

USB Peripheral Support

The **Frame Platform** provides flexible peripheral support to meet diverse enterprise needs through both **browser-based** and **Frame App** deployment options.

For standard office peripherals like keyboards, mice, webcams, and microphones, users can access their devices directly through **browser-based Frame sessions**.

For organizations requiring support for specialized industry devices - common in education, financial services, healthcare, and professional design environments, **Frame App** offers enhanced USB support.

This dual approach ensures organizations can choose the deployment method that best matches their peripheral requirements while maintaining **security and performance**.

Considerations

Before implementing Frame Platform's peripheral support, it's important to assess your organization's specific needs. Consider the types of devices your users require, their work patterns and any industry-specific requirements.

This discovery process will help determine whether browser-based Frame or Frame App is the most suitable deployment option for your environment.

Key Questions

- Are your users primarily using standard keyboards, mice, webcams, and microphones?
- Do your users need to connect specialized USB devices (healthcare peripherals, security dongles, CAD controllers)?
- Will users need access to smart card readers or document scanners?
- Is basic browser-based peripheral support sufficient for your workflows?
- Do you require enhanced USB device support beyond standard input/output devices?

Supported Peripheral Use Cases

User Type	Required Peripherals	Recommended Approach	Additional Details
-----------	----------------------	----------------------	--------------------

Standard Office User

- Standard keyboards and mice
- Webcams
- Headsets/microphones
- Basic document scanners

Frame in browser

- Provides standard peripheral support through browser APIs
- Ideal for common office peripherals
- Simplest deployment option
- No additional software required

CAD/AEC Professional

- 3D mice/Space mice
- Drawing tablets
- Hardware security keys/dongles
- Professional-grade input devices

Frame App 7 with Generic USB enabled

- Required for specialized input devices and security dongles
- Ensures compatibility with professional design peripherals
- Supports high-precision input requirements
- Enables advanced device features

Financial Services Professional

- Smart card readers
- Security tokens
- Signature pads
- Multi-card readers
- Document scanners

Frame App 7 with Generic USB enabled

- Required for secure authentication devices
- Ensures compliance with financial security requirements
- Supports specialized financial hardware
- Enables secure peripheral access

After determining your peripheral requirements and recommended deployment approach, it's important to understand the technical capabilities and limitations of each solution.

Browser-based Frame sessions utilize modern web standards, such as WebHID APIs and MediaDevices APIs, to enable peripheral support directly through the browser. In contrast, **Frame App 7** provides enhanced USB device compatibility through its client architecture.

Understanding Peripheral Support

This section provides a high-level look at how end-user device peripherals communicate with the Frame session when accessing via **Frame App** or a **compatible browser**.

We'll review how devices are supported and what to keep in mind when using each environment.

Frame in a Browser

Frame utilizes several standard browser APIs for peripheral support:

- **WebHID API:** Enables direct communication with Human Interface Devices (HID) such as keyboards, mice, and game controllers.
- **MediaDevices API:** Manages audio and video peripherals like webcams, microphones, and speakers.
- **WebUSB API:** Provides access to certain USB devices in Chromium-based browsers on macOS and Linux.

Note:

Some peripheral features require enabling experimental web platform features in certain browsers. This is particularly relevant for USB and HID device support in Chrome-based browsers.

Instructions will be provided later in this document.

Operating System	Chrome/Edge	Firefox	Safari
Windows	Full peripheral support	Basic peripheral support*	Not supported
macOS	Full peripheral support	Basic peripheral support*	Basic peripheral support*
Linux	Full peripheral support	Basic peripheral support*	Not supported
iOS/Android	Limited support**	Limited support**	Limited support**

* Basic support includes standard audio/video devices only.

** Mobile devices support touch input and basic audio/video features.

Frame App 7

The communication flow between a user's local device and the Frame Workload VM begins with the **Frame App**, where a device type check determines compatibility.

- **WebHID-compatible devices** are routed through **WebHID** or **MediaDevices APIs**.
- Other devices undergo a **WebUSB compatibility check**.
- If compatible, the device connects via **WebUSB API**.
- If incompatible, drivers are replaced with WinUSB.sys on Windows or libUSB on macOS/Linux before routing to WebUSB.

Once connected, the physical USB device communicates with the **VirtualUSB Driver** on the Frame Workload VM, which integrates with the Frame Session to enable optimal peripheral functionality.

This process ensures support for a wide range of peripherals across different devices and operating systems.

A green screen with black textAI-generated content may be incorrect.

Browser-Based Frame Compatibility

This section provides administrators with a high-level overview of how Frame supports peripherals and workflows in browser-based sessions. By leveraging technologies such as WebUSB and WebHID, Frame enables functionality for printers, scanners, and USB HID devices, alongside secure, browser-native file transfer capabilities. It also outlines limitations tied to operating systems and browsers, offering guidance to help organizations optimize peripheral compatibility for their users.

NOTE: Compatibility for peripherals and features in Frame can vary depending on the end user's device, operating system, and the browser used to access Frame. While we strive to provide broad support across platforms, some functionality may be limited or optimized for specific environments.

For assistance in ensuring the best experience for your users or to address specific compatibility questions, we encourage you to work closely with your Dizzion Account Team. They can help you evaluate your needs and recommend the best configurations for your organization.

Printers

Frame supports two options for printing from a session:

A screenshot of a computerAI-generated content may be incorrect.

Printing within a Frame session

1. **Frame Printer:** When an end user initiates a print from any application, the default option will be the Frame Printer. Frame Printer uses WebUSB technology to enable direct printing from the Frame session through your local browser to your USB-connected printer. The print job is processed locally on your device, similar to a "Print to PDF" workflow. The Frame Printer is automatically installed and configured on all Frame Workload VMs, requiring no additional setup on the VM side. Users simply need a USB or network-connected printer on their local device to utilize this functionality.

2. **Network Printing:** Network printing directly from a VM is possible for accounts leveraging Domain Joined Instances where network printer policies are set by the Windows system Admin through their domain controller. Organizations interested in leveraging this option can discuss with their Dizzion Account Team.

Mass Storage and File Transfer

Frame handles file transfers through native browser-based upload and download capabilities rather than direct mass storage access. For security and data integrity, Frame does not support local disk passthrough or direct access to external storage devices (such as USB drives or external hard drives).

A screenshot of a computerAI-generated content may be incorrect.

Users can transfer files efficiently using intuitive browser-based workflows, including drag-and-drop functionality between their local device and Frame sessions. This approach ensures consistent file handling while maintaining data integrity across sessions.

NOTE: Frame does not support operations requiring **asynchronous file transfers**. All file operations are handled through direct browser-based upload and download processes.

HID Device Support

Frame supports **USB Human Interface Devices (HID)**, enabling seamless integration for devices such as keyboards, mice, game controllers, and specialty input devices.

When using **Frame App**, USB HID devices (mouse, keyboard, touchpad, headset, speakers) are supported without requiring users to manually select them. Specialized HID devices (joysticks, gamepads, fingerprint scanners, driving and flight simulators) may require manual selection from the menu to enable access and use within the Frame session.

To jump ahead to the high-level compatibility chart, [click here](#). (link missing)

Frame App Compatibility

Frame App provides robust support for a variety of peripheral devices, offering administrators and end users a reliable and seamless experience. Designed to accommodate the needs of modern enterprise environments, Frame App ensures that USB peripherals integrate effortlessly into virtual sessions with minimal setup. The peripherals listed below require Frame App 7 to connect to a Frame session from the user's local device.

WIA Scanners

Frame enables seamless integration with document WIA scanners through USB redirection, offering a smooth scanning experience within virtual desktop sessions. Users can connect a scanner to their local device, where the operating system typically detects and installs the necessary drivers automatically. Once in a Frame session, the scanner can be virtually attached using the "Add USB Device" feature in Frame Terminal.

After the scanner is recognized within the virtual session, applications running in the Frame environment can access the scanner as though it were directly connected. This allows users to perform tasks like document scanning without the need for complex configurations or manual driver installations. Supported scanners, such as Fujitsu/Ricoh models, have been successfully used in scenarios involving Frame App and Windows-based sessions.

For tailored recommendations or assistance with scanner compatibility, we encourage you to collaborate with your Dizzion Support Team.

NOTE: Scanners that rely on the TWAIN standard are not supported.

Revision #4

Created 27 October 2025 11:20:04 by Stefan Gajic

Updated 15 January 2026 15:11:43 by Dominik Conrad