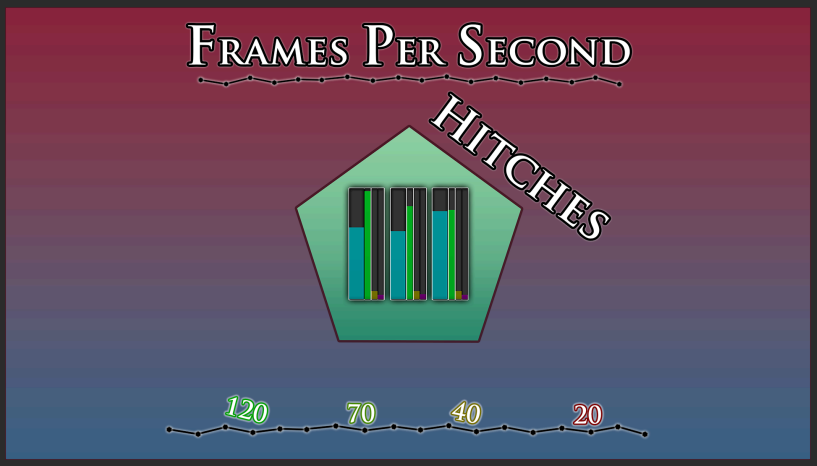


FPS & Hitches Tool Documentation

4.26+

The FPS & Hitches tool allows you to add an fps meter to your screen that can record hitches, visualize them and record events in the editor or in any build, even shipping.

There's an accompanying editor utility widget that will allow you to read all the data saved during play-sessions.

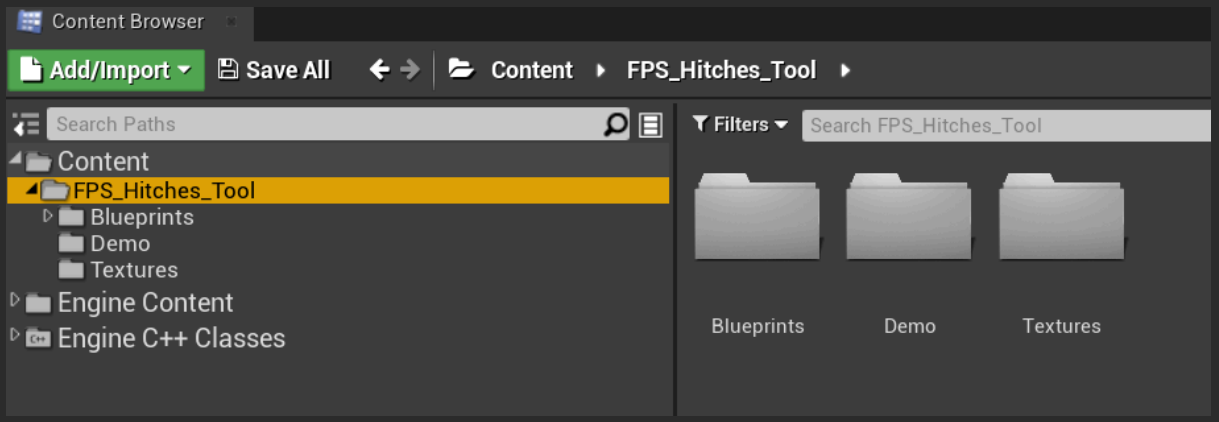
Videos

Overview
Tweaking the fps widget
Capturing FPS data
Add Hitch events

Table of Contents:

Clean-up

When you open the project for the first time you'll be greeted by three folders:



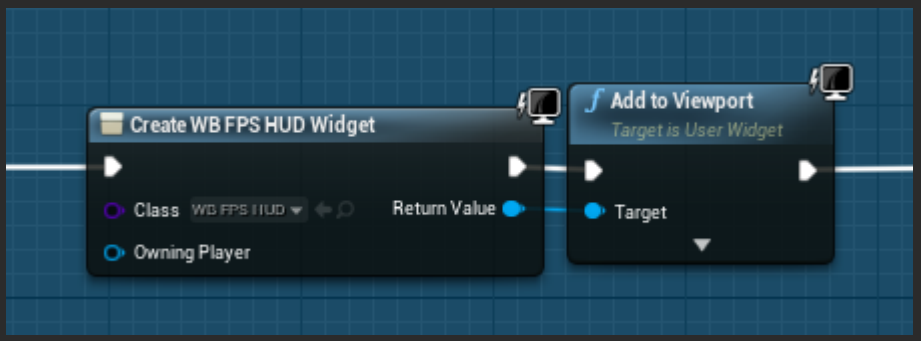
Feel free to **delete** the demo folder unless you want to have an example of how to send hitch FPS event data to the FPS widget.

The textures folder only contains the two images for highest and lowest FPS, feel free to replace these with better looking art.

Demo

Demo -> This map contains a lot of static meshes.
BP_Demo_Actor ->

- This actor will add the FPS counter to your screen.

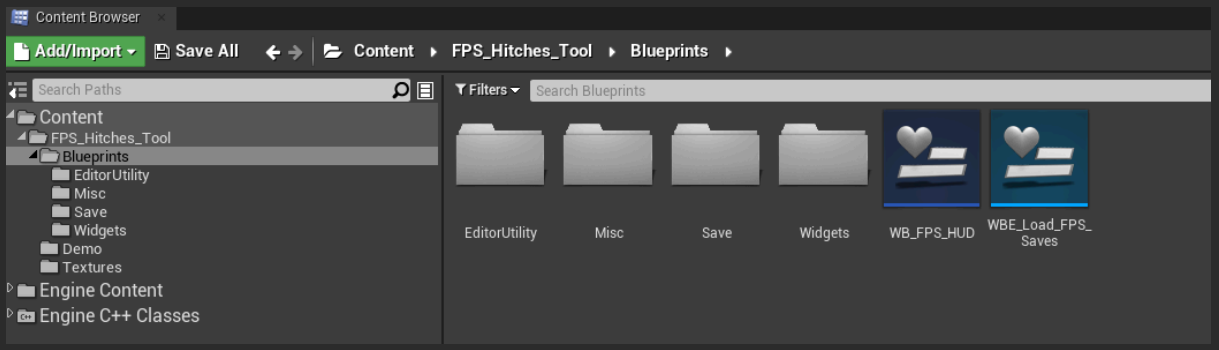


- It will then randomly set a maximum FPS every 4 seconds. (Faking hitches)
- It will try to move every static mesh in the level every 1.5 seconds. (Faking hitches)

This is all there is in the demo folder to showcase the content of this package.

Main Widgets

In the blueprints folder you'll find 4 folders and two widgets.



WB_FPS_HUD -> The widget you should add to your hud that allows you to quickly make changes.

WBE_Load_FPS_Saves -> The editor utility widget that allows you to load old saves and watch the data gathered by it.

FPS Hud

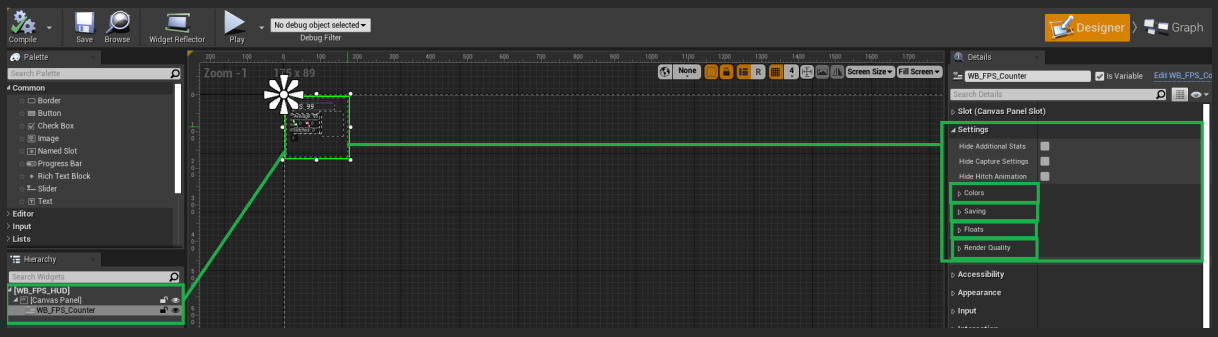
You should add this to your viewport when playing if you want to see the FPS in game.

When adding the FPS meter to your screen you can see the current FPS in the top left.

There's a tiny arrow that you can click on when in a menu, this will drop down the additional info:
Average:
Max|Min:
Hitches:

Below that is another arrow you can select, this will open the capture FPS screen that can run through different settings capturing the FPS stats.

Inside the widget:
When selecting the WB_FPS_Counter in the hierarchy overview you can make changes in the details panel under **settings**.



Settings

- **Hide Additional stats:** *This will disable the drop-down menu that comes with the FPS meter.*
- **Hide Capture settings:** *This will only hide the capture button that allows you to capture the FPS using different quality settings. (These will be hidden by default if Additional stats are hidden)*
- **Hide Hitch Animation:** *This will hide the FPS animation that plays when you have a hitch. (You'll see the FPS of the hitch fall down from the FPS meter making it more obvious)*

Settings/Colors:

- **Use color for FPS:** *When disabled all text will have the no color default color.*
- **Colors:** *These should speak for themselves.*

Settings/Saving:

- **Save FPS:** *When disabled, the game will no longer save FPS data all-together. It will simply just show it to you when playing. (this will prevent you from loading in the data in the editor)*
- **Save Hitches:** *When disabled, the game will no longer save hitch events in the save files.*

Settings/Floats:

- **Hitch FPS Difference:** *What difference in FPS per second is considered a hitch.*
- **Min FPS For Hitch Detection:** *The minimum FPS before a hitch is allowed to be recorded.*
- **Min FPS Force Hitch:** *When FPS hits this or lower than this value it will always be recorded as a hitch.*
- **Average FPS update rate:** *How fast should the average FPS value update.*
- **Time Before Remove Event:** *How long before and after should a hitch record events.*

Settings/Render Quality:

- **Epic - Low:** *Here you can exactly see what console commands are executed for every setting of the FPS recording process. You could add any new console command to the list.*

Predicting Hitch data

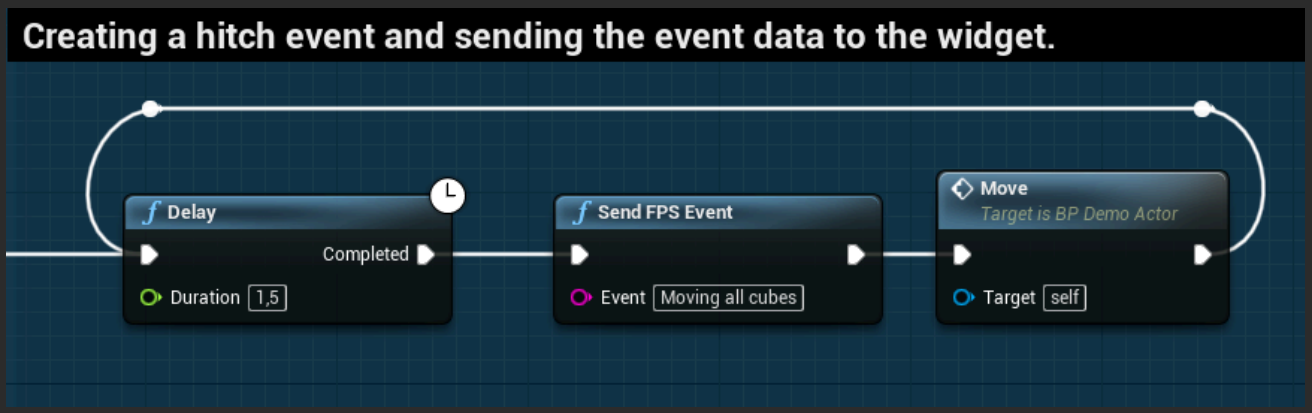
When you're checking for hitches you probably have some idea of where they could come from. You just want to know what event is happening when.

To associate events with the hitches that happen you'll have to add functions to your actors where you expect hitches to happen. *(By default this FPS event is stored for 0.5 seconds, if a hitch occurs, the event will be associated with the hitch and it will show up in the list later on)*

This event is called **Send FPS Event** and **can be called in any actor** as it's part of a blueprint function library. Just add it to your actor and give it a name that tells you what event is happening.

Example:
In the image below I added an event that will try to move every static mesh in the level 5 times. *(Forloop + for each loop)* I know that this is an intense event to ask so I want to see if this causes for a hitch when I create a shipping build.

I add the "**Send FPS Event**" and give it a name that will make it easier for me to recognize what is happening if the hitch does show up.



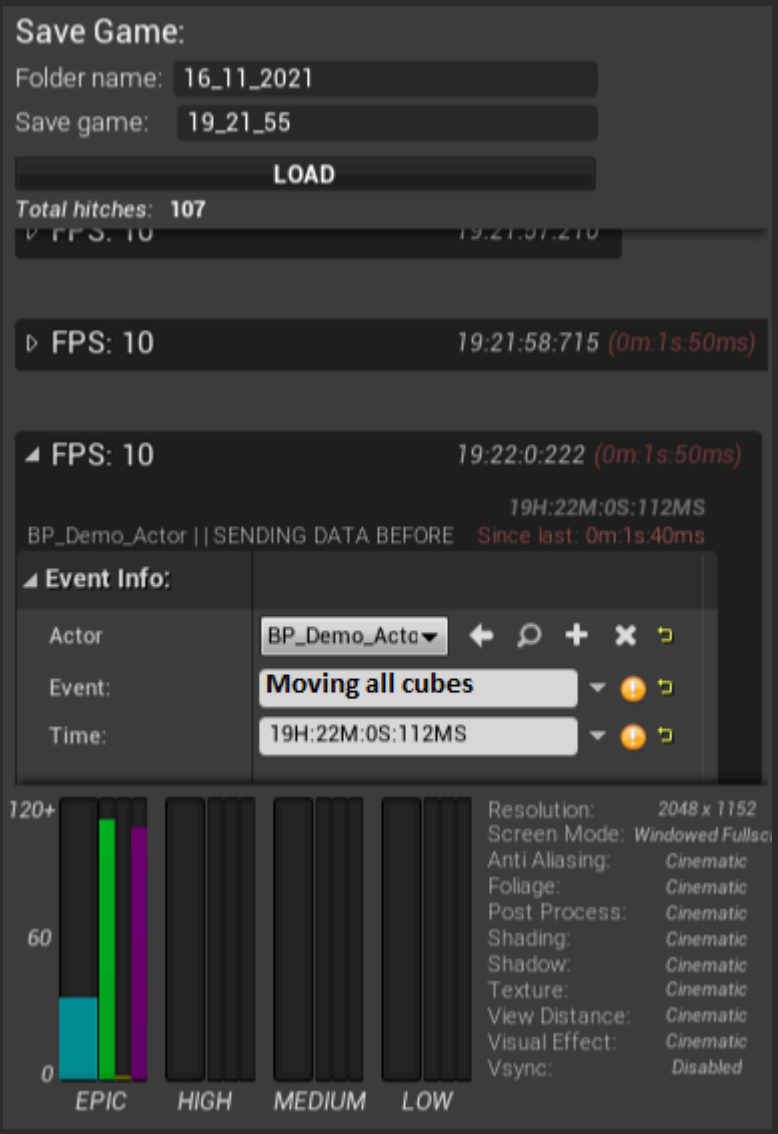
I made sure saving is enabled, I build, play and save. Move over the save files and now it's time to check.

On the next page we'll have a look at the results.

Editor widget

You can press right mouse button on the WBE_Load_FPS_Saves and then press **Run Editor Utility Widget.** This will open the editor utility widget which you can then dock anywhere else in your layout.

When hovering over a lot of the UI you'll get tooltips that tell you what the numbers mean.



Folder name: This is automatically set to the current date. (*Day/Month/Year*)
Save game: Here you should enter the name of your save game that you want to load.
Load: Loads the save game data.
List: A list with saved hitch data will be generated. If there was a valid event being tracked at that time it will show up as you can see on the left. It also shows you the time difference between this hitch and the one before. *(in case you don't have an event this can help you track down the issue)*
Event Info: If valid it will let you know what message was called from what actor and at what time.
Graph: When running a capture you can load in and compare the average FPS, highest, lowest and amount of hitches. They will be presented in a graph. The settings on the right are obtained from the *"Get Game user settings*"

Making core changes

Everything has been made using widgets and editor utility widgets without requiring you to enable any plugin. Feel free to make changes and experiment to make sure it fits your project perfectly.

Examples I implemented for a client project:

- Append the save name with the game mode. (So I always know where the FPS was recorded.
- Add level name as a saved variable in BPS_FPS_Save to ensure I know what level is active.
- Made the Max allowed FPS higher during FPS recording session and restoring it after it is done.

These are just a few examples of what you could add. If you have any questions please feel free to reach out through **discord!**

Shipping build

If you need access to FPS save data from a shipping build you'll have to go here:

%LOCALAPPDATA%/ProjectName/Saved/SaveGames/FPS_Stats

Copy that folder over into your project Saved/SaveGames folder.
If you have a lot of hitches you may want to disable the saving feature for builds that you share with a big group of people. The files don't get that big but it adds up and you might not want to bother users with these files. (One is created per play-session after all)

Patch notes

V1.2 Update

- Preventing hitch data from being saved when saving is disabled.
    - (Prevents warnings from popping up)

V 1.1 Update

Overhauled hitch tracking system

- FPS events can now be used in widgets as well
- Hitch tracking is less performance intense and completely overhauled.
    - *Can now proccess thousands of hitch events per second without impacting performance itself. (Up from a few dozen)*

***** Rating! *****

If you enjoy the package please consider leaving a rating, it helps out with visibility and it's a huge motivator for me to keep going.

If you have recommendations or would like to see changes please don't hesitate to reach out to me on discord! https://discord.gg/kdqVrWZ

Thanks for your support.

- Tim van Kan