

```
url: "https://seimei.do/zh/blog/vmware-fusion-windows-on-arm-devenv/"
title: "在 Mac 上搭建 Windows on ARM 验证环境——VMware Fusion + Windows 11 ARM64"
description: "日本专利厅的电子申请软件只有 Windows 版, Mac 版已停止提供, 也就是说从 Mac 没有申请途径。于
  ↳ 是我在 Apple Silicon Mac 上用 VMware Fusion 搭了一个 Windows 11 ARM64 环境。Fusion 自 2024 年
  ↳ 11 月起连商用也免费, Windows 11 ARM64 的 ISO 也能从 Microsoft 官方获取。本文涵盖安装时最大的关卡
  ↳ ——绕过 Microsoft 账户, 以及把它变成驱动验证机的设置 (关闭 Secure Boot、USB 透传、快照), 基于实际在
  ↳ 用的构成 (Fusion 26.0.0 / M3 Max)。"
pubDate: 2026-08-10
tags: ["windows", "arm64", "vmware", "mac"]
series: ""
related: ["windows-on-arm-rs380-driver"]
graphLabel: "Fusion WoA 验证环境"
graphWeight: 1
lang: zh
draft: false
ogImage: "/blog/vmware-fusion-windows-on-arm-devenv/og-zh.jpg"
```

<https://seimei.do/zh/blog/vmware-fusion-windows-on-arm-devenv/>

搭建这个环境的动机, 是自己申请专利。日本专利厅的互联网电子申请软件只有 Windows 版, Mac 版已于 2021 年 9 月停止提供。也就是说, 从 Mac 没有申请途径。手边的机器只有 Mac。那就用虚拟机把 Windows 跑起来。

它也不只用于申请。公证处的电子确定日期 (电子公证) 之类以 Windows 为前提的行政系统, 同样能直接用这个环境。

即使没有 ARM64 版 Windows 的实机, 只要有一台 Apple Silicon Mac, Windows on ARM 环境就能用虚拟机搭出来。本文是它的搭建记录。手边的实际构成如下。

```
MacBook Pro (Apple M3 Max, 128 GB 内存)
  ↳ VMware Fusion 26.0.0
  ↳ Windows 11 Pro ARM64 (24H2, build 26100) 客户机
    2 vCPU / 4 GB 内存 / 磁盘实际占用约 41 GB / NAT / vTPM
  ↳ USB 透传 ↳ RC-S380 (实机设备)
```

分配给客户机的只有 2 个 vCPU、4 GB 内存, 相当节制, 但装上 MSVC 的 ARM64 工具链 (约 12 GB) 编译驱动也够用了。验证机不必奢侈。

VMware Fusion 如今免费

先说费用。Broadcom 收购后, VMware Fusion 于 2024 年 5 月对个人免费, 自 2024 年 11 月起对商用、教育、个人一律免费且无需授权密钥。业务使用也不产生费用。对 Apple Silicon 的支持从 13.x 系起就是正式功能, 一直延续到现行版本。

虚拟化软件还有别的选择: Parallels Desktop (付费订阅) 和 UTM (免费, 基于 QEMU)。本文的构成用 Fusion, 是因为它免费可商用, 且快照和 USB 透传都齐备。

闲话：大学时买 Parallels

Mac 上的 Windows VM 于我有一段私人渊源。学校指定电脑的系统是 Windows，但我因为是 Sakanaction 的粉丝而买了 Mac。为了两者兼用，入学时买了 Parallels Desktop。

其实没遇到什么麻烦。硬要说的话，是信息课的 Office 套件。Windows 版和 Mac 版在设计和功能上略有差异，稍逊一筹的 Mac 版有时没法照着教科书一步步来。

大一时印象很深的一幕，是用手机热点下载 Windows 8.1 的 ISO。学生优惠白送的那 1 个 Windows 10 授权，不知怎么因为一个报错就浪费掉了，结果用了随时能重新下载的 8.1。

学生时代为了玩 VR，后来还是另买了一台 Windows 台式机。那也是段不错的经历。

所以，喜欢哪个系统就选哪个吧。DIY 攒机党和游戏玩家也许更适合 Windows。不过就我自己而言，当年若没选 Mac，大概不会像现在这样喜欢技术。

Windows 11 ARM64 的 ISO 可从官方获取

以前拿 ARM 版 Windows 得靠 Insider Preview，现在 Microsoft 已官方分发 Windows 11 ARM64 的 ISO。这次用的是 Windows11_26100.4349_Professional_ja-jp_arm64.iso (24H2)。

把下载好的 ISO 拖进 Fusion 的新建 VM 对话框，系统选「Windows 11 64-bit Arm」即可。

Windows 11 的安装要求之一是 TPM，而 Fusion 13.5 起会自动配置 vTPM。新建 VM 时被问到虚拟磁盘的加密方式（快速/完整），是为了满足 vTPM 要求，选快速（只加密配置文件）就够。这次的 VM 也是这个构成。

安装完成后若移除加密或 vTPM 设备，Windows 就无法启动，所以这里别去动。

安装的最大关卡：绕过 Microsoft 账户

安装本身和 x64 版 Windows 没有区别，但 OOBE（初始设置）里有一道关卡。Windows 11 事实上强制要求用 Microsoft 账户登录。我想让验证机用本地账户，因为这是一台会用快照回滚、说坏就重建的一次性机器，不想把个人账户绑上去。

麻烦的是，绕过的手段是和 Microsoft 的猫鼠游戏。曾经的常用手段 BYPASSNRO 脚本，已被 Microsoft 于 2025 年 3 月宣布移除而封堵。

我最初用的是另一个经典手段：「不联网就不会被要求 Microsoft 账户」。在 Fusion 里断开 VM 的网络适配器，就这样走完 OOBE。本地账户确实建成了。然后地狱开始了。

就在到达桌面之前，弹出一个全屏的「解锁您的 Microsoft 体验」，没有关闭按钮。从任务管理器结束掉肇事应用后全屏消失了——结果任务栏和桌面图标再也没出来。

重启 explorer.exe 无效，重启 VM 后直接掉进了 Windows 恢复环境。OOBE 的内部状态大概已经损坏。我放弃逐项修复，把 VM 整个删掉重建。

重建时用的就是 ms-cxh:localonly 方式。这次不断网，正常推进到 Microsoft 账户的登录界面。

1. 推进到「让我们添加你的 Microsoft 账户」界面
2. 用 Shift + F10 打开命令提示符
3. 执行 start ms-cxh:localonly
4. 本地账户创建对话框会直接打开

这条路一路无阻直达桌面。断网方式让 OOBE 走上它意料之外的路径，容易弄坏内部状态；ms-cxh:localonly 更接近预期流程，所以稳定——这是弄坏过一次之后得出的理解。

另外据报告，较新的 25H2 版本连 ms-cxh:localonly 也被封堵了。本文的步骤也请当作有保质期的。比起某个具体的绕过手段，更持久的经验是这个习惯：动手建验证机之前，先确认那一刻还能用的绕过手段是哪个。

让它成为驱动验证机的三个设置

把一台朴素的 Windows VM 变成「驱动验证机」的设置。

1. 在 VM 设置里关闭 Secure Boot

要让自制驱动以测试签名运行，就得开启 testsigning，而这以关闭 Secure Boot 为前提。

在实机上，这意味着进固件设置，还伴随「把日常在用机器的防御降下来」的犹豫。在 VM 上，只需在 Fusion 的高级设置里取消勾选 UEFI Secure Boot。正因为是可丢弃的验证机，才能毫不犹豫地关掉。

虚拟机 > 设置 > 高级 > 取消勾选「启用 UEFI 安全启动」

然后在客户机内执行 bcdedit /set testsigning on 并重启，测试签名的驱动就能加载了。

2. 用 USB 透传把实机设备交给客户机

插在 Mac 上的 USB 设备，可以从 Fusion 的菜单或连接对话框切换到客户机一侧。把 RC-S380 交过去，客户机 Windows 看到的就和物理直插同一个 USB 设备 (usb:054c:06c1)。

在驱动开发的语境里关键的一点是，断开事件可以随手制造。「从 USB 断开自动恢复」的测试，我是靠物理插拔完成的。Fusion 的连接切换应该也能触发同类断开，但那条路我没有试过。

3. 装驱动之前先拍快照

装驱动是会弄脏 OS 状态的操作，失败起来甚至可能到无法启动。用 VM 的话，安装前拍一张快照，不管出什么事都能在几十秒内回到原点。与其为清理 pnputil 注册的驱动包发愁，多数情况下回滚重来更快。

测试签名的证书存储、testsigning 标志、已注册的驱动包——想保住这套组合不被打乱时，一张「能正常工作的状态」的快照就是保险。

VM 验证的局限也写清楚

这里的测试是在「Windows 11 ARM64 的 VM + USB 透传」上做的。没有在搭载 Snapdragon 的实机 Windows PC 上验证过。驱动是 ARM64 原生二进制，原理上应当表现一致，但实机的 USB 控制器或固件差异我还没能排除。若有人在实机上试过，我很想听听结果。

运维中的小毛病：客户机网络单独挂掉的那天

投入使用后，我几次踩到同一个老毛病。某一刻起，只有客户机的网络不通了，VMware Tools 也没了响应。Fusion 的日志里是这样：

```
GuestRpc: app toolbox's second ping timeout; assuming app is down
```

乍看像是客户机里的 Tools 死了。可按时间戳往回翻，早在几个小时之前，另一条消息就一直在刷。

```
VNET: MACVNetPortVirtApiPrimaryIfaceChanged: Global state changed
VNET: MACVNetLinkStateEventHandler: 'ethernet0' state from 4 to 6.
VNET: MACVNetLinkStateTimerHandler: 'ethernet0' state from 6 to 1.
VMXNET3 hosted: Cannot retrieve the buffer descriptors per rx packet. ← 此后这一行不停地刷
```

起点在宿主侧。Mac 的首选网络接口切换的那一刻，客户机的虚拟网卡（VMXNET3）接收处理就对不上了，同一条错误行一直刷下去。翻我自己的日志，量大的那一代里这一行超过了一万行。GuestRpc 的超时不过是下游症状，Tools 是受害者，真凶在别处。

我这边是重启 VM 来恢复（日志上也是一重启就暂时平息），但把它当作一个只要宿主网络一变就可能复发的老毛病，长期相处。

后来，这个老毛病的「关机侧」我也实测了一回。某次复发时，我要关掉一台已经连续运行两天、CPU 烧在 190% 附近、错误行刷了八千多条的 VM。

动手才发现，正常关机的路径全灭了。vmrun stop <vmx> soft 被 VIX_E_TOOLS_NOT_RUNNING 弹回；用 runProgramInGuest 去调客户机里的 shutdown.exe，同样走 Tools，从一开始就不通。

Fusion 菜单里的「关机」（按这台 VM 的设置相当于 ACPI 电源键，本不该依赖 Tools 的路径）也毫无反应。最后只有 vmrun stop <vmx> hard 把它放倒了。

也就是说，这个老毛病拉着 Tools 一起死，从外侧把正常关机的选项一个个夺走：走 Tools 的两条路当场毙命，ACPI 在 CPU 烧着的情况下也可能不通。

hard 等于直接拔电源。但证书库和 testsigning 的设置早已落盘，如果烧的只是网络接收路径，也不存在写盘写到一半的问题。我是按这个判断拉的闸。

这种 hard 关机我这边已经重复过很多次，之后的开机每次都正常。毕竟和拔电源是同一个操作，谈不上推荐——但就这个老毛病而言，至今没有造成实际损失。

教训和驱动开发里踩的地雷是同一个套路：显眼的症状（GuestRpc timeout）和真因（宿主网络变化 → 虚拟网卡）不在同一处。看日志时，先数同一行刷了几千遍，比先去精读那条显眼的更快。

在这个环境之上做了什么？让一台厂商只提供 x64 驱动的 USB 设备跑在 Windows on ARM 上——这个故事写在下一篇里。