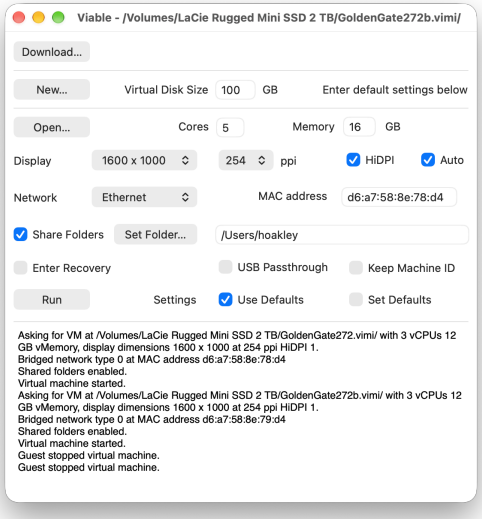
Start

Viable uses lightweight virtualisation in macOS Monterey 12.4 and later to create and run macOS virtual machines (VMs) on Apple silicon Macs. It can run one or two VMs simultaneously at speeds at or close to those of macOS running native. Its VMs can also be run from a double-click using Vimy and enhanced using ARD.



Before building your first VM, you should:

- *either* create a new folder to contain your VMs, and add that to your Mac's backup exclusion list,
- *or* create a new volume and add that to your Mac's backup exclusion list (preferred).

Storing your VMs in a folder saves them from being backed up every time they change. However, they'll still be included in any snapshots made of that volume. The second option ensures that they don't take any space in snapshots either.

➜ Workflow summary

To quit Viable, close its main window or use the **Quit** command.

➜ Controls
➜ Network settings
➜ Machine ID
➜ Download
➜ Display settings
➜ VM problems
➜ Create new VM
➜ Shared folders
➜ Settings file
➜ Open & run VM
➜ Features
➜ Change list
➜ Main settings
➜ Enhance with ARD

➜ Start

Workflow summary

Download the IPSW for your chosen version of macOS; for the current version click on **Download**. ➜ Download

Create a new VM

Set **Virtual Disk Size**, and the default settings below that. Then click on **New** or *Command-N,* select the IPSW to be used, and save the new VM. Once complete, click on **Run** or *Command-R* to run and configure your new VM. ➜ Create new VM

Open an existing VM

Click on **Open** or *Command-O,* then either √ **Use Defaults** to run it using its default settings, or *untick* that and adjust other settings. When happy, click on **Run** or *Command-R.* ➜ Open & run VM

Double-click a VM to run it

Download and install Vimy: it's what it does.

Quit a running VM

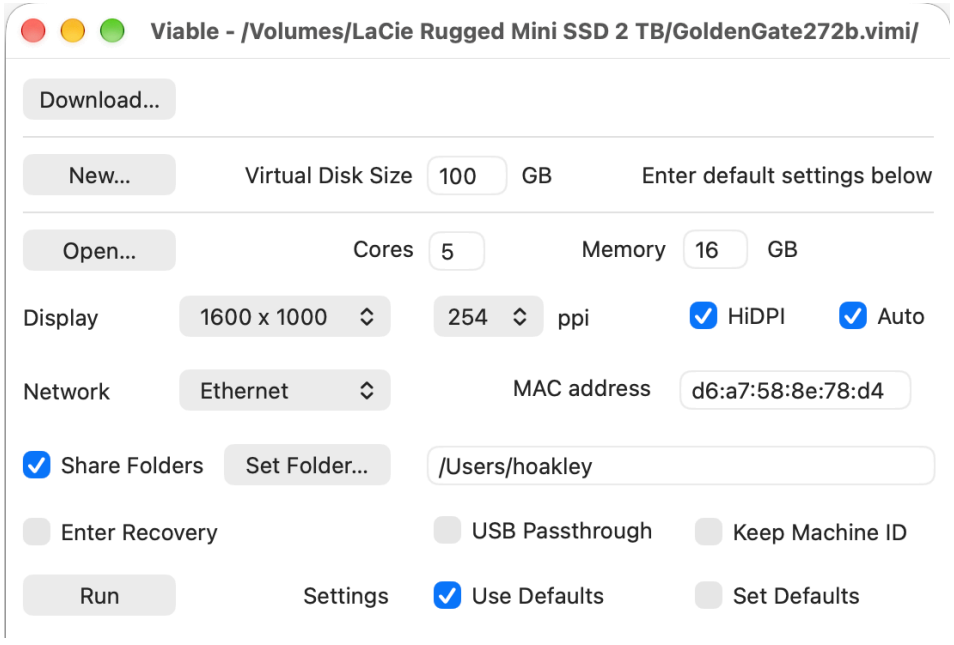
Use the **Shut Down** menu command in the VM.

Force quit a running VM

Quit Viable.

➜ Start

Controls



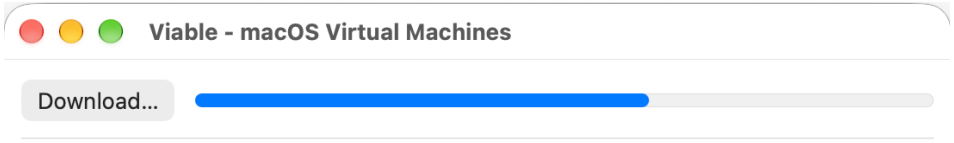
Viable's main window provides four buttons to perform its functions, each of which is also available from its **File** menu:

- **Download**: downloads the latest release version of macOS as an IPSW image file, and saves that to the location you choose. This can be on a different volume or disk from the VM you'll create, and will be left there during VM creation.
- **New**: takes an IPSW image you have already downloaded from Apple and builds it into a fresh VM ready to personalise and configure on its first run using the **Run** button. The size of the VM is set as the **Virtual Disk Size**, and creates a default settings file using values in the controls below.
- **Open**: opens an existing VM ready to run with the settings in the controls.
- **Run**: runs the VM you have either just created using the **New** button, or have opened using the **Open** button, using the settings in this window, or those already saved in the VM bundle, depending on whether the **Use Defaults** box is ticked. The first time you start a VM that has been freshly installed, macOS takes you through initial configuration, including personalisation, just as when setting up a new Mac.

➜ Workflows
➜ Download
➜ Create new VM
➜ Open & run VM
➜ Main settings
➜ Network settings
➜ Display settings
➜ Shared folders
➜ Settings file

➜ Start

Download



macOS Virtual Machines are built not from macOS Installer apps, but from special IPSW image files containing all the firmware and cryptexes required, in addition to the whole of that version of macOS. Those can only be obtained from Apple, and the **Download** button provides convenient access to the IPSW file for the latest release version of macOS, direct from Apple's servers.

Click the **Download** button and you will be prompted to provide a name for the file and its location. This can be anywhere your Mac has access to, and need not be where you keep your VMs. The bar will then show the progress of the download until it's complete.

Currently, IPSW files are about 26 GB in size. As with other files, they are downloaded into a temporary file in your Mac's Data volume, and moved from there to the chosen destination as soon as the download is complete. If there's a snapshot made of the Data volume during downloading, don't be surprised if it contains that large temporary file until that snapshot is removed.

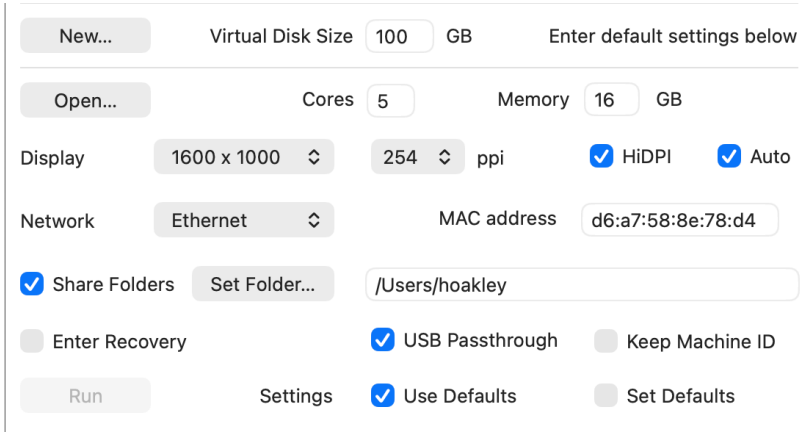
When you create a new VM using Viable, your chosen IPSW file will be copied into the VM for installation, and deleted from there once that's complete. That ensures your original copy of the IPSW file is left intact and safe.

To obtain IPSW files for older versions of macOS, use a site such as Mr. Macintosh, who provides links to Apple's servers. Those for beta releases are available for members of beta-testing programmes.

➜ Create new VM ➜ Open & run VM

➜ Start

Create a new VM



Create a new VM using the IPSW image file for the version of macOS you want it to run. If you want that to be the current version, use the **Download** button to obtain it direct from Apple. For older versions, the most complete collection of links to Apple's IPSW images files is that of Mr. Macintosh.

Before clicking on the **New** button, check the settings to be used for your new VM:

- **Virtual Disk Size**: sets the size of the virtual block storage for the boot disk. Given in GiB, it can range between 15-2024, with a recommended minimum of 35 GiB. If you want to be able to update macOS within the VM, then a minimum size of 50 GiB is recommended, to ensure sufficient free space for updating. This size is fixed and can't changed once the VM has been created.
- **Cores**, **Memory**, etc.: although these aren't baked into the VM like disk size, and can be changed whenever you wish, settings here are saved in the defaults file inside the VM bundle, and will be used when you run that VM with the **Use Defaults** box ticked, or when double-clicked to run it in Vimy.
- **Keep Machine ID**: this sets the machine identity of the VM to be the same as an ID already loaded. Normally, this should be left unticked so each VM has a unique ID. ➜ How to create VMs with the same machine ID.

Save the VM using the .vimi extension so that it can be opened easily using Viable and Vimy.

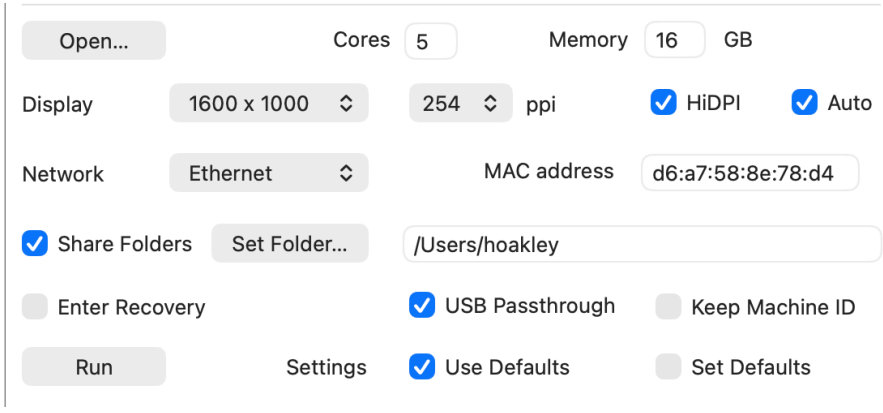
At the start of creation, the IPSW file to be installed in the VM will be copied into the new VM bundle. This requires around 26 GB of disk space. Once creation is complete, that copy is deleted automatically, freeing up that space while leaving your original IPSW file intact and untouched.

Progress during creation is indicated by a spinning indicator to the right of the **New** button. Once that's complete, your new VM is left open in Viable, its path and name shown in the window title bar. When you're happy with its settings, click the **Run** button and your VM will open in a new window and start its configuration sequence, just like a new macOS installation.

➜ Open & run VM ➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders

➜ Start

Open and run a VM



Before you can run a saved VM, click on the **Open** button, select and open it. That opens the default settings saved inside that VM's bundle so you can opt to use those, by ticking √ the **Use Defaults** box, or running it using the custom settings in the controls in this window. If you want to make those the defaults for that VM, tick √ the **Set Defaults** box and they will replace the previous defaults for that VM.

When you open a VM its path and name are displayed in the window title bar.

When you're happy with the settings to be used, click on the **Run** button to start that VM.

When you have finished running a VM, shut it down using the **Shut Down** menu command in the VM, and it will close down and its window will close automatically. The red **Close** button in the VM's window *won't* close a running VM or its window.

If a VM won't shut down or freezes, quit Viable, wait a few seconds, then open Viable, open that VM and try running it again. Provided the VM hasn't suffered internal damage, it should start up correctly again.

➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders ➜ Settings file

Main settings

Standard options supported are:

Cores: this specifies the number of cores on which the VM will be run, which is also the vCPU number. It can any number from 1 upwards, with a recommended minimum of 2, and normally 4. The maximum used is limited to the number of cores in the host to prevent its cores from locking.

Memory: this specifies the size of memory to be given to the VM in GiB, and can range between 4-256, with a normal recommendation for at least 6.

Keyboard: the VM is configured to pass through special key features such as the Globe key by default. When running on Sonoma hosts, additional keyboard support is also provided by default. No separate control is provided.

Share Folders: tick this box if you want the VM to share folders with the host macOS.
Set Folder: click on this button to select a special folder to share between the host macOS and the guest. The path to that folder will then be shown in the box to the right.

Enter Recovery: tick this box if you want the VM to start up in Recovery Mode.
USB Passthrough: this is a teaser for a feature to come in a future version, and only appears when Viable is running in macOS 27 on the host.
Keep Machine ID: this box should normally be left unticked.

➜ Network settings ➜ Display settings ➜ Shared folders ➜ Machine ID ➜ Settings file

Network settings

Viable supports two different types of network connection: NAT and bridged. While NAT is simpler, it's also not as fast, and is relatively inflexible. When using a NAT connection, your VM will have a local IP address assigned to it on a different sub-net, and packets will be transferred through the host Mac's network connection using address translation. That has the advantage of using any filtering or protection provided by the host. Bridged networking shares one of the host's network ports, and gets its own IP address on your local network, as if it was another Mac.

Network popup menu: this offers a choice between bridged ports such as Ethernet and Wi-Fi, NAT, and no network connection. You should prefer Ethernet; Wi-Fi access isn't currently supported, so may not work and fall back to Ethernet.

Address: set the desired MAC address in this text box using hex notation. This defaults to Apple's standard of d6:a7:58:8e:78:d4. If you change it and want to return to the default, simply remove all text from the box and Viable will revert to the default for you. This is most important if you intend running two VMs at the same time. As they can't have the same MAC address, you should then alter the MAC address of one of them to a different setting. Otherwise it's best to leave this alone unless you understand what you're doing.

➜ Main settings ➜ Display settings ➜ Shared folders ➜ Settings file

Display settings

Display sets the native resolution of the virtual display, chosen from 2560 x 1440, 2200 x 1375, 1920 x 1200, 1728 x 1117, 1600 x 1000, 1512 x 982, and 1400 x 875 . Following those are pixel densities ranging from 72 to 254 pixels/inch, a checkbox to determine whether to enable **HiDPI**, and whether to use **Auto** configuration.

These have become simpler to use in more recent versions of macOS on the host: set the initial size using the first popup menu, then set **ppi** to 254, and tick both **HiDPI** and **Auto**. That should provide a window that resizes and adjusts to whatever you want. You should only need to change those settings in exceptional cases.

HiDPI doubles the native resolution set, for example from 2560 x 1440 to 5120 x 2880. When your VM first boots at such a high resolution, windows might appear small. To obtain the best image possible on Retina displays, open the **Displays** pane/settings in the VM, select a Scaled resolution there labelled as **HiDPI** or the lower of the two offered, selecting **Show all resolutions** if necessary. Once set, so long as you open that VM with the same display settings, it will remain in that HiDPI resolution and deliver the best quality. When running a VM on a non-Retina display, you may need to experiment with different settings until you achieve the best image.

To adjust window size to suit virtual display resolution, use the **Snap** command in Viable's Window menu (not in the VM).

➜ Main settings ➜ Network settings ➜ Shared folders ➜ Enhance with ARD ➜ Settings file

➜ Start

Using shared folders

Shared folders are only supported when running Ventura or later in the VM. Three shares are provided as standard when **Share Folders** is ticked:

- Host Documents: ~/Documents
- Host Downloads: ~/Downloads
- Host Home: ~/

The fourth, Custom, is set using the **Shared Folder** button.

Viable isn't sandboxed and its shared folders can be almost any folder on the host Mac. Standard shared folders should normally require the user's consent before Viable can obtain access to the Documents and Desktop folders. Although you could avoid those prompts by giving Viable **full disk access** in **Privacy & Security** settings, that would be more extensive than Viable needs.

While you can run apps from a shared folder, and can set /Applications as your VM's custom shared folder, it's generally best to copy all wanted apps to the guest /Applications folder before launching them from there. Sadly, you can't use this as a workaround to run third-party App Store or other apps that rely on your Apple ID.

ViableS is a sandboxed version of Viable without access to shared folders, intended for security research purposes.

➜ Main settings ➜ Network settings ➜ Display settings ➜ Settings file

➜ Start

VM features currently supported

Two concurrent VMs: given sufficient host resources, you can run two VMs at once. *Don't* try duplicating an existing VM to use for the second, as that will use an identical machine ID. Apple's licence limits the number to two, which is enforced by macOS. To avoid network conflicts, give each a different MAC address.

Rosetta 2: Intel apps can be run within a VM. Additional software required to support Rosetta 2 is downloaded when required.

Bridged and **NAT network** connections. These work sufficiently to connect to the host using File Sharing (using smb://), to download and install macOS updates and security data updates, and for apps that need to access remote servers. iCloud is fully supported in VMs running macOS Sequoia and later.

Virtio and other devices: accelerated graphics, bridged and NAT networking, pointing devices (mouse, trackpad, with full support over Universal Control), Mac keyboard (ANSI layouts, or ISO with a missing key), audio (input and output), entropy (random number generation), FileVault. Although explicit trackpad support is used in Ventura and later, the Trackpad settings item isn't available in the VM.

Configurable CPU core count, memory, display resolution, network address.
Block storage.
Shared folders (Ventura and later only).
Creation of multiple VMs with the same machine ID.
Guest operating systems supported are Monterey and later.
The ability to save and restore VM state isn't yet supported, but is coming soon.

Note that the Virtualization framework doesn't currently support nesting VMs, in that you can't run Viable within a VM.

➜ Controls ➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders
➜ Enhance with ARD ➜ Machine ID

➜ Start

Enhanced features using ARD or VNC

Connect to a VM using Apple Remote Desktop (ARD) or the Screen Sharing app in /System/Library/CoreServices/Applications, started by entering the VM's IP address in the host's browser in a URL of the form `vnc://192.168.64.15`.

Ensure the VM has a network connection, and in its System Preferences or System Settings, open (General) Sharing. Tick the **Remote Management** item there, and tick all the boxes in its dialog for complete control as a VNC client for ARD or Screen Sharing.

In the host select the VM in ARD's **Scanner** view, and click the Control tool. After authenticating to it, use the ARD window containing the VM display and minimise the original VM window into the Dock.

You can now view and control the host in its ARD window, including:

- Drag and drop files between Finder windows in the VM and those on the host, to copy them.
- Change the keyboard on the **host** *(not the VM)* and use that layout; if the keyboard is in ISO format, the additional key at the upper left (under the Escape key) now works as you'd expect.
- Ensure the Clipboard tool in the ARD window is set to use the shared clipboard, and you can copy and paste between the ARD window on the VM and the host.
- Scale the ARD window on the VM to make that window smaller. You can't, though, use it to scale the contents of the window to a larger size.
- Use standard key shortcuts to make screenshots of windows or the display in the ARD window.
- Trackpad controls include gestures set on the host, and work smoothly and reliably with all guests.

These work uniformly across Monterey, Ventura and Sonoma VMs, even though Monterey VMs don't themselves support some of those features.

Shut the VM down using the **Shut Down** menu command in the ARD window on the VM, rather than in the original VM window.

Don't use ARD when you are going to update macOS in the VM, as that will disconnect ARD and could lead to problems.

➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders

➜ Start

Create VMs with the same machine ID

Normally, every VM created is assigned a unique machine identity. This ensures that, if you load and run two at once, they can't have the same ID, which would lead to "undefined behaviour". In other words, don't even try it. Unfortunately, virtualisation doesn't let you assign VMs specific serial numbers or UUIDs, but this is the closest you can come to that, as VMs with identical machine IDs have the same serial numbers and UUIDs.

Viable lets you create as many VMs as you wish with identical machine IDs, but it doesn't check whether you try to load two with the same ID: you have to be meticulously careful to avoid doing that.

Each time Viable starts up, it creates a unique machine ID you can use for this purpose. You can also load into memory any existing machine ID from another VM. With either of those, ticking the **Keep Machine ID** checkbox will keep using the same machine ID for all VMs that you create, until you untick the checkbox or quit the app.

To load a machine ID from an existing VM, use the **Load Machine ID…** command in Viable's **File** menu. In the file selector, load the file named *MachineIdentifier* from inside the VM bundle. If you want to build a library of such IDs for future use, simply copy those MachineIdentifer files into a folder, giving each an appropriate name. When Viable tries to load that file, if it isn't a proper machine ID you'll be informed, and you can try again.

You can only set machine IDs at the time you create a VM. Once it has been saved in the VM's **MachineIdentifier** file, it's fixed for that VM. Trying to change it by replacing the MachineIdentifier file inside the bundle will only cause that VM to freeze during loading, and it will no longer run until you replace its original MachineIdentifier file.

➜ Create new VM ➜ Open & run VM

➜ Start

When a VM goes wrong

The most common problem encountered with failing VMs normally occurs when they run out of resources, memory in particular. Rather than crashing, they usually grind to a halt and freeze. When this happens, quit Viable, and start again, this time giving the VM more memory.

Another issue that can occur is trackpad/mouse/keyboard input behaves bizarrely, or stops working properly. For instance, you can't enter any text in Terminal. A simple trick which can solve that is to select Viable's main window to bring it to the front, then click/tap back in the VM's virtual display. If that doesn't work, try shutting the VM down as normally as possible, quit Viable, and try again.

➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders ➜ Machine ID

➜ Start

Saved settings files

When you create a new VM in Viable, it saves a file inside the VM bundle containing the settings used to run that VM. Whenever you run that VM after that, you can either override those settings with those in Viable's window, or use the saved settings for that VM. Those are the settings used automatically when you run that VM using Vimy, where they can't be overridden or overwritten.

Current saved settings are stored in a property list file named `Settings.plist` inside the VM bundle. As that file *isn't* managed by `cfprefsd`, and is only read when running a VM, you are free to edit it manually using a suitable editor if you wish.

When you first overwrite saved settings in a VM, the previous settings are copied to a backup settings file named `PreviousSettings.plist`. Only one backup settings file is kept, and you can make that the current settings file simply by renaming it `Settings.plist` if you want.

Unlike machine IDs, settings files aren't unique to a VM, and you can copy them between VM bundles if you wish.

➜ Main settings ➜ Network settings ➜ Display settings ➜ Shared folders

Change list

1.1 (build 100):
- remaining tweaks and fixes from previous build
- new Help book
- synced Run button for dual VM running.

1.1 (build 96):
- major revisions to interface and workflows.

1.0.12 (beta 12):
- fixed a bug that displayed the file open dialog when restoring a VM from a minimised window.

1.0.11 (beta 11):
- added bridged networking
- added network popup menu
- improved transcript in scrolling text view.

1.0.10 (beta 10):
- corrected Help book
- added advice to Help book on use of ARD
- fix bug that prevented custom MAC addresses from being used
- improved error messages.

1.0.9 (beta 9):
- added vimi document type
- updated UTType support
- added saved settings features
- redesigned main window, removing redundant keyboard items
- customised menus to buttons
- added support for moving IPSW files
- changed VM info on launch
- replaced RTF help with PDF
- added full tooltips.

1.0.8 (beta 8):
- added support for auto sizing of displays (Sonoma)
- added support for keystroke pass through
- added support for new Mac keyboard (Sonoma)
- added support for custom network addresses
- disable VM window closing while the VM is still running
- improved saving of VM settings.

➜ Start

1.0.7 (beta 7):
- refactored the VM to handle errors more gracefully
- automatically close VM windows when they shut down
- added Shared Folders checkbox (macOS 13 hosts)
- further improvements to UTI/bundle handling
- improved information about VM when it's run
- corrected text for HiDPI checkbox.

1.0.6 (beta 6):
- restored the default .bundle extension when saving VM bundles.

1.0.5 (beta 5):
- revised and fixed shared folder support for Ventura
- added full support for the trackpad in Ventura
- removed the sandbox
- prepared clipboard support, although it remains non-functional.

1.0.4 (beta 4):
- added support for creating VMs with the same machine ID.

1.0.3 (beta 3):
- added more display resolutions
- added HiDPI option
- augmented text output.

1.0.2 (not released):
- added ppi popup menu
- tried (unsuccessfully) shared folder support in macOS 12 host.

1.0.1 (beta 2):
- added Snap command to Window menu, to resize windows
- added 1400 x 875 virtual display resolution
- rerouted errors to report to app main window.

1.0 (beta 1):
- first beta release.

11 November 2026.

➜ Additional licences

Additional licences

The following licence applies to source code incorporated from Apple's sample code into the source code of Viable:

Copyright © 2022 Apple Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright © 2026 Apple Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

➜ Start

Acknowledgement

I’d like to thank Tobias Bussmann who suggested using ARD to enable enhanced features.