

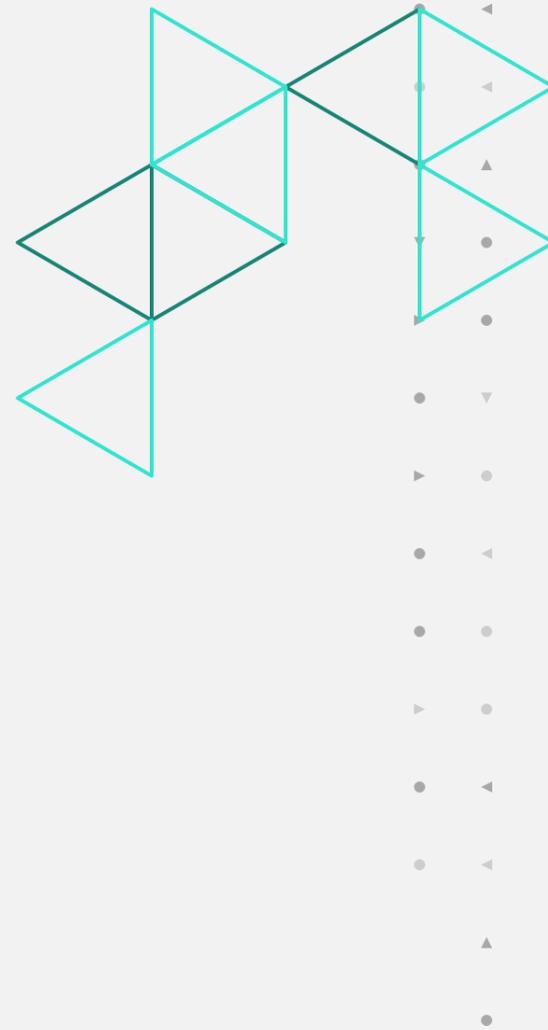


Microsoft Game Stack Live



Accelerating DirectX Innovation

Jacques van Rhyn

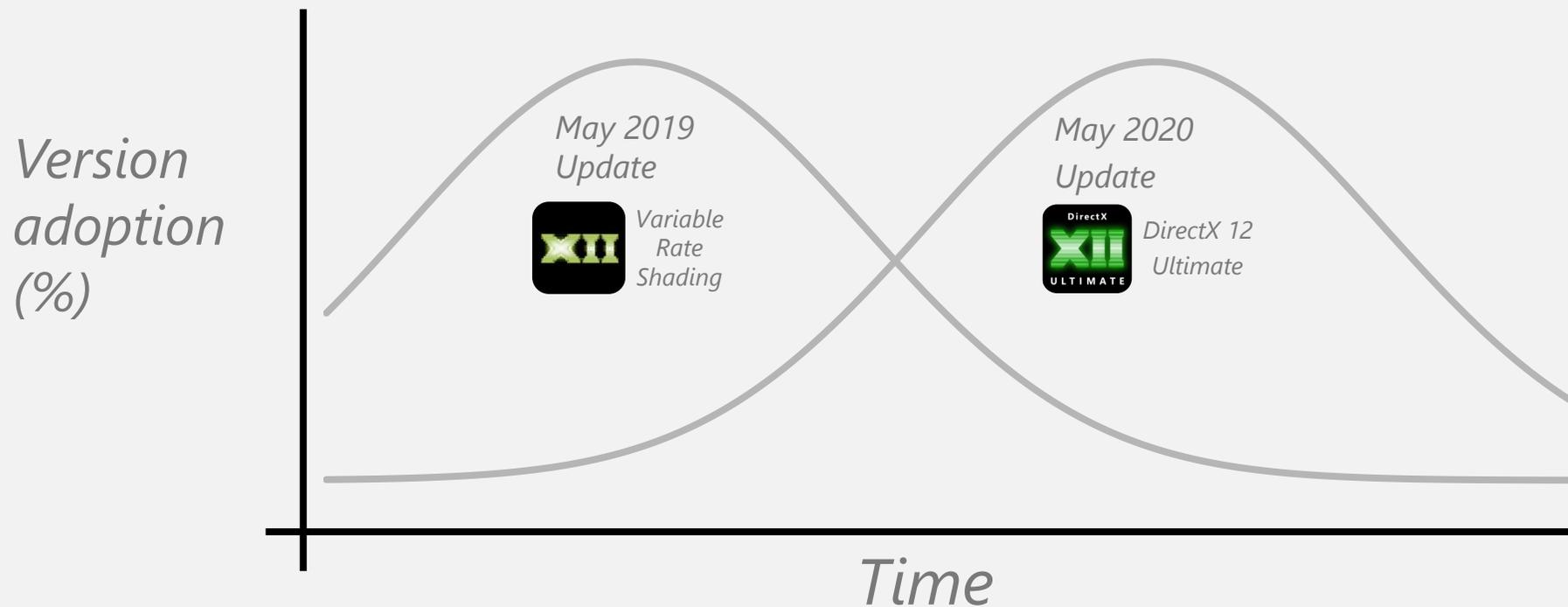


Windows 10 and DirectX 12



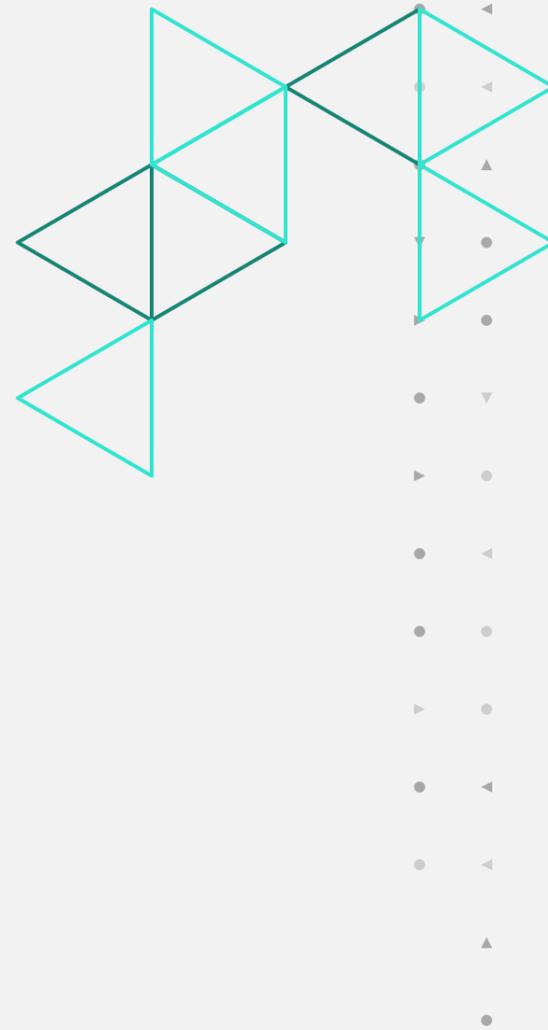
Windows 10 and DirectX 12

Tying DirectX 12 updates to Windows 10 releases means the default experience of devs and gamers gets gradually upgraded



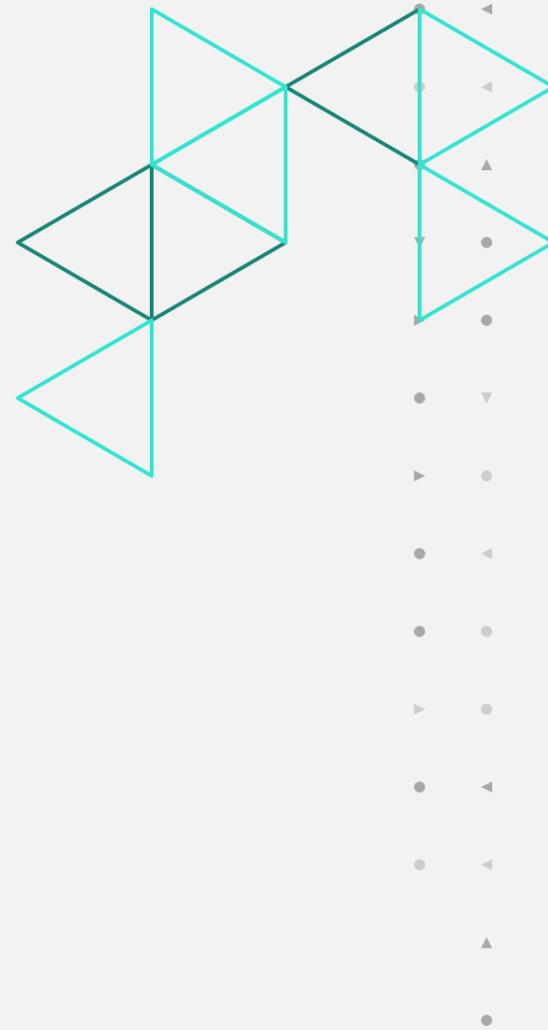
This model works, but we've heard one consistent piece of feedback loud and clear:

Game developers love our new features, but wish there was a way to light them up faster on their entire install base



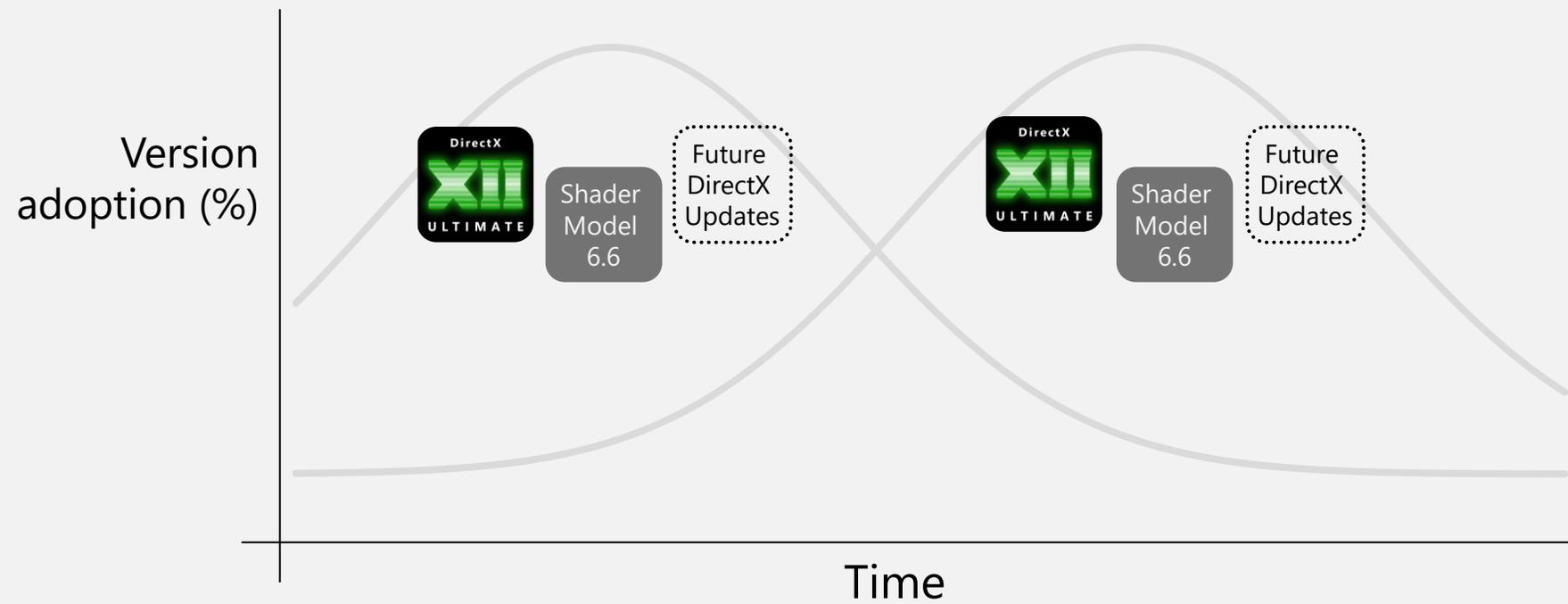
Announcing...

The DirectX 12 Agility SDK



Wide reach for all our features

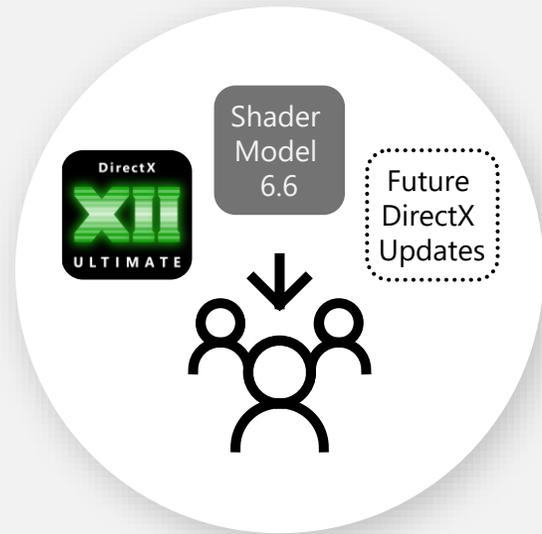
With the Agility SDK, all DirectX 12 features, new and old, run on a massive install base



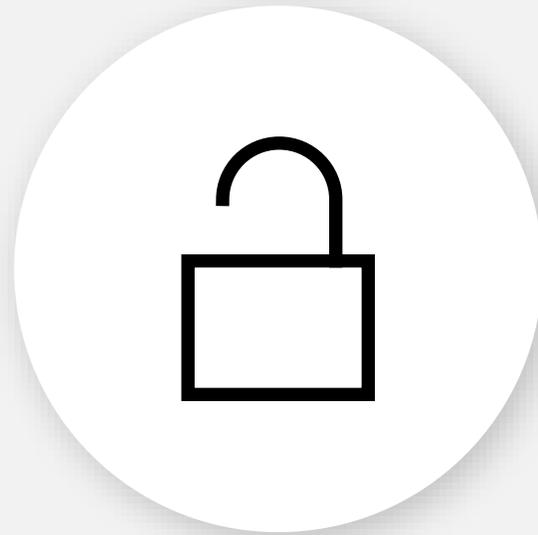
What does it mean for devs and gamers?



What does the Agility SDK mean for developers?



Runs on
most of the 1B+
Windows 10 devices



Easy to adopt



Bugfixes you
can count on



No interruptions
to existing games

Agility SDKs

Devs can get started developing with NuGet today!

Go to: aka.ms/directx12agility



DirectX 12 Agility SDK Downloads

Welcome to the DirectX 12 Agility SDK landing page, where you'll find links to Agility SDKs and other helpful info.

Today, our table of downloads has one link to our first SDK; we'll update it as we release subsequent SDKs.

Check out our [Announcement blog](#) and [GameStack Live talk](#).

For information on getting started and where to find drivers, please see our [Getting Started Guide](#).

Link to SDK	D3D12SDKVersion	Net-new features in this SDK
SDK 1.4	4	DirectX 12 Ultimate Shader Model 6.6

First Agility SDK

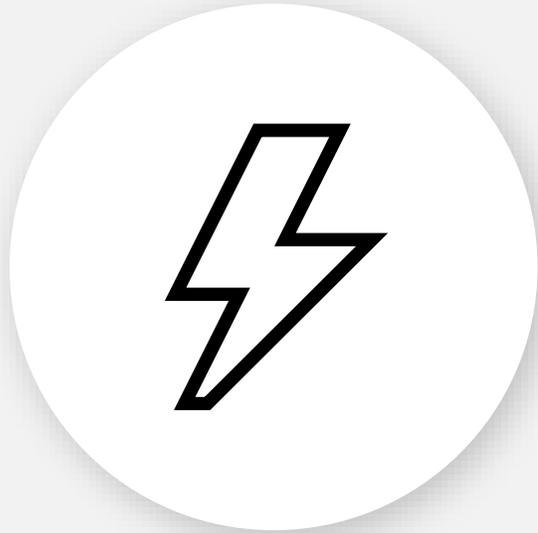


Unlocks DirectX 12
Ultimate Feature Set

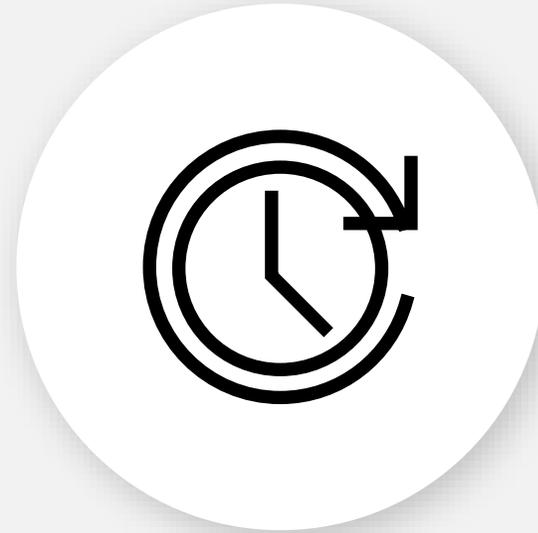


Supports a brand
new shader model

What's good for developers is good for gamers



Devs can bring new features to games faster



New hardware capabilities light up sooner

Industry support



"Our collaboration with Microsoft on the DirectX 12 Agility SDK enables us to easily implement forward-looking Unreal Engine features, and the new distribution model makes them quickly available to our developer and player communities."

**Nick Penwarden, Vice President,
Engineering, Epic Games**

"The DirectX 12 Agility SDK allows us at 343 to be confident that the latest innovation from the DirectX team runs for nearly all of our PC players"

**Tom Holmes, Engine Architect
at 343 Industries**

"The DirectX 12 Agility SDK will allow us to adopt the latest DirectX 12 features faster than ever. We are really excited about what this change means for the future of Forza Tech at Turn 10 and Playground Games"

**Chris Tector, Studio Software
Architect at Turn 10 Studios**

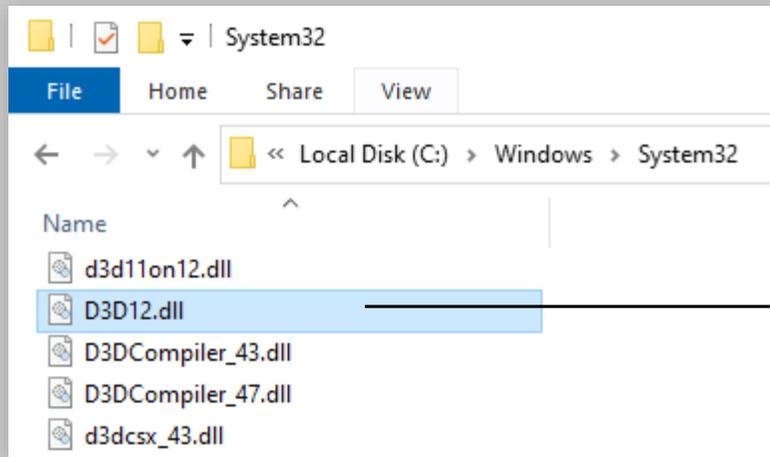
How do DirectX 12 Agility SDKs work?



Agility SDK relies on an OS change

Games rely on D3D12.dll in /System32

We needed to ensure existing games can carry on as usual, while also allowing newer D3D features to get loaded



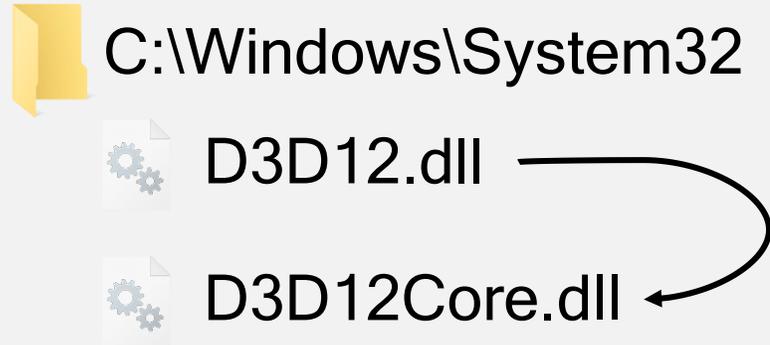
 D3D12.dll

now just a thin loader

 D3D12Core.dll

now the bulk of the old d3d12.dll

How does the new d3d12 loader work?

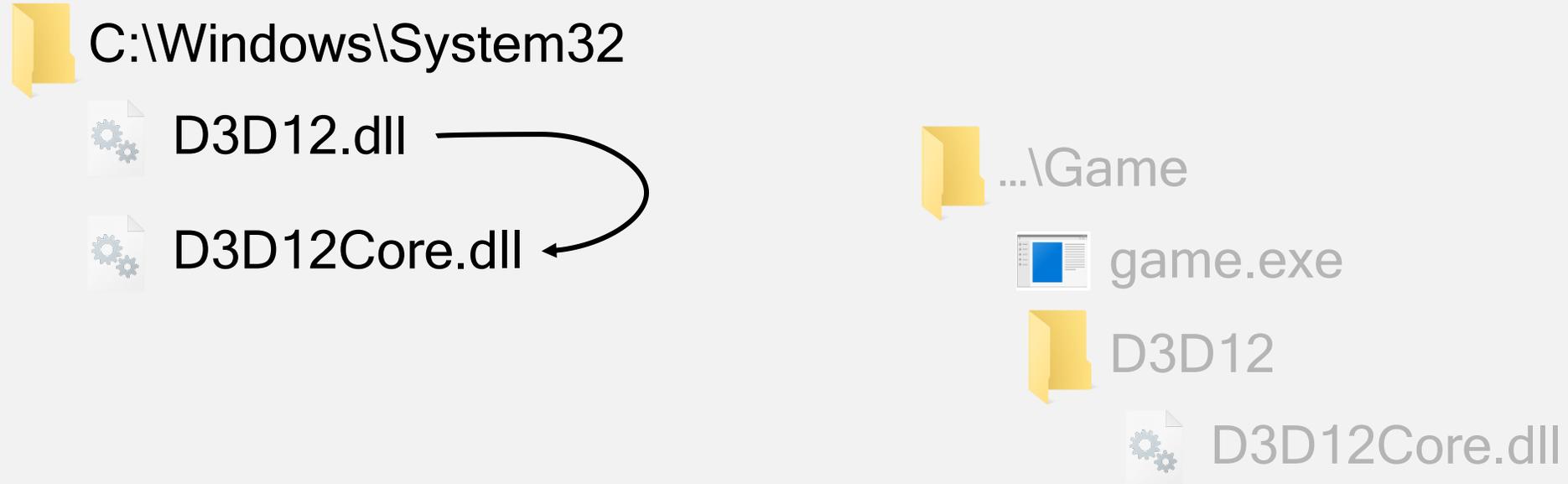


How does the new d3d12 loader work?



It will load a game's D3D12Core.dll if it's newer than the one in `\System32`

How does the new d3d12 loader work on newer OSes?



It will load the `\System32 D3D12Core.dll` when it's newer than what a game shipped with



Getting started: setting up your machine

This section is a how-to for developers interested in using the Agility SDK

Feel free to ask questions in the chat or in discord.gg/directx

What do I need to start developing?



Right OS Version

1909 and more recent



Visual Studio

We recommend VS 2019



Drivers with support for the Agility SDK

See aka.ms/directx12agility for where to find them



The latest shader compiler (depends on the feature)

See <https://aka.ms/hls1>



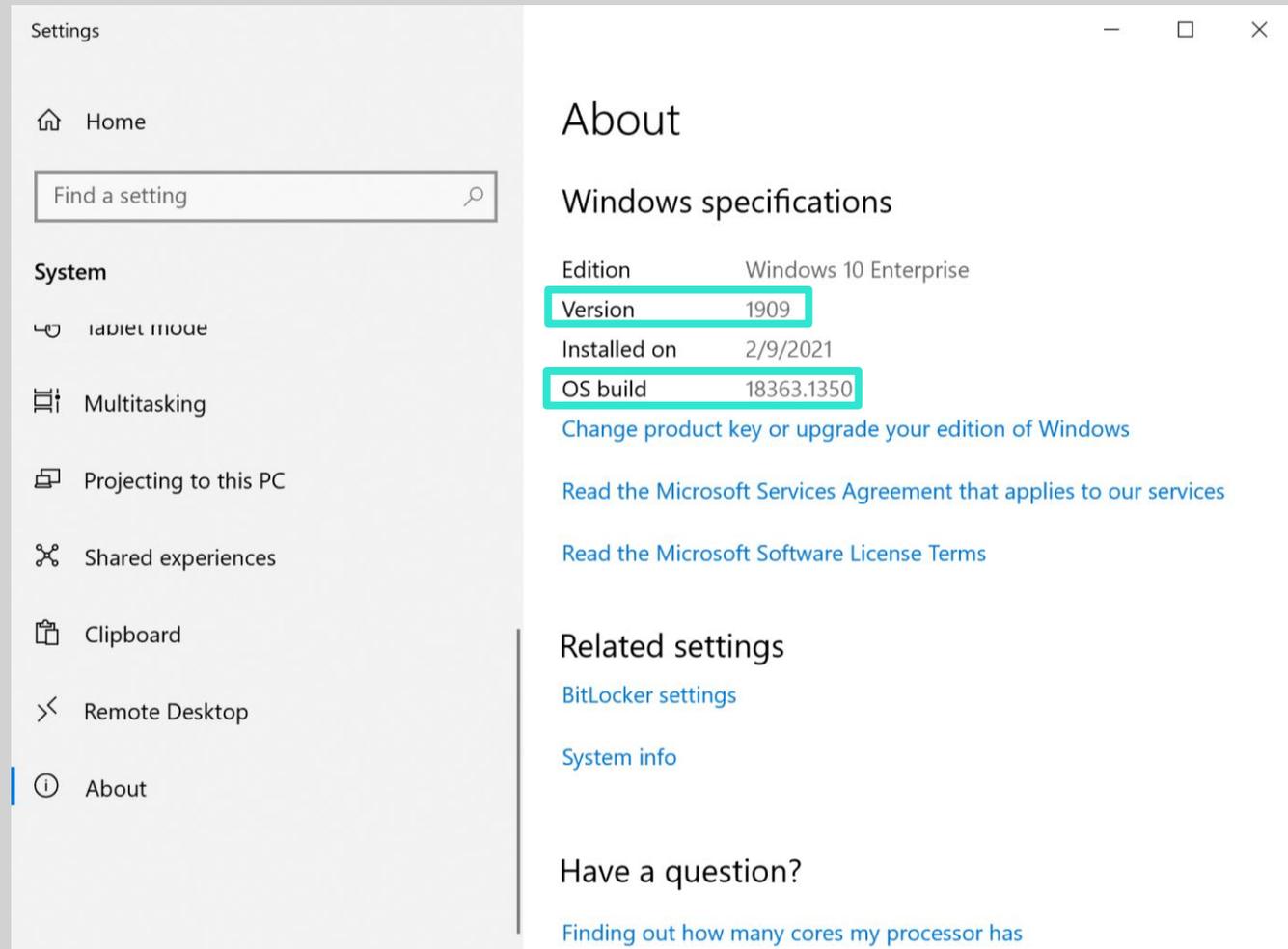
PIX

See <https://aka.ms/pixonwindows>



Right OS Version

Go to **Settings > About**
to double check you're on
OS version **1909** or above



Visual Studio 2019

We recommend Visual Studio 2019, but the 2017 edition works as well

Visual Studio Installer

Installed

Available



Visual Studio Community 2019

16.8.4

Powerful IDE, free for students, open-source contributors, and individuals

[Release notes](#)

Modify

Launch

More ▾

Drivers with Agility SDK support

You'll also need drivers with support for the Agility ADK

See aka.ms/directx12agility

Shader compiler

The Windows 10 SDK comes with a copy of the DirectX Shader Compiler

For a more recent copy, go to aka.ms/hlsl

PIX

PIX, our Graphics debugger, has native support for the Agility ADK

For more information go to aka.ms/pixonwindows



Getting started: using the Agility SDK

Once you've set up your dev machine, here's what's next:

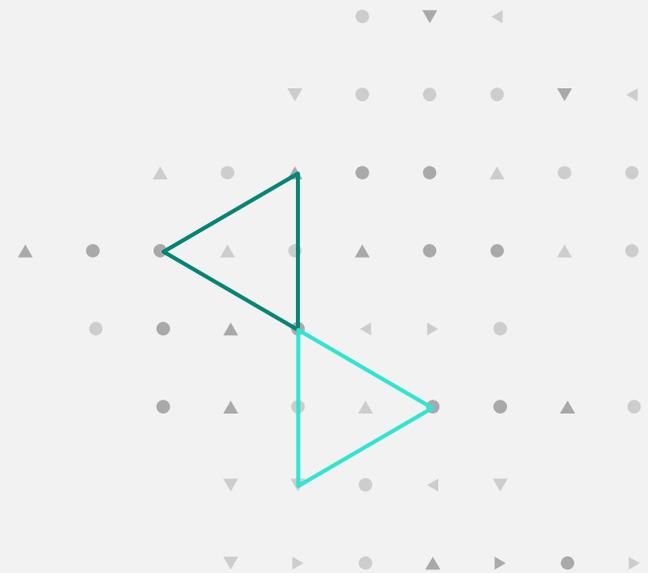


What's in each Agility SDK?

D3D12Core.dll

And all the components needed to build against it

- Headers (D3D12.h, D3D12SDKLayers.h, D3D12Video.h, etc.)
- The Debug Layer (D3D12SDKLayers.dll)
- D3DConfig.exe



Requesting a newer D3D12Core.dll

Games set required parameters by exporting constant data via well-known symbols:

D3D12SDKVersion

D3D12SDKPath

Here we see an app exporting these constants via a .def file

config.def

```
EXPORTS
```

```
    D3D12SDKVersion DATA PRIVATE
```

```
    D3D12SDKPath DATA PRIVATE
```

Requesting a newer D3D12Core.dll

Games set required parameters by exporting constant data via well-known symbols:

D3D12SDKVersion

D3D12SDKPath

And declaring them in code

config.def

EXPORTS

D3D12SDKVersion DATA PRIVATE

D3D12SDKPath DATA PRIVATE

main.cpp

```
extern "C" extern const  
uint32_t D3D12SDKVersion = 4;
```

```
extern "C" extern LPCSTR  
D3D12SDKPath = u8".\\D3D12\\";
```

```
extern "C" extern const  
uint32_t D3D12SDKVersion = 4;
```

```
extern "C" extern LPCSTR  
D3D12SDKPath = u8".\\D3D12\\";
```

D3D12SDKVersion = 4 in the first SDK

*Today, this exceeds any of the
D3D12SDKVersions of any of inbox
/System32 D3D12Core.dlls in any retail OS*

Developers must include the D3D12Core.dll
they built their app with

And also specify the path to D3D12Core.dll
relative to the app .exe



Did everything work?

Querying CheckFeatureSupport still helps you ensure the features you're lighting up are available

main.cpp

```
D3D12_FEATURE_DATA_FEATURE_LEVELS cap{};
cap.NumFeatureLevels = 1;
D3D_FEATURE_LEVEL requested = D3D_FEATURE_LEVEL_12_2;
cap.pFeatureLevelsRequested = &requested;
if (SUCCEEDED(device->CheckFeatureSupport(
    D3D12_FEATURE_FEATURE_LEVELS,
    &cap,
    sizeof(cap)))
    && cap.MaxSupportedFeatureLevel == requested)
{
    // feature level is supported on the device
}
```

Call to action

Major developers are using the Agility SDK to light up the latest DirectX functionality

You can do the same today!

Further resources

Check out the DirectX devblogs for our announcement and getting started guide

<https://devblogs.microsoft.com/directx/>

You can also find Agility SDKs and useful links at

<https://aka.ms/directx12agility>

DirectX Discord: <http://discord.gg/directx>



Happy coding!



FAQ



FAQ

Q: Will the redist work on Win7?

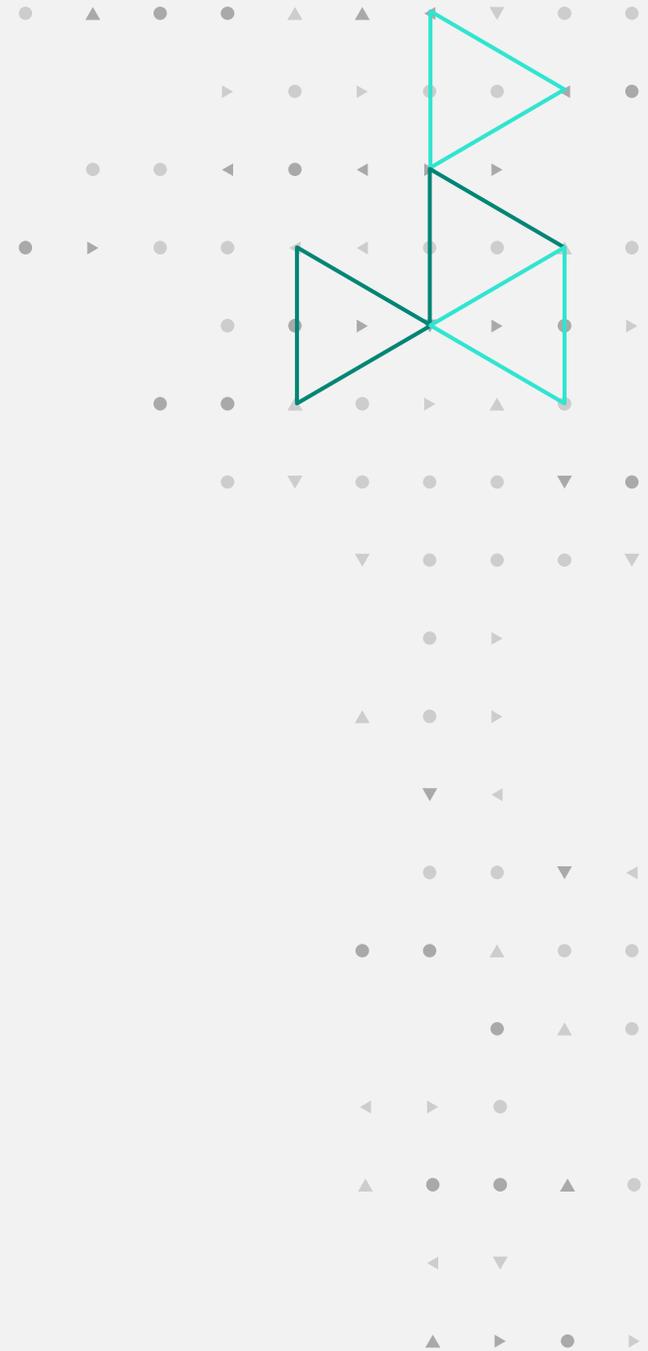
A: No, see 12on7

Q: What about developers who don't want or need the latest graphics features?

A: No action required

Q: Do gamers have to do anything?

A: No action required beyond installing drivers



Big opens



Thank you!

Come to our sessions, ask questions there...
You can always reach us on Discord...





Setup

1) OS

- Developing games that use the redist can be done on any Win10 OS **version 1903+**

2) Visual Studio

- Install **Visual Studio 2019** if you don't have it already
- Install the **Windows 10 SDK** from <https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/>)

3) Download and install drivers

- Go to aka.ms/___ for the latest on where to get drivers with redist support

4) PIX

- Go to <https://devblogs.microsoft.com/pix/download/> for the latest version of PIX

Motivation (points to hit, not final)

Light up new features on more sockets sooner, instead of asking devs to wait for most recent OS to reach enough gamers

Move to a much more agile model, getting bugfixes and tweaks to devs in the next redist SDK (in days/weeks) vs in the next OS (6-12 months)

Keep PC and Console DX12 stories in sync, since we can light up new features on PC and Console at similar times

Address a key developer pain point, by removing Windows Insider Program build requirement for new features

What does the redist mean for developers?

Tie back in with motivation. Other things to hit:

- **Broad gamer coverage**

- New features are available on every OS 1903+ as soon as they are released

- PC and Console stories stay more in sync as console evolves

- **More agile development**

- Bugfixes ship with new redist SDKs vs in the next OS/servicing patch

- Lumberyard no longer needs to use WIP builds to adopt bleeding-edge features

- **Low adoption cost**

- Very cheap for devs to adopt the redist

What does the redist mean for developers?

Large install base on day one

We've brought support for the redist to every OS 1903 and more recent

This guarantees the majority of Win10 machines can run games that use the very latest D3D12 features

Low cost of adoption

Redist SDKs are easy to integrate into existing engines by design

Only price of admission is that redist bits haven't gone through the full gamut of validation that goes with each OS

No interruptions to existing games

The redist model is purely opt-in

Devs who don't need the latest HW features can carry on like normal

Enter a more agile world

The DirectX team can now get bugfixes to devs via the redist

Undocking means more agility

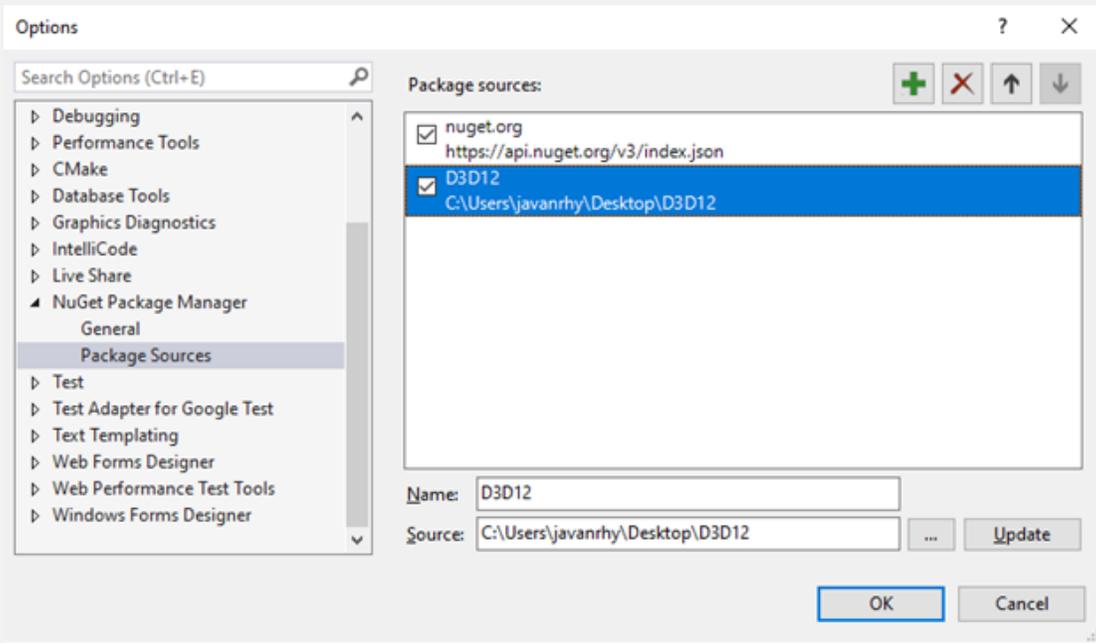
Bugfixes and tweaks can now get to devs in days vs asking them to wait for an OS to saturate

Microsoft's platform advantage continues to grow

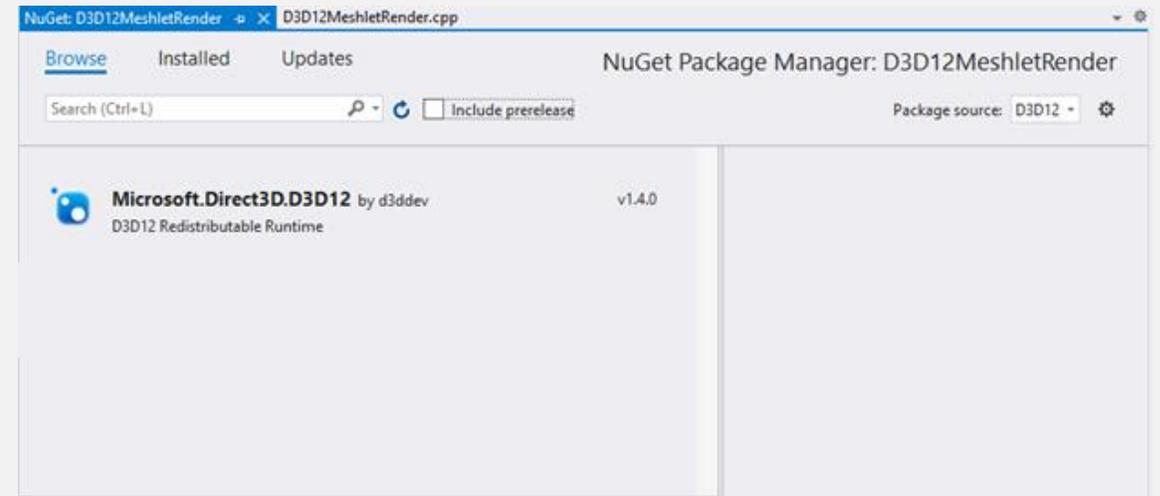
The DirectX team has the flexibility to ship whenever we want;

We can light up new features on PC and Console at the same time, making life easier for crossplat developers

Download and install the D3D12 redistributable SDK of your choice (Jacques to add video)



In Visual Studio go to Tools > Nuget Package Manager > Package Manager Settings > PackageSources. Add the package as a source



Right click on the solution > Manage Nuget Packages
Select the D3D12 redistrib SDK and Install