echoLogin PostFX Studio

For support, please email core@echologin.com and include your invoice number in the subject.

Alternatively, support can be reached via Skype by adding 'echoLogin' to your contacts. Please include your invoice number in the contact request.

To get the most out of PostFX Studio, it is recommended to watch the related tutorial videos.

If this is your first time using PFX Studio, it is also a good idea to crack open the sample projects and take a look at what's going on. The sample projects are located at *EchoLogin/PFX/SampleProjects*

Quick-start

If upgrading from version 1.x, you must completely remove the 1.x files before installing version 2.x (Delete the 'echoLogin/PFX' folder) as well as any managers in your scene.

If upgrading from a previous 2.x beta, export your settings to XML and remove all the beta version files and any managers from your scenes. After following the steps below, import the saved XML file and it will try to preserve as many settings as possible. Some things may be lost and have to be reset.

1. Import the PostFX Studio package into your project
2. Add a new echoPFXManager to your scene by going to GameObject > Create Other > echoLogin > _echoPFXManager.
3. Select the newly-added _echoPFXManager in your project hierarchy

Note: *Be sure to create a new _echoPFXManager for each scene. Doing so allows you to create unique shaders and effects for each scene. You will notice that after compiling, the group of shaders in the resources folder are named after the current scene you are working in.*

While the manager may appear daunting at first, you will see that it is straightforward to set up once reading through the settings and values descriptions:

Basic Settings

Manager Depth

Any cameras in your scene that you want overlaid on top of effects *without* being affected by them should have a depth value higher than this. A good example would be if you have a HUD rendered on top of everything that you don't want effects on. It is recommended to just start out those cameras at a high depth such as 10.

Any camera with a depth setting of less than 0 will be rendered in the background with no PostFX. There are a few things to setup before using PostFX in this way:

-First you must set the Pass1 and/or Pass2 Alpha Channel setting to ALPHA_ON, and in the final pass set the blend mode to Transparent.
-This RenderGroup's first camera needs to "clear" like any normal camera, but it needs to be set to Solid Color, Black, with 0 Alpha (100% transparent).
NOTE: *this is an uncommon use for PostFX and could cause your game to run a little slower.*

Frame Rate

Set this to your desired frame rate. There's no need to call Application.targetFrameRate in your script.

This is also used when setting auto detail mode to maintain a set frame-rate.

Auto Include Cameras

When this option is checked, the manager will add effect cameras to the proper group by using the RenderGroup's ID as a camera depth. For instance, if the first render group has an ID of 0, it would use all cameras from 0.0 up to but not including 1.0.

Enable PlayMaker Actions

Toggle this on when using PlayMaker - it will add the PFX Actions under "echoLogin PFX" in the PlayMaker Action browser.

Javascript Support

This copies the necessary files to Assets/Plugins so that you can use JavaScript with PostFX Studio.

Export Button

Use this feature if you'd like to save all settings to an XML file for backup purposes or for sharing your effects with others. It is recommended to backup your effects and settings when upgrading to newer versions of PostFX Studio.

Import Button

Loads a saved XML file. This will replace all current settings with the backed up settings from a previous export.

Render Groups

A Render Group is a group of cameras used for PFX Studio that share common effects. You can use any amount of Render Groups, but using more will slightly impact the frame rate each time.

Add Render Group
This button will add a new Render Group. Pressing the "-" button will remove it.

Effects

Post Effects are the heart and soul of PostFX Studio. They are built from the effect options under the Shader and Effect tabs. See following section for more details.

Add Effect
This button will add a new effect. Pressing the "-" button will remove it.

Render Group Options

Here is where you will set various options to affect screen render texture, assign options to be included in the post effect shader, and make the actual effects.

Group Tab

This is where you can set the options for the selected Render Group.

Camera Depth Range *Note: Only available when Auto Include Camera option is selected*

This is only available when the Auto Include Camera option is selected. Use this to set the desired depth range;

Render Group Name

The name of the selected render group. It is a good idea to name this something meaningful.

Screen Mesh Quads

Setting for how many quads that will make up screen for Distortion and Shockwave effects. If you're not using those options set this to the lowest number (1 is lowest).

Render Texture Size

`DEVICE_SIZE`	RenderTexture will be the size of the Device's Screen.
`DIVIDE`	Divides Device Screen size by value.
`AUTO_DETAIL`	Maintains 60 fps on slower machines by changing detail per frame.
`CUSTOM`	Set the Screen Size to a custom value.

Drag And Drop Camera(s) here

All cameras that you want to be used in a render group must be added here. Simply drag them from the scene. You can also use the arrows to change the draw order. Top cameras are drawn first, bottom ones are drawn last.

Shader Tab

Important: After making modifications underneath the shader tab, you will notice the “Compile Post FX Shaders” button turn red. *You must compile the shaders for any changes to take effect.*

Under the Build tab are the global shader options to be included for the selected Render Group

Copy 1
This will copy the screen before any effects have been applied to the next render group

Pass 1
This is first effect pass - this is what will be used most often

Copy 2
This will copy the screen after Pass 1 effects have been applied to next render group

Pass 2
This pass is processed after Pass 1

Frag Precision

The higher the precision, the more performance is impacted

`FIXED`	Low precision
`HALF`	Medium precision
`FLOAT`	High precision

Blend Mode

Blend mode for this pass, will automatically be disabled for the first Render Group.

Alpha Channel

Adds alpha channel support for when overlaying render group output on the screen.

Filter Mode

Point, Bilinear or Trilinear.

Add Option to Shader

Choose from a list of options that you would like to add to the shader.

DISTORTION	Waves, ripples, shockwave, bulge, pinch and fisheye.
GREYSCALE	Used also for de-saturation.
COLOR	Blend modes: color, add, multiply, screen and overlay
COLOR_CORRECT	Use a texture to adjust screen color.
CONTRAST	Can be global or per color channel.
GAMMA	Can be global or per color channel.
POSTERIZE	Limits colors and can have different values per color channel.
INVERSE	Invert colors on screen.
RGB_SEPARATE	Simple chromatic aberration effect.
LUMRAMP	Use a ramp texture based on pixel luminance (example thermal vision).
SCANLINES	Horizontal and/or vertical lines.
AVERAGE_PIXEL	Average surrounding pixels (used mainly for smoother bloom).
TEXTURE	Blend modes: normal, add, subtract, multiply, screen and overlay.
CUSTOM_FRAG	Enter custom shader code.

Global Shader Options

Field 1
If an E appears here this means this option is used in the currently selected effect

Field 2
Shader option name, pressing this button will bring up the sub-options for this shader effect.

Field 3
Always on. When highlighted, the effect will always be executed and will not use up one of the six shader keywords for this pass.

Field 4
Move up in render order *(please note that some options cannot be reordered)*

Field 5
Move down in render order *(please note that some options cannot be reordered)*

Field 6
"-" Remove this option

Sub Options *(please note that not all of the shader options have the same sub-options)*

Use Unique TexCoord
This will let you scroll a shader option that uses a texture.

Blend Mode
This lets you set the blend mode for shader options that use a texture.

Mask
This is a texture that is used to mask a shader effect. If the same texture is used on multiple shader options, the mask texture is only read once.

Mask Dissolve Effect
When checked, the mask's alpha channel is used for a dissolve data.

Shader Fade Option

`DEFAULT`	Lets you fade effect from 0.0 through 1.0
`ALWAYS_1`	The effect's fade is hardcoded to always be 1.0
`FASTER_WHEN_1`	This uses an extra shader keyword to switch to faster code when the effect is at 1.0 fade level (this can cause an "out of shader keywords" error if overused).

Value Blend
This is how values are combined when multiple shader options of same type are in effect simultaneously.

`DIRECT`	Values set directly so the last active shader options value will be used.
`ADD`	Adds values per frame.
`MULTIPLY`	Multiplies values per frame.
`AVERAGE`	Averages values per frame.

Effects Tab

This is where you create the actual effects and configure its options. Use the Add Effect Option drop-down menu to select which shader options to use and expose the settings for that option.

Timing

Delay
The delay before this effect option starts.

Fade In
This is how long it takes (in seconds) for the effect to fade in.

Sustain
This is the time the effect will remain on. Enter -1 if you want effect to stay on this stage. You can use scripting *(see the Scripting section)* to stop the effect.

Fade out
The time it takes for the effect to fade out.

The Curve fields let you alter how each stage performs. Default is linear.

Duration
The total duration of the effect option.

Hold on Sustain
If this is checked, the effect will loop on the sustain stage forever. It can only be ended via script where it will then go through the fade out stage and completion.

Strength
This value is the current fade * strength. For example, a 0.5 strength will make the fade in/sustain/fade out values halve.

Specific Effects Options

Color Options

COLOR OPTIONS
Choose color or gradient texture. Using a gradient texture is the same speed as using a solid color, it does not incur a texture read in the shader.

Amplify
This will let you make colors/gradients overbright.

Texture Options

TEXTURE
Choose texture for overlay

Scanlines Option

HORIZONTAL
Divide screen height by count
When checked, the count is a divisor and when unchecked it is a literal count.

Count
Enter a non zero value

Scroll
A positive or negative value to scroll the horizontal scanlines

VERTICLE
Divide screen width by count
When checked, the count is a divisor and when unchecked it is a literal count.

Count
Enter a non zero value

Scroll
A positive or negative value to scroll the vertical scanlines

Scripting

PostFX Studio allows for the control of effects through code:

Start()

Stop()

DistortionCenter() (only used for distortion effects that have a center)

Example

```
private EchoPFX _fx1;
 //=============
 void Start()
 {
    // always do this on startup.
    _fx1 = new EchoPFX ( "RenderGroup:0", "MyEffect" );
 }

 // normal effect
 //=============
 public void TriggerPostFX1()
 {
    _fx1.Start();  // can also called  _fx1.Start( timescalevalue );
 }

 // if the fx was a shockwave
 //=============
 public void TriggerPostFXShockwave( int ipass, int iid, float ixper, float iyper )
 {
    _fx1.DistortionCenter ( ipass, iid, ixper, iyper );
    _fx1.Start();  // can also called  _fx1.Start( timescalevalue );
 }

 // to stop an effect call Stop,  this is only needed for effects with a -1 sustain ( infinite )
 //=============
 public void StopPostFX1()
 {
    _fx1.Stop();  // you can also call Stop fx1.Stop ( fadeouttime );
 }
```

Public Functions

The following are the available functions in an EchoPFX object:

```
//=============================================================
public void DistortionCenter ( int ipass, int iid, float icenterx, float icentery )

   //=============================================================
public void  DistortionCenter ( int ipass, int iid, int iscreenx, float iscreeny )

//=============================================================================
public void SetColor ( int ipass, int iid, Color icolor, float iamplify = 1.0f )

//=============================================================================
public void  SetTexture ( int ipass, int iid, Texture2D itex )

//=============================================================================
public void SetCustom ( int ipass, int iid, Vector4 iparam1, Vector4 iparam2 )

//=============================================================================
public void SetCustom ( int ipass, int iid, Vector4 iparam1 )

//=============================================================================
public void Start()

//=============================================================================
public void Start( float iscale )

//=============================================================================
public void Stop()
```