OpenXR 1.1 Launch
Empowering Portable Immersive Experiences
OpenXR Cross-Platform Portability

Before OpenXR: Applications and engines needed separate proprietary code for each device on the market.

Applications and engines can portably access any OpenXR-conformant hardware

OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.
## OpenXR Device Support

<table>
<thead>
<tr>
<th>Device Manufacturer</th>
<th>Product Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft HoloLens</td>
<td>Hand and eye tracking extensions</td>
</tr>
<tr>
<td>Meta Rift S, Quest 3, Quest 2 and Quest Pro</td>
<td>Meta Deprecated own API for OpenXR</td>
</tr>
<tr>
<td>HTC Vive Focus 3, Vive Cosmos, Vive XR Elite, Vive Wave Runtime</td>
<td></td>
</tr>
<tr>
<td>Valve</td>
<td>Valve Index, Valve Deprecated OpenVR APIs for OpenXR</td>
</tr>
<tr>
<td>Varjo</td>
<td>All Varjo Headsets are fully compliant XR-3, XR-4</td>
</tr>
<tr>
<td>Canon</td>
<td>MREAL X1</td>
</tr>
<tr>
<td>Magic Leap</td>
<td>Magic Leap 2</td>
</tr>
<tr>
<td>XREAL</td>
<td>XREAL Air 2, Air 2 Pro, Air 2 Ultra</td>
</tr>
<tr>
<td>Qualcomm Snapdragon Spaces XR Development Platform</td>
<td></td>
</tr>
<tr>
<td>Spatial Labs</td>
<td>Display Series</td>
</tr>
<tr>
<td>Neo 3 and Pico 4</td>
<td></td>
</tr>
<tr>
<td>Sony Spatial Reality Displays</td>
<td></td>
</tr>
</tbody>
</table>
Engines, Browsers, and Libraries with OpenXR

<table>
<thead>
<tr>
<th>Unreal Engine</th>
<th>Unity</th>
<th>Godot Game engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unreal has been providing support since 4.24. UE 5.0 supports OpenXR</td>
<td>Unity’s OpenXR plugin available since 2020 LTS</td>
<td>Godot provides OpenXR support since March 2023 (Core 4.0 Alpha 4)</td>
</tr>
<tr>
<td>Autodesk VRED Library</td>
<td>NVIDIA Omniverse and CloudXR Platforms</td>
<td>WebXR in Chrome, Edge, and Firefox uses OpenXR as the default backend</td>
</tr>
<tr>
<td>OpenXR supported since VRED 2023.4</td>
<td>Open-source OpenXR Implementation</td>
<td>Autodesk open-source mixed reality library for building HoloLens and VR applications</td>
</tr>
</tbody>
</table>

A lightweight XR Meta XR Simulator to Speed Unity OpenXR Development
OpenXR 1.1

Consolidates multiple extensions into OpenXR 1.1 core
Streamlined development and reduced fragmentation

OpenXR 1.1 Feature Enhancements
Additional functionality
Spec clarifications and improvements

Continue leveraging OpenXR’s Flexible Design
To foster innovation in developing extensions to explore new use cases

Drive for Immersive Experience Portability
Increased focus on integrating widely adopted extensions into core for cross-platform portability
The OpenXR Story So Far...

Empowering developers to create cross-platform, immersive spatial computing experiences

- **OpenXR Working Group Formed**
  - 2017
  - Vendor API fragmentation
  - Clear industry demand need for a cross-platform XR open standard

- **OpenXR 1.0 Released**
  - 2019
  - Establishing baseline functionality
  - OpenXR 1.0 provides baseline functionality and the foundation for experimentation with new functionality through extensions
  - Exploring new functionality e.g., body tracking and advanced spatial computing
  - OpenXR achieves wide industry adoption

- **OpenXR 1.1 Released**
  - 2024
  - Still discovering new use cases
  - Regular core spec updates to balance the need to ship new functionality AND consolidating proven technology to reduce fragmentation

This work is licensed under a Creative Commons Attribution 4.0 International License
OpenXR 1.1 Updates

• Key Extensions Promoted to Core
  - Local Floor Reference Space
    - Gravity-aligned world-locked origin for standing-scale content that can be recentered to the current user position at
      the press of a button without a calibration procedure - with an estimated floor height built in
  - Stereo with Foveated Rendering
    - Eye-tracked or fixed foveated rendering for XR headsets across multiple graphics rendering APIs
  - Grip Surface
    - Anchors visual content relative to the user's physical hand
    - Whether the hand position is tracked directly or inferred from a physical controller’s position and orientation

• Feature Enhancements
  - Interaction Profile improvements
  - Spec language cleanup and clarifications
Coming Soon...

- Extending hand tracking
  - To include full body tracking
- Enhanced handling of spatial entities
  - Standardized methods to interact with the user’s environment
  - Support for advanced spatial computing applications
- Expanded haptics support
  - Support immersive experiences through PCM, vibrotractiles, and transients
- Controller render models (gltF)
  - Dynamically highlight pressed buttons or show menus pointing to buttons
- Increased Accessibility
  - Input rebinding UI: input re-binding at runtime
  - Use any button/interaction mapping you wish
- Metal (Mac OS) Support
  - Provide swapchain images on Metal
OpenXR Development Resources & Tools

- **OpenXR SDK**
  - Headers, source code, and build scripts
  - [https://github.com/KhronosGroup/OpenXR-Docs](https://github.com/KhronosGroup/OpenXR-Docs)

- **Reference Pages** and **Reference Guide**
  - Developer documentation

- **OpenXR Tutorial**
  - For creating applications using Android, Linux, Windows

- **Conformance Test Suite**
  - Development version freely available in open source
  - Used to test for formal conformance

- **Support & Community Forums**
  - OpenXR on [Discord](https://discord.com)
  - [OpenXR Forum](https://khronos.figurate.com/forums) at Khronos
  - [OpenXR Issue Tracker](https://github.com/KhronosGroup/OpenXR) on GitHub
  - Developing OpenXR Resources? Let us know!
Get Involved!

Provide feedback and requirements on GitHub, Discord, or OpenXR Forums
Get questions answered and make suggestions for improvement!

Join Khronos and the OpenXR Working Group
https://www.khronos.org/openxr/
https://github.com/KhronosGroup/OpenXR-Docs
Khronos and W3C: Bringing XR to the Web

XR Applications and Engines use an API from both the 3D and XR Stacks

3D Stack
Driving GPUs to render scenes and augmentations

XR Stack
Handling XR Devices for creating UI

Engines
## OpenXR Games and Applications

<table>
<thead>
<tr>
<th>Blender</th>
<th>Adobe Substance 3D Modeller</th>
<th>Kitware’s Paraview</th>
<th>Meta Horizon Workrooms</th>
<th>OpenBrush</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses OpenXR for native scene inspection in VR</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for Desktop and Quest support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Blender</th>
<th>Adobe Substance 3D Modeller</th>
<th>Kitware’s Paraview</th>
<th>Meta Horizon Workrooms</th>
<th>OpenBrush</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses OpenXR for native scene inspection in VR</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for Desktop and Quest support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>War Thunder</th>
<th>Cubism</th>
<th>Vermillion</th>
<th>The Light Brigade</th>
<th>XPlane12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses OpenXR</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minecraft</th>
<th>Microsoft Flight Simulator</th>
<th>Zombie Land Headshot Fever</th>
<th>Phasmophobia</th>
<th>Beat Saber</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses OpenXR for desktop VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Uses OpenXR for VR support</td>
<td>Switched from OpenVR to OpenXR</td>
<td>Uses OpenXR on PC</td>
</tr>
</tbody>
</table>

| Supports over 27 devices thanks to OpenXR | Phasmophobia switched from OpenVR to OpenXR | Beat Saber on PC uses OpenXR | | |

This work is licensed under a Creative Commons Attribution 4.0 International License
Monado from Collabora

- Open source OpenXR Runtime and Framework
- Framework provides building blocks to simplify XR development
OpenXR SDK 1.1.36 - April 2024

- 13 new interaction profiles added to the core spec
- 5 new extensions rolled into core (available on all platforms)

- Added notes for application developers:
  - benefits of foveated rendering
  - which reference space to use
  - which pose identifier to use
  - unified extension error codes

- A number of clarifications for runtimes to provide a consistent user and developer experience across platforms
OpenXR CTS 1.1.36 - April 2024

- Enable running the CTS in 1.0 mode to ensure backcompatibility
- New interactive and automated tests:
  - local floor
  - palm pose
  - interaction profiles
  - foveated rendering
  - locate spaces