes` settings
- Better Syntax highlighting for types and methods
- Fix bugs that StructuredBuffer show as Buffer type in hover information
- Fix some formatting bugs that breaking shader compile

### v1.0.5

- Better code completion match
- Add version update notification
- Fix some minor bugs

### v1.0.4

- Improve highlighting for all occurrences will ignore comment line
- Improve performance for large files
- Fix bug that highlighting is not work for last comment line of code

### V1.0.3

- Add `placing open brace on new line` formatting style
- Add type info in hover information
- Fix bug that variables in method cannot found in `Go To Definition`
- Fix type name is wrong for some variables/members in some scenarios

### V1.0.2 Beta3

- Add `Go To Definition` features for methods/variables/types
- Add highlighting for all occurrences for selected word
- Add signature help support for method defined by macros
- Improve brace match marker background color
- Fix bug that method defined by macros are missing in completion
- Fix bug that duplicated completion item in Code Complete
- Fix bug that built-in included shader libraries are missing

### V1.0.1 beta2

- Improve syntax highlighting
- Add code snippets support

### v1.0 beta1

init commit

## For more information

---

Visit site <http://www.amlovey.com/shaderlabvs>