Fly Elise-ng B.V.
Waterstad 31
5658RE Eindhoven
The Netherlands
Web: http://fly.elise-ng.net
Email: info@elise-ng.net
Tel: +31 (0)40-7114293

Fly Elise-ng
Immersive Display Frame Interpolation
AI Assisted Frame Interpolation

Copyright © 2012/2024 Fly Elise-ng, All Rights Reserved

1 Introduction 3
2 Fly Elise-ng AI based frame interpolation 4
3 Examples frame interpolation configurations 6
3.1 Prepar3D 6
3.2 Microsoft Flight Simulator MSFS2020 9
3.3 X-Plane 11/12 12

1 Introduction

Frame interpolation, a technique that generates additional frames between existing ones to enhance visual smoothness and reduce motion blur, has become a crucial component of modern gaming. NVIDIA's DLSS 3 and AMD's FSR 3 are two leading contenders in this space, each offering unique approaches to improve gaming experiences.

NVIDIA DLSS 3

DLSS 3, the latest iteration of NVIDIA's Deep Learning Super Sampling technology, introduces a feature called Frame Generation. This technology leverages AI to create entirely new frames, effectively doubling frame rates without compromising image quality. DLSS 3 achieves this by analyzing the scene and predicting the motion of objects, allowing it to generate realistic intermediate frames. However, DLSS 3 requires RTX 40 series GPU.

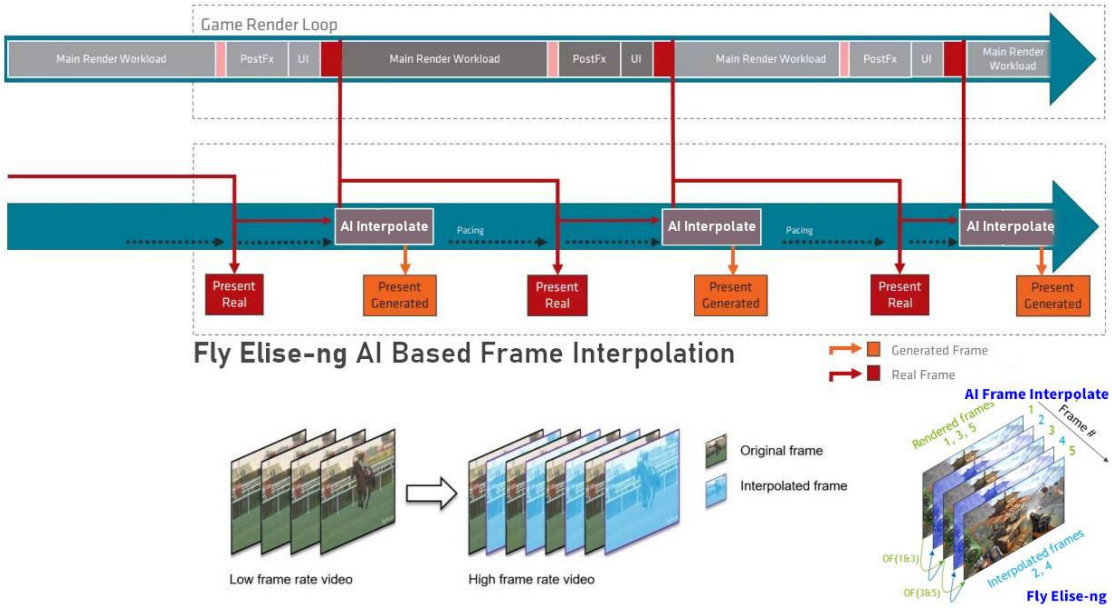
AMD FSR 3

AMD's FSR 3 also offers frame interpolation capabilities, but it takes a more traditional approach. FSR 3 utilizes motion vectors and optical flow analysis to predict intermediate frames between existing ones. While it doesn't offer the same level of frame generation as DLSS 3, FSR 3 still provides significant improvements in smoothness and reduces motion blur.

Both NVIDIA DLSS 3 and AMD FSR 3 are powerful tools for enhancing gaming experiences by increasing frame rates and improving visual quality. The choice between the two ultimately depends on individual preferences, hardware compatibility, and specific game requirements.

However, both NVIDIA DLSS 3 and AMD FSR 3 require tight integration with the game engine. Both require additional information from the game engine for each rendered frame. The game engine needs to calculate and provide motion vectors and scene depth information along with the rendered frame to DLSS 3 or FRS 3 engines. Although the set of games engines that support DLSS 3 or FRS 3 is growing, there is a huge amount of game engines that are not able to integrate with DLSS 3 or FRS 3. In addition, DLSS 3 requires at least RTX 40 series GPU and does not work on RTX 30 or older GPUs.

2 Fly Elise-ng AI based frame interpolation



Fly Elise-ng adds a seamless integration of frame interpolation for any game engine based on DirectX, OpenGL or Vulkan. It **does NOT require motion vectors or any other motion flow information from the game engine**.

Fly Elise-ng frame interpolation is based on a set of novel, highly optimized AI-trained algorithms executed on the GPU between each frame generated by the game engine. This effectively doubles the output frame rate (2 x FPS) while allowing the game engine to spend more time in processing and generating the game frames.

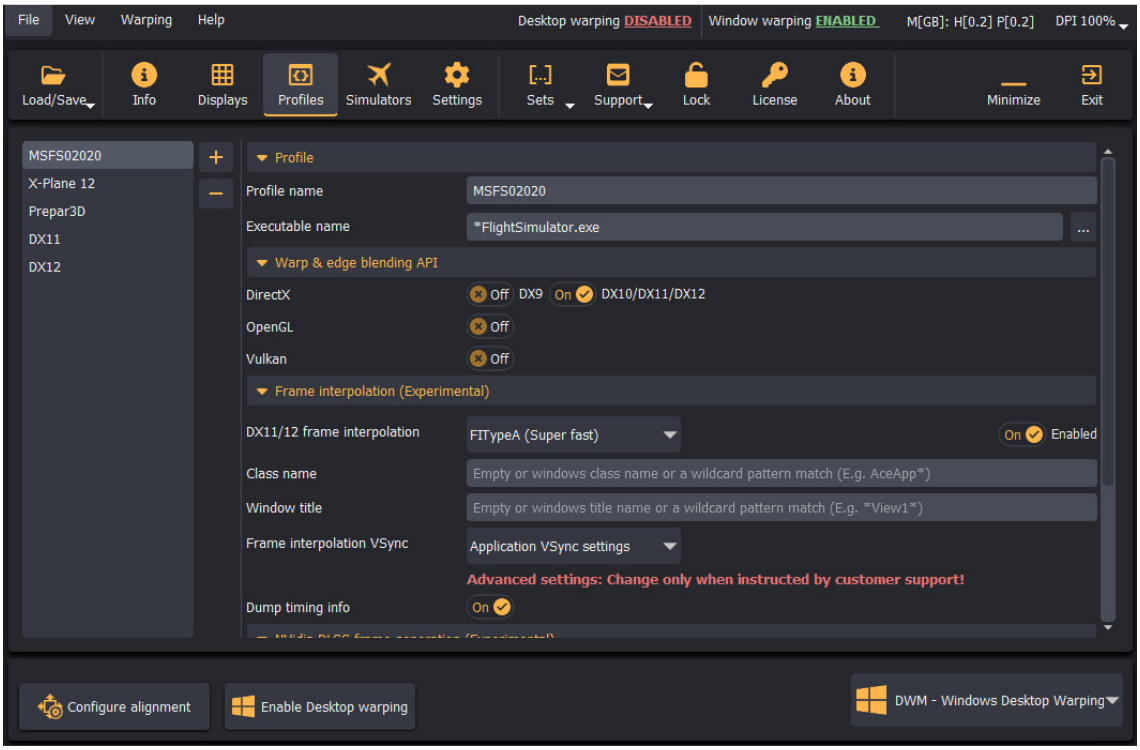
Frame interpolation takes only a fraction of the time needed to process and render the game frame. On a typical setup with 60 Hz displays, the game engine has a deadline of 16 ms to prepare and render each frame.

Our frame interpolation algorithms take between 0.5 and 3 ms per frame on the GPU, thus allowing the game engine to spend double the time per frame.

Just limit the application frame rate and increase rendering quality in the game. The game can render at 30 FPS, and for each game frame we will generate an interpolated frame and ensure both the real game frame and the interpolated frame are paced and presented with 60 FPS.

Another use case is ignoring the VSync and simply doubling the game frame rate. Let the game render as many frames as possible (unlimited FPS) and for each game frame we will generate an interpolated frame and ensure both the real game frame and the interpolated frame are paced and presented with 2 x FPS.

Frame interpolation can be enabled and configured per game, using the Immersive Display PRO – Ultimate or Immersive LCD PRO – Advanced Profile settings.



Two frame interpolation algorithms are available: FITypeA (Super fast) and FITypeB (Super accurate) with a simple tradeoff between speed and fidelity.

In some specific cases the interpolation can be limited to a specific application window based on the windows class name or window captions. Those can be defined using full explicit names or using the wildcards “*” and “?’. They can match any or exactly one character. For most of the cases, the windows class and title names can be left empty.

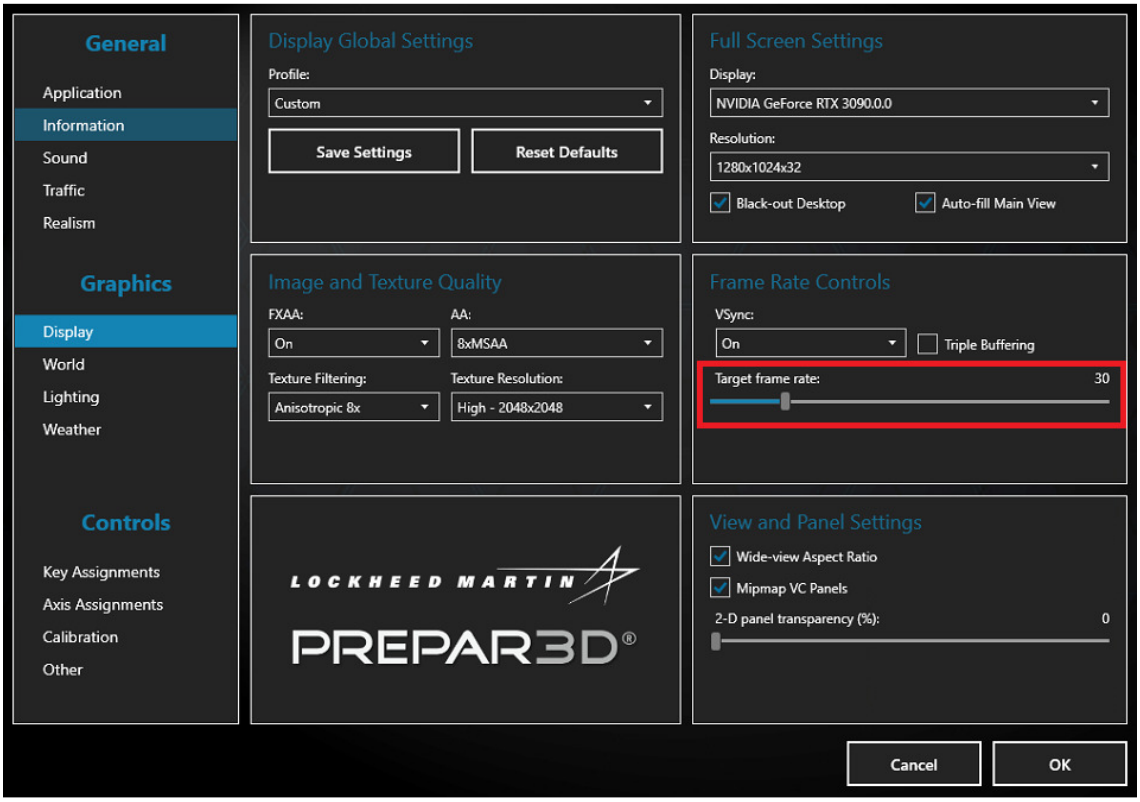
The VSync settings determine the maximum frame rate of the game and interpolated frames. The available options are ‘Application VSync settings”, “VSync On” and “VSync Off’.

When application-defined vSync is On or Frame Interpolation VSync is On, then the total frame rate (real + interpolated) is limited to the connected display refresh rate. In case of 60 Hz displays, the application will render 30 frames per second and the frame interpolation will insert additional 30 frames per second.
In cases when the VSync is off, then the application frame rate is unbounded. In such a case, frame interpolation will again generate a new frame for each application frame and by this doubling the total frame rate.
Frame interpolation can be temporarily enabled or disabled using the Enable toggle.

3 Examples frame interpolation configurations

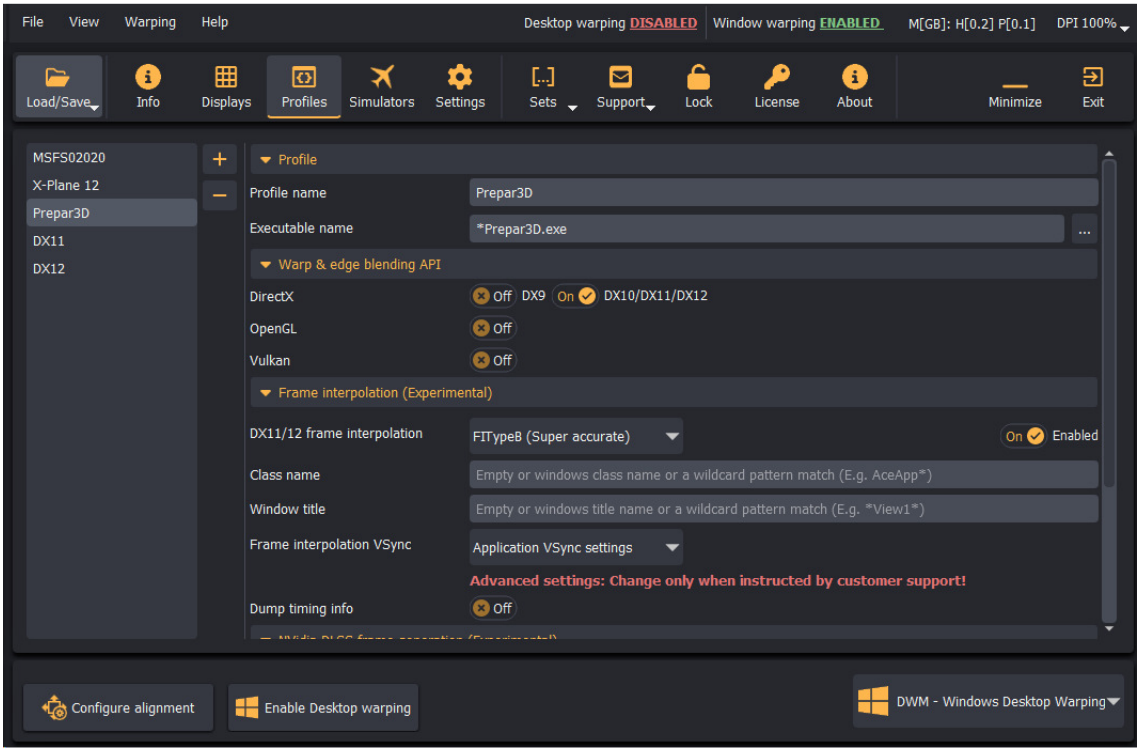
3.1 Prepar3D

Prepar3D is based on the DirectX 12 rendering pipeline. Frame interpolation can be enabled in both single view mode (2D Surround) and multi-view mode (ViewGroups).
Start P3D and configure the P3D target frame rate to 30 FPS and keep the VSync On.



Create a profile for P3D. In the Executable name field enter *Prepar3D.exe (do not forget the *). This will ensure that Prepar3D.exe from any system path will be used for frame interpolation.

Leave the VSync to “Application defined VSync” and Enabled to True. Select one of the available frame interpolations FITypeA or FITypeB. Start with FITypeA (fastest) and if some GUI artifacts are observed, try FITypeB.



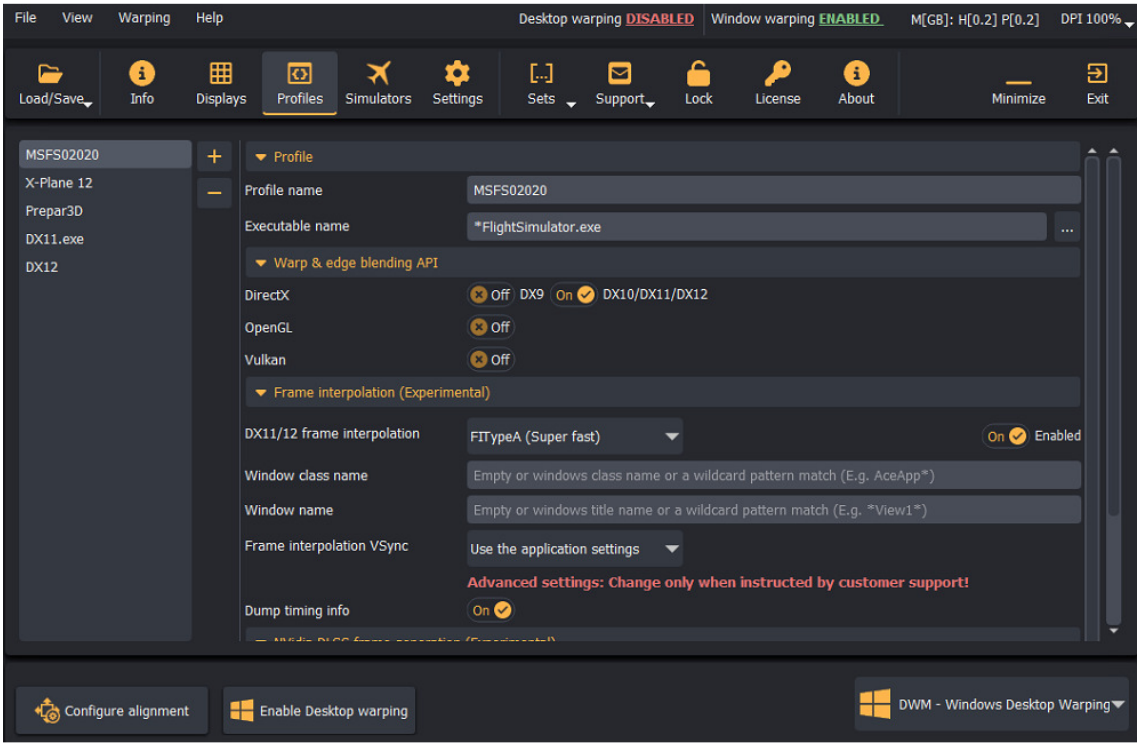
After Prepar3D is started it will report the game FPS of max 30 frames per second. However, output sent to the GPU (shown by Display PRO frame timing graph) is doubled to max 60 FPS.

3.2 Microsoft Flight Simulator MSFS2020

Microsoft Flight Simulator is based on the DirectX 11 or DirectX 12 (experimental) rendering pipeline. Frame interpolation can be enabled for both DirectX 11 and 12 modes and in both single view mode (2D Surround) and multi-view mode ungrouped displays mode.
Start MSFS2020 and limit MSFS2020 frame generation to 50% of the supported display frame rate. Keep the VSync On.

Create a profile for MSFS. In the Executable name field enter *FlightSimulator.exe (do not forget the *). This will ensure that FlightSimulator.exe from any system path will be used for frame interpolation, for both standalone or Steam based installations.

Leave the VSync to “Application defined VSync” and Enabled to True. Select one of the available frame interpolations FITypeA or FITypeB. Start with FITypeA (fastest) and if some GUI artifacts are observed, try FITypeB.



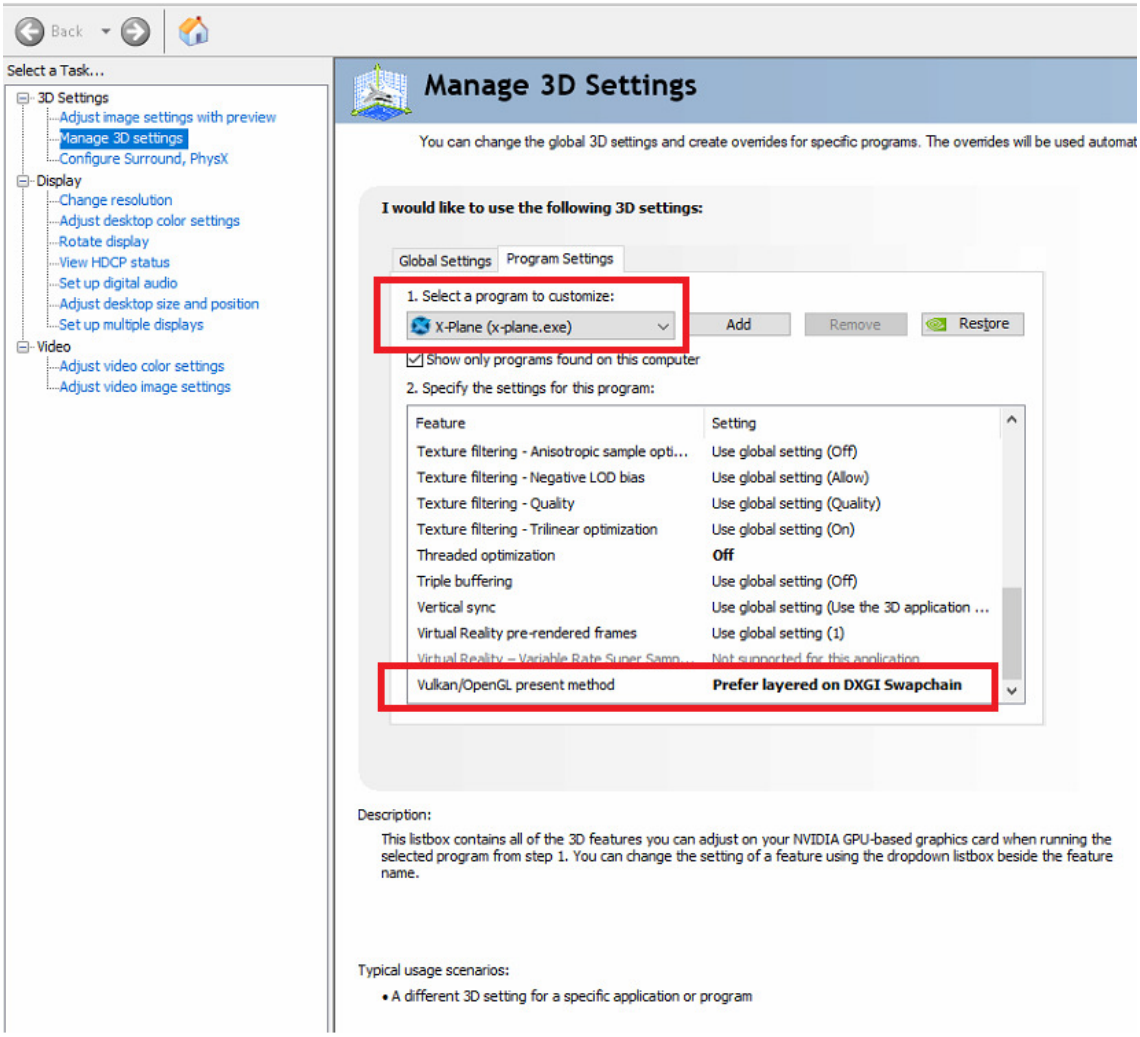
After MSFS2020 is started it will report the game FPS of max 30 frames per second. However, output sent to the GPU (shown by Display PRO frame timing graph) is doubled to max 60 FPS. The main thread can spend up to 30 ms per frame while the output is generated every 16 ms with one game frame and one interpolated frame.

3.3 X-Plane 11/12

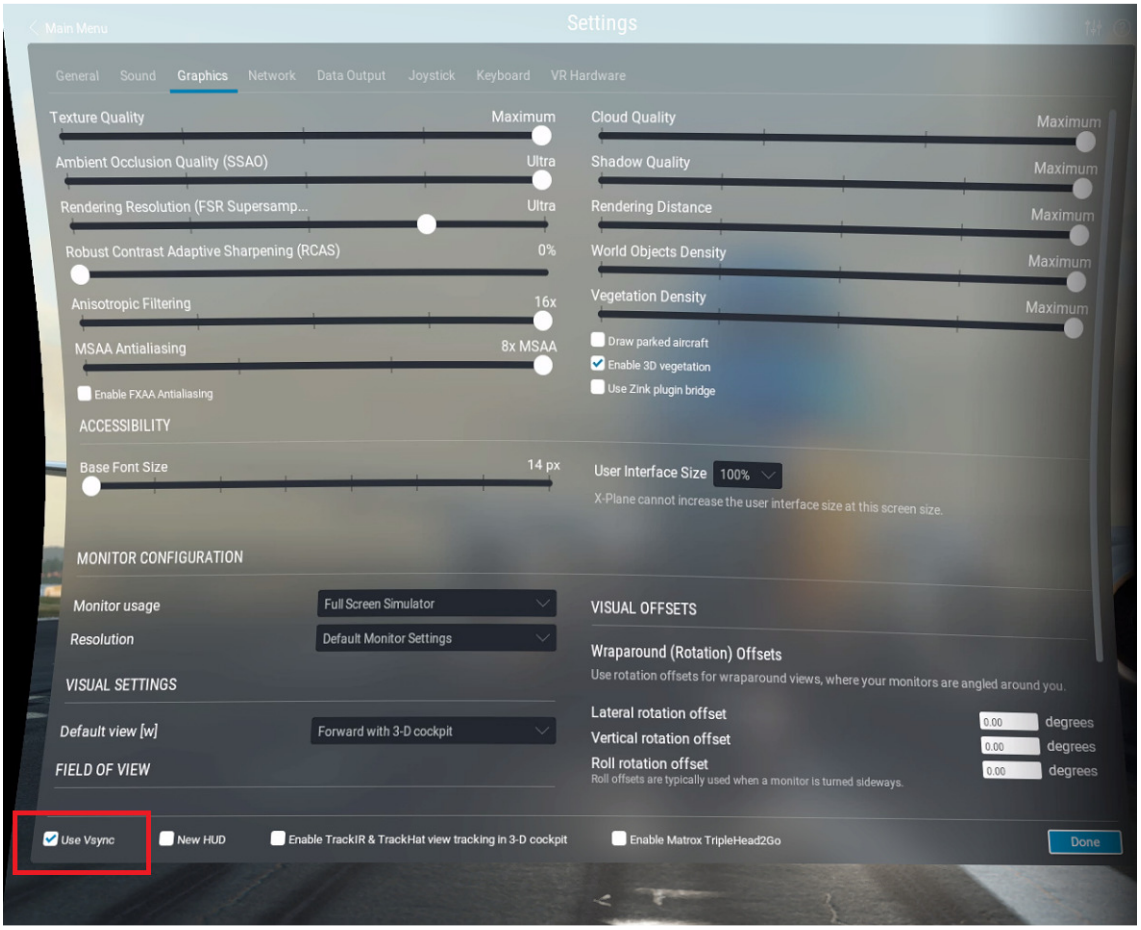
Both X-Plane 11 and X-Plane 12 are based on the Vulkan rendering pipeline. Optionally, X-Plane 11 can be configured to use the OpenGL rendering pipeline. Frame interpolation can only be applied on the DXGI based rendering pipeline used by DX11 and DX12.

Fortunately, the recent NVidia drivers and control panel allow configuring Vulkan/OpenGL based software to use the DXGI layer for presentation. Thai makes it possible to apply frame generation to all Vulkan and OpenGL based game engines.

Open the NVidia Control Panel, create a profile for XPlane and at the bottom of the settings select “Prefer layered on DXGI Swapchain” for Vulkan/OpenGL present method. Apply the settings and exit the NVidia Control Panel.



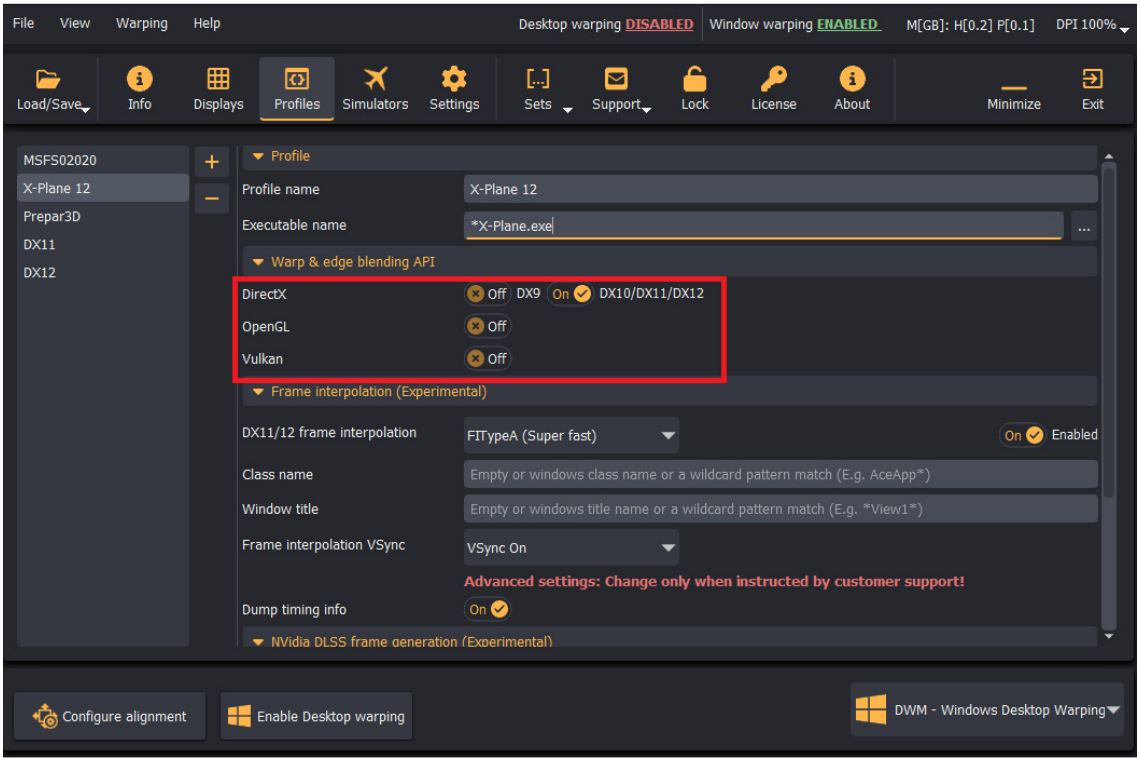
Start X-Plane and configure the graphics settings. Keep the Use VSync option On.



Create a profile for X-Plane. In the Executable name field enter *X-Plane.exe (do not forget the *). This will ensure that X-Plane.exe from any system path will be used for frame interpolation.

Leave the VSync to “Application defined VSync” and Enabled to True. Select one of the available frame interpolations FITypeA or FITypeB. Start with FITypeA (fastest) and if some GUI artifacts are observed, try FITypeB.

Important: In the profile Warp and edge blend API make sure that Vulkan and OpenGL are disabled and only DirecX10/11/12 is enabled. This will ensure that DXGI will be used for presentation and frame interpolation.



After X-Plane is started it will report the game FPS of max 30 frames per second. However, output sent to the GPU (shown by Display PRO frame timing graph) is doubled to max 60 FPS.

f-act f-sim frame cpu gpu grnd flit
29.934 39.681 0.0334 0.0254 0.0087 1.0000 1.0000
/sec /sec time time time ratio ratio
real totl missn timer zulu local hobbs
80.323 26.653 0.0000 12.447 13.447 0.2697
time time time time time time time
Frame timing
100
80
60
40
20
0
FPS(I) 60.0
FPS(R) 30.0
300
250
200
150
100
50
0
Total[us] 13.3
Warp[us] 2.4
Prep[us] 9.4