VirtualBox使用手册

作者：许富博

第一章 第一步 5

1.1、虚拟化为什么有用？ 5

1.2、一些术语 5

1.3、功能概述 6

1.4、支持的主机操作系统 8

1.5、安装VirtualBox和扩展包 8

1.6、启动VirtualBox 8

1.7、创建第一个虚拟机 10

1.8、运行虚拟机 13

1.9、使用VM组 16

1.16 可选的前端 23

第二章 VirtualBox安装 24

第三章 配置虚拟机 25

3.1。支持的客户操作系统 25

3.2。仿真硬件 25

3.3。常规设置 26

第四章 客户机增强（Guest Additions） 38

4.1、简介 38

4.2、安装和维护Guest Additions 38

4.3、共享文件夹 41

4.4、拖放 44

第五章 存储虚拟化 53

第六章 网络虚拟化 64

6.1、虚拟网络硬件 64

6.2、网络模式简介 65

6.3、网络地址转换（NAT） 65

6.4、网络地址转换服务（NAT网络） 68

6.5、桥接网络 69

6.6、内部联网 70

6.7、仅主机网络 72

6.8、UDP隧道网络 74

6.9、VDE网络 74

6.10、限制网络I / O的带宽 74

6.11、提高网络性能 75

第七章 远程虚拟机 76

第八章 VBoxManage 86

第九章 高级主题 87

第十章 技术细节 132

第十一章VirtualBox编程接口 141

第十二章 问题排查 142

第十三章 安全指南 157

第十四章 已知限制 161

第十五章 更改日志 162

前言

在平时的学习过程中，需要各种各样的计算平台，特别是常用的笔记本为Windows，而工作中的服务器清一色的为Linux，这时，常常使用虚拟机来模拟Linux环境，工欲善其事必先利其器，选择并学习一款合适的虚拟机就很重要了，VMWare好用，但是不是免费软件，因此选择了开源的虚拟机软件：VirtualBox，为了将其使用的更好，计划对其使用进行全面的学习，学习来源来自其官方手册。此学习笔记也来源于其手册。

结构如下：

第一章：简单的介绍了VirtualBox，看完这一章就能构建一个VM，即，快速入门。

第二章：详细介绍VirtualBox在各个平台的安装。

第三章：介绍了虚拟机的配置。

第四章：对客户机进行了更多的介绍。

第五章：介绍了存储的虚拟化。

第六章：介绍了网络虚拟化。

第七章：介绍了远程虚拟机。

第八章：介绍了VirtualBox的命令行管理。

第九章：介绍了一些高级主题。

第十章：介绍了VirtualBox的一些技术细节。

第十一章：介绍VirtualBox的编程接口，想要扩展VirtualBox的人需要阅读。

第十二章：介绍了故障排查。

第十三章：安全指南。

第十四章：已知限制。

第十五章：VirtualBox各个版本的更改记录。

附录A：第三方资料和许可

附录B：VirtualBox隐私信息

词汇表。

本手册主要学习第一章至第十三章，后续的章节需要的话直接到官网去查看相关信息。

VirtualBox官方网址：<https://www.virtualbox.org/>

# 第一章 第一步

VirtualBox是一个跨平台的虚拟化应用程序。这意味着什么？一方面，它安装在现有的基于Intel或AMD的计算机上，无论它们是运行Windows，Mac，Linux还是Solaris操作系统。其次，它扩展了现有计算机的功能，以便它可以同时运行多个操作系统（在多个虚拟机内）。例如，你可以在Mac上运行Windows和Linux，在Linux服务器上运行Windows Server 2008，在Windows PC上运行Linux。你可以根据需要安装和运行尽可能多的虚拟机，唯一的限制是磁盘空间和内存。

VirtualBox简单明了，但非常强大。可以运行在小型嵌入式系统、桌面类机器、一直到数据中心、云环境。

## 1.1、虚拟化为什么有用？

VirtualBox提供的虚拟化技术和功能对于以下几种场景是有用的：

* **同时运行多个操作系统。**VirtualBox允许你一次运行多个操作系统。这样，你可以运行另一个操作系统上编写的软件（例如，Linux或Mac上的Windows软件）。因为你可以为每个这样的操作系统配置特定类型的“虚拟”硬件，所以即使你的实际计算机的硬件不被该操作系统支持，你任然可以在虚拟机中安装，如DOS或OS / 2。
* **软件安装更加方便。** 软件供应商可以使用虚拟机来打包整个软件配置。例如，在真实的机器上安装完整的邮件服务器可能是一个繁琐的任务。使用VirtualBox，可以将这样一个复杂的设置（通常称为“设备”）打包到虚拟机中。安装和运行邮件服务器就像将这样的设备导入VirtualBox一样简单。
* **测试和灾难恢复。** 一旦安装，虚拟机和其虚拟硬盘可以被当做一个容器被任意冻结，唤醒，复制，备份，以及在主机之间传递。最重要的是，通过使用另一个被称为“快照”的VirtualBox功能，你可以保存虚拟机的特定状态，并在必要时恢复到该状态。这样一来，人们就可以自由地实验一个计算环境。如果出现问题（例如在安装不正常的软件或感染病毒后），可以轻松地切换回以前的快照，这可以避免频繁的备份和恢复。你可以创建任何数量的快照，你也可以在虚拟机运行时删除快照以回收磁盘空间。
* **基础设施整合。** 虚拟化可以显着降低硬件和电力成本。大多数情况下，当前的电脑很多情况下的平均系统负载很低，从而浪费了大量的硬件资源和电力。所以，我们不是运行这么多负载这么低的物理计算机，而是可以将许多虚拟机打包到几个强大的主机上，并平衡它们之间的负载。从而充分的利用资源。

## 1.2、一些术语

在研究虚拟化（以及了解本文档的其他部分）时，熟悉以下术语将非常有帮助：

**主机操作系统（host OS）。**

安装VirtualBox的物理计算机的操作系统。本手册讨论所有的VirtualBox版本，如果确实存在主机相关的差异，会在说明时指出来。

**客户操作系统（guest OS）。**

这是在虚拟机内运行的操作系统。理论上，VirtualBox可以运行任何x86操作系统（DOS，Windows，OS / 2，FreeBSD，OpenBSD），但是为了实现虚拟机上近乎原生物理机的性能，我们针对某些操作系统进行了优化。所以当你最喜欢的操作系统作为客户操作系统运行时，可以查看我们是否针对该系统做了优化。

虚拟机（VM）。

这是VirtualBox在客户机操作系统运行时创建的特殊环境。换句话说，你在VM中运行客户机操作系统。通常情况下，虚拟机显示为计算机桌面上的窗口，但根据你使用的VirtualBox的各种不同的前端，你可以以全屏模式显示，也可以在另一台计算机上远程显示。如果以更抽象的方式来看VM，在内部，VirtualBox将VM视为确定其行为的一组参数。它们包括硬件设置（VM应该有多少内存，VirtualBox应该通过哪个硬盘虚拟化容器文件，安装哪些CD等）以及状态信息（VM当前正在运行，保存，还是快照等））。这些参数在VirtualBox Manager窗口或者VBoxManage命令行程序中进行设置，换句话说，虚拟机也是你可以在对话框中看到的东西。由此可见，这里的虚拟器不仅包含了硬件参数，也包含了安装的软件（操作系统、应用软件等）。

**客户机增强。**

这是指与VirtualBox一起提供的特殊软件包，这些软件包用于安装在VM中，以提高客户操作系统的性能并增加额外的功能。

## 1.3、功能概述

以下是VirtualBox主要功能的概述：

* **可移植性。**VirtualBox运行在大量的32位和64位主机操作系统上。VirtualBox是"hosted" hypervisor（有时称为"type 2" hypervisor），需要安装在现有的操作系​​统上。而"bare-metal" 也叫做 "type 1" hypervisor可以直接在硬件上运行。在很大程度上，VirtualBox在所有主机平台上的功能是相同的，并且使用相同的文件和图像格式。这就允许你可以在Windows上创建一个虚拟机，然后在Linux下运行。此外，虚拟机可以使用开放式虚拟化格式（OVF），你甚至可以导入使用不同虚拟化软件创建的OVF。
* **不需要硬件虚拟化支持。**大多数情况下，VirtualBox不需要像Intel VT-x或AMD-V这样内置的处理器功能。这与许多其他虚拟化解决方案相反，因此即使在不存在这些功能的旧硬件上也可以使用VirtualBox。
* 客户机增强（Guest Additions**）：共享文件夹，无缝窗口，3D虚拟化。**VirtualBox Guest Additions是可以安装在受支持的客户虚拟机系统内的软件包， 以提高其性能，**并提供与主机系统的额外集成和通信**。安装客户机增强后，虚拟机将支持自动调整分辨率，无缝窗口，加速3D图形等等。**特别是Guest Additions提供了“共享文件夹”，可让你在客户机系统中访问主机系统的文件**。
* **丰富的硬件支持**：
  + **客户机对称多处理（SMP）。**VirtualBox可以向每个虚拟机最多显示32个虚拟CPU，而不管主机上实际存在多少CPU内核。
  + **USB设备支持。** VirtualBox实现虚拟USB控制器，并允许你将任意USB设备连接到虚拟机，而无需在主机上安装特定的设备驱动程序。USB支持不限于某些设备类别。
  + **硬件兼容性。**VirtualBox虚拟化了大量的虚拟设备，其中包括许多通常由其他虚拟化平台提供的设备。包括IDE，SCSI和SATA硬盘控制器，几个虚拟网卡和声卡，虚拟串行和并行端口以及许多现代PC系统中的输入/输出高级可编程中断控制器（I / O APIC）。
  + **完整的ACPI支持。**VirtualBox完全支持高级配置和电源接口（ACPI）。这简化了将PC映像从真实机器或第三方虚拟机克隆到VirtualBox中的过程。凭借其独特的 ACPI电源状态支持， VirtualBox甚至可以向ACPI感知客户机操作系统报告主机的电源状态。对于使用电池运行的移动系统，客户可以实现节能，并向用户通知剩余电量（例如，以全屏模式）。
  + **多屏幕分辨率。** VirtualBox虚拟机支持的屏幕分辨率是物理屏幕的多倍。
  + **内置对iSCSI的支持。** 这个独特的功能允许你将虚拟机直接连接到iSCSI存储服务器，而无需通过主机系统。VM直接访问iSCSI目标。
  + **PXE网络启动。**VirtualBox的集成虚拟网卡完全支持通过预引导执行环境（PXE）远程启动。
* **多代分支快照。**VirtualBox可以保存虚拟机状态的任意快照。你可以及时返回并将虚拟机还原到任何此类快照，并从中启动备用的VM配置，从而有效地创建一整个快照树。你可以在虚拟机运行时创建和删除快照。
* **VM组。**VirtualBox提供了一个组的功能，使用户能够统一或者单独控制虚拟机。除了基本组之外，任何VM都可以在多个组中，并且组可以嵌套，即组的组。通常，可以对组执行的操作与可以对单个虚拟机执行的的操作相同，例如开始，暂停，重置，关闭（保存状态，发送关闭，关机），丢弃保存状态，在文件系统中显示，排序。
* **清晰的体系结构; 前所未有的模块化。**VirtualBox具有非常优秀的模块化设计，具有非常明确的内部编程接口，客户端和服务器具有清晰的分离界限。这可以让你轻松地从多个接口同时对VirtualBox的虚拟机进行控制：例如，你可以通过单击VirtualBox图形用户界面中的按钮启动虚拟机，然后从命令行甚至远程控制该虚拟机。得益于其模块化的体系架构，VirtualBox还可以通过**软件开发工具包（SDK，即：VirtualBox的扩展编程API）**来公开其全部功能和可配置性 **，**从而将VirtualBox的各个方面与其他软件系统集成。
* **远程机器显示。**VirtualBox远程桌面扩展（VRDE）允许对任何正在运行的虚拟机进行高性能的远程访问。此扩展程序支持最初内置于Microsoft Windows中的远程桌面协议（RDP），还特别添加了完善的客户端USB支持。VRDE不依赖于内置于Microsoft Windows中的RDP服务器; 而是直接插入虚拟化层。因此，它适用于Windows之外的客户机操作系统。

除此之外，VirtualBox还提供了更多独特的功能：

* + **可扩展的RDP认证。**VirtualBox已经支持Windows上的Winlogon和Linux上的PAM进行RDP身份验证。此外，它还包含一个易于使用的SDK，允许你为其他身份验证方法创建任意接口。
  + **USB over RDP。**通过RDP虚拟通道支持，VirtualBox还允许您将本地的USB设备连接到在VirtualBox RDP服务器上远程运行的虚拟机。

## 1.4、支持的主机操作系统

目前，VirtualBox可以在以下主机操作系统上运行：

* Windows主机。
* Mac OS X主机（64位）。
* Linux主机（32位和64位）。
* Solaris主机（仅限64位）。

本节更加详细的信息，去Oracle官网去查询。

## 1.5、安装VirtualBox和扩展包

VirtualBox有许多不同的软件包，安装哪个取决于您的主机操作系统。VirtualBox的安装非常简单：在每个主机平台上，VirtualBox使用最常见且易于使用的安装方法。如果遇到麻烦或有特殊要求，有关各种安装方法的详细信息，请参阅第2章。

从4.0版开始，VirtualBox被分为几个组件。

1. 基本软件包由所有开放源代码组件组成，并根据GNU通用公共许可证V2许可。
2. 可以下载附加的扩展包，以扩展VirtualBox基本包的功能。目前，Oracle提供了一个扩展包，可以在[http://www.virtualbox.org](http://www.virtualbox.org/" \t "_top)上找到并提供以下扩展功能：
   1. 虚拟USB 2.0（EHCI）设备; 请参阅[第3.10.1节“USB设置”](https://www.virtualbox.org/manual/ch03.html#settings-usb)。
   2. 虚拟USB 3.0（xHCI）设备; 请参阅[第3.10.1节“USB设置”](https://www.virtualbox.org/manual/ch03.html#settings-usb)。
   3. VirtualBox远程桌面协议（VRDP）支持; 请参见 [第7.1节“远程显示（VRDP支持）”](https://www.virtualbox.org/manual/ch07.html#vrde)。
   4. 主机摄像头直通; 请参见[第9.7.1节“在访客中使用主机网络摄像头”一](https://www.virtualbox.org/manual/ch09.html#webcam-passthrough)章。
   5. 英特尔PXE引导ROM。
   6. 在Linux主机上PCI实时支持实验支持; 请参见[第9.6节“PCI通过”](https://www.virtualbox.org/manual/ch09.html#pcipassthrough)。
   7. 使用AES算法进行磁盘映像加密; 请参见[第9.31节“磁盘映像的加密”](https://www.virtualbox.org/manual/ch09.html#diskencryption)。

VirtualBox扩展包有 .vbox-extpack文件扩展名。要安装扩展程序，只需双击包文件，将出现一个网络操作管理器窗口，指导你完成所需的步骤。

要查看当前安装的扩展包，请启动VirtualBox Manager（请参阅下一节）。从“文件”菜单中选择“首选项”。在显示的窗口中，转到“扩展”类别，显示当前安装的扩展，在这里你可以删除已有的扩展包或添加新的扩展。或者，你可以在命令行中使用VBoxManage。

## 1.6、启动VirtualBox

安装完成后，可以启动VirtualBox：

* 在Windows主机上，在标准的“程序”菜单中，单击“VirtualBox”组中的项目。
* 在Mac OS X主机上，在Finder中，双击“Applications”文件夹中的“VirtualBox”项目。
* 在Linux或Solaris主机上，根据你的桌面环境，“VirtualBox”项可能已放置在“应用程序”菜单的“系统”或“系统工具”组中。或者，你可以在终端键入 VirtualBox。

当你第一次启动VirtualBox时，应该出现以下窗口：

|  |
| --- |
| https://www.virtualbox.org/manual/images/virtualbox-main-empty.png |

这个窗口叫做**“VirtualBox Manager”。**在左侧，你可以看到一个稍后会列出所有虚拟机的窗格。由于你尚未创建任何内容，因此列表为空。上面的按钮允许您创建新的VM并在现有的虚拟机上工作，一旦你有一些VM之后。右侧的窗格显示当前选择的虚拟机的属性。由于你还没有任何虚拟机，窗格将显示欢迎消息。

为了给你一个直观的感受，VirtualBox创建了许多虚拟机之后可能看起来像下边的样子：

|  |
| --- |
| https://www.virtualbox.org/manual/images/virtualbox-main.png |

## 1.7、创建第一个虚拟机

点击VirtualBox Manager窗口顶部的“新建”按钮。将弹出向导以引导你设置新的虚拟机（VM）：

|  |
| --- |
| https://www.virtualbox.org/manual/images/create-vm-1.png |

在页面上，向导会询问你创建虚拟机所需的基本信息，特别是：

1. **虚拟机名称**稍后将在VirtualBox的管理器窗口中的虚拟机列表中显示，并且它将被用于虚拟机磁盘上的文件。即使可以使用任何名称，但是还是注意最好取一个有意义的名字。
2. 对于**“操作系统类型”，**请选择稍后要安装的操作系统。 如果要安装非常不寻常的未列出的系统，请选择“其他”。根据您的选择，VirtualBox将启用或禁用客户机操作系统可能需要的某些VM设置。这对于64位客户机来说尤其重要。
3. 在接下来的下一页中，需要选择VirtualBox每次启动虚拟机时应分配的**内存（RAM）**。这里给出的内存量将从你的主机系统中取出并呈现给客户机操作系统，该客户机操作系统会将此大小报告为（**虚拟**）计算机的已安装RAM。

**注意：**

请仔细选择RAM这一设置！**在虚拟机运行时，你给虚拟机的内存将不会被你的主机操作系统使用，因此不要设置多于主机系统的内存。**例如，如果你的主机具有1 GB的RAM，并且为VM指定了512MB，而该VM正在运行，那么主机系统上的**所有其他软件**将只剩下512 MB。如果同时运行两个虚拟机，则VirtualBox也将为第二个虚拟机分配内存（如果该内存不可用或者不够用，虚拟机甚至可能无法启动）。一个Windows XP的客户虚拟机将需要至少几百MB的RAM来正常运行，而Windows Vista在RAM少于512 MB时甚至会拒绝安装。当然，如果要在虚拟机中运行图形密集型应用程序，则可能需要更多的RAM。因此，作为经验法则，如果你的主机中有1 GB或更多RAM，通常可以安全地为每个VM分配512 MB。但是，无论如何，请确保主机操作系统上至少有256到512 MB的RAM。否则，可能会导致主机操作系统的虚拟内存系统过度地将内存交换到硬盘，从而使你的主机系统几乎处于停顿状态。与其他设置一样，你可以在创建虚拟机后更改此设置。

1. 接下来，你必须为**虚拟**机指定一个**虚拟硬盘**。

VirtualBox为VM提供硬盘空间方法多种多样（详细信息请参阅[第5章虚拟存储](https://www.virtualbox.org/manual/ch05.html)），但最常见的方法是在“真实”硬盘上使用大型映像文件， VirtualBox会向你的虚拟机呈现其中的内容，就像它是一个完整的硬盘一样。该文件代表整个硬盘，因此你甚至可以将其复制到另一台主机系统，并与另一台VirtualBox一起使用，向导将显示以下窗口：

|  |
| --- |
| https://www.virtualbox.org/manual/images/create-vm-2.png |

这里有以下选项：

* + 要创建一个新的空虚拟硬盘，请按 **“新建”**按钮。
  + 选择**现有的**磁盘映像文件。

在下拉列表窗口中将呈现目前由VirtualBox记忆的所有的磁盘映像。你也可以单击下拉列表旁边的小文件夹按钮打开标准的文件对话框，从而可以在主机系统的磁盘上选择任何磁盘映像文件。如果你是第一次使用VirtualBox，你将需要创建一个新的磁盘映像。因此，按“新建”按钮。这将弹出另一个窗口“创建新的虚拟磁盘向导”，它可以帮助你在新虚拟机的文件夹中创建新的磁盘映像文件。

VirtualBox支持两种映像文件：

* + **动态分配的文件：**只会越来越庞大，大小为客户机系统实际存储在其虚拟硬盘上的数据的大小。因此，主机硬盘驱动器最初会很小，只有随着数据的填充才能增长到指定的大小。
  + **固定大小的文件：**客户机系统将立即占据指定大小的真实磁盘空间，固定大小的文件比动态分配的文件稍快。

更加详细的差异信息，请参见[第5.2节“磁盘映像文件（VDI，VMDK，VHD，HDD）”](https://www.virtualbox.org/manual/ch05.html#vdidetails)。

为了防止物理硬盘被填满，VirtualBox限制了映像文件的大小。然而，它任然需要足够大以容纳你要安装的操作系统和应用程序的内容。对于现代Windows或Linux客户机系统，你可能需要几GB才能进行使用。对于映像文件大小的限制稍后可以更改（详见[第8.23节“VBoxManage modifymedium”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)）。

|  |
| --- |
| https://www.virtualbox.org/manual/images/create-vdi-1.png |

选择或创建磁盘映像文件后，再次按 “下一步”进入下一页。

1. 单击**“完成”后**，将创建新的虚拟机。然后，你将在“管理器”窗口左侧的列表中看到它，并显示最初输入的名称。

## 1.8、运行虚拟机

要启动虚拟机，有如下几种方式：

* 在“管理器”窗口中双击其列表中的条目
* 在“管理器”窗口中的列表中选择其条目，然后按顶部的“开始”按钮
* 对于使用VirtualBox 4.0或更高版本创建的虚拟机，请导航到系统用户主目录中的“VirtualBox VMs”文件夹，找到要启动的计算机的子目录，然后双击虚拟机设置文件（具有 .vbox文件扩展名） 。

一般来说，你可以像使用真正的计算机一样使用虚拟机。有几点需要特别说明一下。

**1.8.1 首次启动新的VM**

当VM首次启动时，将弹出另一个向导 - **“首次启动向导”**，以帮助你选择**安装介质**。由于VM刚刚创建，因此它像一个没有安装操作系统的真实计算机一样：它不会执行任何操作，同时显示一条错误消息指出没有找到可启动的操作系统。

因此，该向导可帮助你从中选择一种介质来安装操作系统。

* 如果你有要安装客户机操作系统的物理CD或DVD介质，请将介质放入主机的CD或DVD驱动器。然后，在向导的安装介质下拉列表中，选择具有正确驱动器号的**“主机驱动器”**（或者在Linux主机的情况下，选择设备文件）。这允许你的虚拟机访问主机驱动器中的安装介质，你可以从那里继续安装。
* 如果以ISO映像文件的形式从互联网下载安装介质（例如Linux发行版），则通常将该文件刻录到空的CD或DVD，然后按照上述进行操作。使用VirtualBox，你可以跳过此步骤并直接安装ISO文件。然后，VirtualBox将该文件作为CD或DVD-ROM驱动器呈现给虚拟机，就像虚拟硬盘映像文件一样。对于这种情况，向导的下拉列表包含以前与VirtualBox一起使用的安装介质的列表。如果你的安装介质不在列表中（特别是如果你是第一次使用VirtualBox），请选择下拉列表旁边的小文件夹图标以显示标准文件对话框，选择主机磁盘上的安装介质。

在这两种情况下，在向导中进行选择之后，你将可以安装操作系统。

**1.8.2 捕获和释放键盘和鼠标**

该节大意是说VirtualBox中的虚拟机如何捕获键盘和鼠标，在客户机操作系统中安装Guest Additions之前，键盘和鼠标只能被虚拟机或者主机系统之中的一个使用，要在二者之间切换，你需要按一个被称为**host key**的键，mac下默认情况是左侧的Command键。你可以在VirtualBox全局设置中更改此默认值，请参见[第1.15节“全局设置”](https://www.virtualbox.org/manual/ch01.html#globalsettings)。 以免您忘记它，在任何情况下，host key的当前设置始终显示在VM窗口的右下方：

|  |
| --- |
| https://www.virtualbox.org/manual/images/vm-hostkey.png |

由于这种行为可能不方便，因此VirtualBox为客户机系统提供了一套名为“VirtualBox Guest Additions”的工具和设备驱动程序，使VM键盘和鼠标操作更加无缝。第四章会对这些信息进行详细描述。

**1.8.3 键入特殊字符**

操作系统有时通过某些组合键启动某些程序。这些组合键中的一些可能难以进入虚拟机，因为有三个候选人等待接收键盘输入：主机操作系统，VirtualBox或客户机操作系统。具体一个按键被谁接收取决于许多因素，包括键本身。

* 主机操作系统为自己保留某些组合键。例如，如果要在虚拟机中重新启动客户机操作系统，则不能输入**Ctrl + Alt + Delete**组合，因为此组合键通常会连接到主机操作系统（Windows和Linux都拦截此操作） ，然后按此组合键将重启主机。

此外，在使用X Window系统的Linux和Solaris主机上，组合键**Ctrl + Alt + Backspace**通常会重置X服务器（重新启动整个图形用户界面）。由于X服务器拦截此组合，所以按下它通常将会重新启动主机图形用户界面（并且在该过程中杀死所有正在运行的程序，包括VirtualBox）。

第三，在支持虚拟终端的Linux主机上，组合键**Ctrl + Alt + Fx**（其中Fx是从F1到F12的功能键之一）通常允许在虚拟终端之间切换。与Ctrl + Alt + Delete一样，这些组合由主机操作系统拦截，因此始终切换主机上的终端 。

如果要将这些组合键发送到虚拟机中的 客户机操作系统，则需要使用以下方法之一：

* + 使用虚拟机窗口的“输入”→“键盘”菜单中的项目。在那里你会发现“插入Ctrl + Alt +Delete”和“Ctrl + Alt + Backspace”; 然而，后者只会对Linux或Solaris客户机系统产生影响。
  + 按下host key和特殊键的组合，然后VirtualBox会为虚拟机翻译：
    - **host key+ Del将**发送Ctrl + Alt + Del（重启客户机）；
    - **host key+ Backspace**发送Ctrl + Alt + Backspace（重新启动Linux或Solaris 客户机虚拟机的图形用户界面）；
    - **host key+ F1**（或其他功能键）来模拟Ctrl + Alt + F1（或其他功能键，即在Linux客户机中的虚拟终端之间进行切换）。
* 对于某些其他键盘组合（如**Alt-Tab**（在打开的窗口之间切换）），VirtualBox允许你配置这些组合键是否会影响主机或客户机系统。这是影响所有虚拟机的全局设置，可以在"File" → "Preferences" → "Input" → "Auto-capture keyboard"下找到。

**1.8.4 更换可移动介质**

虚拟机正在运行时，你可以在虚拟机窗口的“设备”菜单中更改可移动介质

**1.8.5 调整机器的窗口大小**

虚拟机的窗口在运行时可以调整大小。在这种情况下，会发生如下三件事情之一：

1. 如果 启用了**“缩放模式”**，则虚拟机的屏幕将缩放到窗口的大小。如果你有许多机器正在运行，并希望在后台运行时查看其中一个，这将非常有用。如果虚拟机的输出屏幕非常小，则放大窗口可能会很有用，例如你正在运行旧的操作系统。要启用缩放模式，请按**host key+ C**，或从VM窗口的“机器”菜单中选择“缩放模式”。要离开缩放模式，请再次按**host key+ C**。调整窗口大小时，会保留客户机系统屏幕的宽高比。要忽略宽高比，请在调整大小操作期间按Shift键。
2. 如果你安装了Guest Additions并支持自动**调整大小**，则Guest Additions将自动调整客户机操作系统的屏幕分辨率。例如，如果你运行的客户机系统为分辨率为1024x768像素的Windows，然后调整虚拟机窗口的大小以使其宽100像素，则Guest Additions会将Windows显示分辨率更改为1124x768。有关Guest Additions的更多信息，请参考第四章。
3. 如果窗口大于VM的屏幕，屏幕将居中。如果它较小，则滚动条将被添加到虚拟机窗口中（**这个非常不好，因此最好安装Guest Additions（见第四章）并支持自动调整大小或者启用缩放模式**）。

**1.8.6 保存机器的状态**

当你单击虚拟机窗口的“关闭”按钮（在窗口的右上方，就像您将关闭系统上的任何其他窗口）时，VirtualBox会询问你是否要“保存”或“关闭” “虚拟机 （作为快捷方式，你也可以使用“host key + Q”。）

|  |
| --- |
|  |

这三个选项的区别至关重要。他们的意思是：

* **保存机器状态：**通过这个选项，VirtualBox可以通过将其状态完全保存到本地磁盘来“冻结”虚拟机。稍后再次启动虚拟机时，你会发现虚拟机运行在停止的位置。你的所有应用程序仍然会打开，你的计算机将恢复运行。因此，保存虚拟机的状态在某些方面类似于暂停笔记本电脑（例如通过关闭其盖子）。
* **发送关机信号。** 这将发送一个ACPI关闭信号到虚拟机，这具有与在真实计算机上按下电源按钮相同的效果。只要VM运行现代的操作系统，这应该会在VM内触发正确的关机机制。
* **关闭机器：**使用此选项，VirtualBox也将停止运行虚拟机，但不保存其状态。

**警告：**

这相当于真实主机系统的直接断电。如果在断电后再次启动虚拟机，则客户机操作系统将必须完全重新启动，并可能会对其（虚拟）系统磁盘进行长时间的检查。因此，我们通常不要这么做，因为它可能会导致数据丢失或客户机系统在磁盘上不一致的状态。

作为例外，如果你的虚拟机具有任何快照（请参阅下一章节），你可以使用此选项来快速**恢复**虚拟机**的当前快照**。在这种情况下，关闭机器的电源不会中断其状态，但是自从创建快照以来所做的更改将会丢失。

VirtualBox Manager窗口中的**“Discard”**按钮会丢弃虚拟机的保存状态。这与启动它的效果相同。

## 1.9、使用VM组

虚拟机组使用户能够创建自组织的虚拟机，并对其进行集中管理和执行功能，以及单独执行。有一些与群组有关的功能：

1. 使用GUI选项创建组1）在另一个VM的顶部拖动一个虚拟机。

使用GUI选项创建组2）选择多个虚拟机，然后在右键菜单上选择“组”，如下所示：

|  |
| --- |
|  |

1. 命令行选项1）创建一个组并分配一个虚拟机：

VBoxManage modifyvm“Fred”--groups“/ TestGroup”

创建一个组“TestGroup”并将VM“Fred”附加到该组。

命令行选项2）从组中分离虚拟机，并将组删除为空：

VBoxManage modifyvm“Fred”--groups“”

它从VM“Fred”中分离所有组，并删除空组。

1. 多组例如：

VBoxManage modifyvm“Fred”--groups“/ TestGroup，/ TestGroup2”

它创建组“TestGroup”和“TestGroup2”（如果它们还不存在），并将VM“Fred”附加到它们两者中。

1. 嵌套组 - 组的层次结构例如：

VBoxManage modifyvm“Fred”--groups“/ TestGroup / TestGroup2”

它将VM“Fred”附加到“TestGroup”组的“TestGroup2”子组中。

1. 组命令摘要：启动，暂停，复位，关闭（保存状态，发送关机信号，关机），丢弃保存状态，在文件系统中显示，排序。

**1.10 快照**

使用快照，您可以保存虚拟机的特定状态供以后使用。在任何时候，您都可以恢复到该状态，即使自那以后，您可能会大大改变VM。因此，虚拟机的快照类似于如上所述的“保存”状态的机器，但是可以存在许多，并且保存这些保存的状态。

您可以通过首先在VirtualBox Manager中选择一台计算机，然后单击右上角的“快照”按钮来查看虚拟机的快照。在您拍摄机器的快照之前，快照列表将为空，除了“当前状态”项目，它表示虚拟机生命周期中的“现在”点。

**1.10.1 拍摄，恢复和删除快照**

有三个与快照相关的操作：

1. 您可以**拍摄快照**。这将复制机器的当前状态，您可以在任何给定的时间返回到该状态。
   * 如果您的VM当前正在运行，请从VM窗口的“Machine”下拉菜单中选择“Take snapshot”。
   * 如果您的虚拟机当前处于“已保存”状态或“关机”状态（VirtualBox主窗口中虚拟机旁边显示），请单击主窗口右上方的“快照”选项卡，然后
     + 或者在小型相机图标（“拍摄快照”）或
     + 右键单击列表中的“当前状态”项，然后从菜单中选择“拍摄快照”。

在任何情况下，将会弹出一个窗口，并要求您提供快照名称。此名称仅供参考，以帮助您记住快照的状态。例如，一个有用的名称将是“从头开始新鲜安装，无客户添加”或“仅安装Service Pack 3”。如果需要，还可以在“说明”字段中添加更长的文本。

您的新快照将显示在快照列表中。在新快照下方，您将看到一个名为“当前状态”的项目，表示您的VM的当前状态是基于您之前采用的快照的变体。如果以后再拍摄快照，您将看到它们将按顺序显示，每个后续快照都是从较早的快照派生的：

|  |
| --- |
|  |

VirtualBox对您可以拍摄的快照数量没有限制。唯一的实际限制是主机上的磁盘空间：每个快照都存储虚拟机的状态，因此占用了一些磁盘空间。（有关快照中存储的内容的详细信息，请参阅下一节。）

1. 您可以 通过右键单击在快照列表中拍摄的任何快照来**还原**快照。通过恢复快照，您可以及时返回（或转发）：机器的当前状态丢失，并且机器恢复到拍摄快照时的确切状态。[[4](https://www.virtualbox.org/manual/ch01.html" \l "ftn.idm561) ]

**注意**

恢复快照会影响连接到虚拟机的虚拟硬盘驱动器，因为虚拟硬盘驱动器的整个状态也将被还原。这也意味着自快照创建的所有文件和所有其他文件更改*都将丢失。*为了在仍然利用快照功能的同时防止这种数据丢失，可以使用该VBoxManage界面以“直写”模式添加第二个硬盘驱动器， 并使用它来存储数据。快照中*不*包括直写硬盘驱动器 ，当机器恢复时，它们保持不变。见[第5.4节，](https://www.virtualbox.org/manual/ch05.html#hdimagewrites)

为了避免在还原快照时丢失当前状态，可以在还原之前创建新的快照。

通过恢复早期的快照并从中获取更多快照，甚至可以创建一种替代现实，并在虚拟机的这些不同历史之间进行切换。这可以导致整个虚拟机快照树，如上图所示。

1. 您也可以**删除**不影响虚拟机状态的**快照**，但只能释放VirtualBox用于存储快照数据的磁盘上的文件，从而释放磁盘空间。要删除快照，请在快照树中右键单击它，然后选择“删除”。从VirtualBox 3.2起，即使在机器运行时也可以删除快照。

**注意**

而拍摄和恢复快照是相当快速的操作，删除快照可能需要相当长的时间，因为可能需要在多个磁盘映像文件之间复制大量数据。临时磁盘文件在运行过程中也可能需要大量的磁盘空间。

有一些情况在VM运行时无法处理，并且您将在VM关闭时收到执行快照删除所需的适当消息。

**1.10.2 快照内容**

将快照视为您保存的时间点。更正式地，快照包括三件事：

* 它包含VM设置的完整副本，包括硬件配置，以便在恢复快照时，VM设置也将被还原。（例如，如果您更改了硬盘配置或VM的系统设置，那么在还原快照时该功能将被撤消。）

设置的副本存储在机器配置中，一个XML文本文件，因此占用的空间很小。

* 所有连接到机器的虚拟磁盘的完整状态都被保留。回到快照意味着，一直以来，对机器磁盘（逐个文件）进行的所有更改也将被撤消。自创建以来的文件将消失，被删除的文件将被还原，对文件的更改将被还原。

（严格来说，这只适用于“正常”模式下的虚拟硬盘，如上所述，您可以配置磁盘与快照的行为不同;请参见[第5.4节“特殊图像写入模式”，](https://www.virtualbox.org/manual/ch05.html#hdimagewrites)更正式和技术上正确，恢复快照时，虚拟磁盘本身不会恢复，而是在拍摄快照时，VirtualBox会创建差异图像，其中仅包含快照拍摄后的更改，并且快照恢复后，VirtualBox将丢弃该快照差异化图像，从而恢复到以前的状态，这更快，磁盘空间更少，细节复杂， 请参见[第5.5节“差分图像”](https://www.virtualbox.org/manual/ch05.html#diffimages)。）

创建差分图像最初不占用主机磁盘上的大量空间，因为差分图像最初将为空（并且随着每次写入操作到磁盘时动态增长）。创建快照后使用机器的时间越长，但差异图像的大小越大。

* 最后，如果在机器运行时拍摄快照，则机器的内存状态也会保存在快照中（与关闭虚拟机窗口相同的方式可以保存内存）。恢复这样的快照时，执行恢复正好是拍摄快照时的点。

内存状态文件可以与虚拟机的内存大小一样大，因此也将占用相当多的磁盘空间。

**1.11 虚拟机配置**

从“管理器”窗口中的列表中选择一个虚拟机时，您将在右侧看到该机器设置的摘要。

单击顶部工具栏中的“设置”按钮可打开一个详细的窗口，您可以在其中配置所选虚拟机的许多属性。但要小心：即使可以在安装客户机操作系统后更改所有VM设置，某些更改可能会阻止客户机操作系统在安装完成后正常工作。

**注意**

当VM处于“运行”或“保存”状态时，“设置”按钮被禁用。这仅仅是因为设置对话框允许您更改为客户机操作系统创建的虚拟计算机的基本特性，并且，例如，如果将其一半内存从其下的内存中删除，则此操作系统可能无法正常运行脚。因此，如果禁用“设置”按钮，请先关闭当前虚拟机。

VirtualBox提供了可以为虚拟机更改的大量参数。在“设置”窗口中可以更改的各种设置在[第3章*配置虚拟机*](https://www.virtualbox.org/manual/ch03.html)中有详细描述。VirtualBox命令行界面可以使用更多的参数; 参见[第8章*VBoxManage*](https://www.virtualbox.org/manual/ch08.html)。

**1.12 删除虚拟机**

要删除不再需要的虚拟机，请在管理器的VM列表中右键单击它，从出现的上下文菜单中选择“删除”。

将出现一个确认窗口，允许您选择机器是否应该仅从机器列表中删除，还是与其关联的文件也应该被删除。

机器运行时，“删除”菜单项被禁用。

**1.13 克隆虚拟机**

要实验VM配置，测试不同的客户机操作系统级别或仅仅备份VM，VirtualBox可以创建现有虚拟机的完整或链接副本。[[5](https://www.virtualbox.org/manual/ch01.html" \l "ftn.idm611) ]

向导将引导您完成克隆过程：

|  |
| --- |
| https://www.virtualbox.org/manual/images/clone-vm.png |

可以从Manager的VM列表的上下文菜单（选择“Clone”）或所选VM的“Snapshots”视图调用此向导。首先为克隆选择一个新名称。当您选择**重新初始化所有网卡的MAC地址时，每个网卡** 都会获得分配的新MAC地址。当源VM和克隆的VM都必须在同一网络上运行时，这是非常有用的。如果您保持不变，所有网卡具有与源虚拟机中相同的MAC地址。根据您如何调用向导，克隆操作有不同的选择。首先，您需要确定克隆应该链接到源虚拟机还是应该创建完全独立的克隆：

* **完全克隆：**在此模式下，所有依赖的磁盘映像都将复制到新的VM文件夹。克隆可以在没有源虚拟机的情况下完全运行。
* **链接克隆：**在此模式下，将创建新的差异磁盘映像，其中父磁盘映像是源磁盘映像。如果您将源VM的当前状态选择为克隆点，则将隐式创建新的快照。

选择克隆模式后，您需要确定应该克隆的是什么。您可以随时创建当前状态的克隆或全部。当您选择全部时，当前状态以及所有快照都被克隆。您是否从具有其他子项的快照开始，还可以克隆当前状态和所有子项。这将从此快照开始创建一个克隆，并包含所有子快照。

根据连接的磁盘映像的大小和计数，克隆操作本身可以是一个冗长的操作。还要记住，每个快照都有不同的磁盘映像，需要克隆。

机器运行时，“克隆”菜单项被禁用。

有关如何在命令行中克隆VM，请参见[第8.9节“VBoxManage clonevm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-clonevm)。

**1.14 导入和导出虚拟机**

VirtualBox可以以行业标准的开放式虚拟化格式（OVF）导入和导出虚拟机。[[6](https://www.virtualbox.org/manual/ch01.html" \l "ftn.idm638) ]

OVF是许多虚拟化产品支持的跨平台标准，允许创建现成的虚拟机，然后可以将其导入虚拟机（如VirtualBox）。VirtualBox使OVF导入和导出易于访问，并从Manager窗口以及其命令行界面支持。这允许包装所谓的**虚拟设备**：磁盘映像以及易于分发的配置设置。通过这种方式，可以提供完整的即用型软件包（具有应用程序的操作系统），不需要配置或安装，除非导入到VirtualBox。

**注意**

OVF标准很复杂，VirtualBox中的支持是一个持续的过程。特别是，不保证VirtualBox支持其他虚拟化软件创建的**所有**设备。有关已知限制的列表，请参见[第14章*已知限制*](https://www.virtualbox.org/manual/ch14.html)。

OVF格式的设备可以有两种变体：

1. 它们可以作为一个或多个磁盘映像（通常以广泛使用的VMDK格式）（参见[第5.2节“磁盘映像文件（VDI，VMDK，VHD，HDD）”](https://www.virtualbox.org/manual/ch05.html#vdidetails)）和一个XML文本描述文件具有.ovf扩展名的方言。这些文件必须位于VirtualBox的同一目录中才能导入。
2. 或者，上述文件可以一起打包成一个档案文件，通常带有 .ova扩展名。（这样的存档文件使用TAR存档格式的变体，因此可以在任何可以解压缩标准TAR文件的实用程序之外将其解压缩到VirtualBox外部。）

要以上述格式**导入**设备，只需双击OVF / OVA文件即可。[[7](https://www.virtualbox.org/manual/ch01.html" \l "ftn.idm656) ]或者，从“管理器”窗口中选择“文件”→“导入设备”。在出现的文件对话框中，使用文件扩展名.ovf或 导航到.ova文件。

如果VirtualBox可以处理该文件，将出现类似于以下内容的对话框：

|  |
| --- |
| https://www.virtualbox.org/manual/images/ovf-import.png |

这将显示OVF文件中描述的虚拟机，并允许您通过双击描述项来更改虚拟机设置。单击**“导入”后**，VirtualBox将复制磁盘映像，并使用对话框中描述的设置创建本地虚拟机。这些将显示在经理的虚拟机列表中。

请注意，由于磁盘映像往往较大，虚拟设备附带的VMDK映像通常以特殊压缩格式运输，不适合直接由虚拟机使用，因此需要首先解压缩并复制映像花几分钟。

有关如何在命令行中导入映像的信息，请参见[第8.10节“VBoxManage导入”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-import)。

相反，要**导出** VirtualBox中已有的虚拟机，请选择“文件”→“导出设备”。显示一个不同的对话框窗口，可以将多个虚拟机组合到一个OVF设备中。然后，选择存储目标文件的目标位置，转换过程开始。这可能需要一段时间。

有关如何在命令行导出映像，请参见[第8.11节“VBoxManage导出”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-export)。

**注意**

OVF无法描述为虚拟机拍摄的快照。因此，导出具有快照的虚拟机时，将仅导出计算机的当前状态，导出中的磁盘映像将具有与虚拟机的当前状态相同的“扁平化”状态。

**1.15 全局设置**

可以通过**文件**菜单，选择 **首选项...**项来达到全局设置对话框 。它提供了适用于当前用户的所有虚拟机的设置选项，或者对于整个系统的**扩展**情况：

1. **常规**使用户可以为VM文件和VRDP认证库指定默认文件夹/目录。
2. **输入**使用户可以指定主机密钥。它标识切换光标是否位于VM或主机操作系统窗口焦点的键（请参见[第1.8.2节“捕获和释放键盘和鼠标”](https://www.virtualbox.org/manual/ch01.html#keyb_mouse_normal)），还可用于触发某些VM操作（请参阅[第1.8.3节“打字特殊字符”](https://www.virtualbox.org/manual/ch01.html#specialcharacters)）
3. **更新**使用户可以指定自动更新的各种设置。
4. **语言**使用户能够指定GUI语言。
5. **显示**使用户可以指定屏幕分辨率及其宽度和高度。
6. **网络**使用户可以配置主机网络的详细信息。
7. **扩展**使用户能够列出和管理已安装的扩展包。
8. **代理**使用户可以配置HTTP代理服务器。

## 1.16 可选的前端

如[第1.3节“特性概述”中](https://www.virtualbox.org/manual/ch01.html#features-overview)提到的，VirtualBox具有非常灵活的设计，允许使用多个接口来控制相同的虚拟机。例如，你可以使用VirtualBox Manager窗口启动虚拟机，然后从命令行停止。通过VirtualBox对远程桌面协议（RDP）的支持，你甚至可以在远程运行虚拟机，并通过网络重定向所有图形输出。

具体来说，以下前端在VirtualBox标准包中发布：

1. VirtualBox是VirtualBox Manager。该图形用户界面使用Qt工具包；本用户手册的大部分内容都用于描述它。它最容易使用，但是不包括一些更高级的VirtualBox功能。
2. VBoxManage是命令行界面，用于自动化和精细地控制VirtualBox的各个方面。它在第8章中描述[第8章中，VBoxManage](https://www.virtualbox.org/manual/ch08.html)。
3. VBoxSDL是一个替代的，简单的图形前端，具有有限的功能集， VBoxSDL在[第9.1节“VBoxSDL，简化的VM显示器”中有描述](https://www.virtualbox.org/manual/ch09.html#vboxsdl)。本手册我们不学习该前端，本手册主要学习简单实用的VirtualBox Manager和命令行界面VboxManage。
4. 最后，VBoxHeadless是另一个前端，在主机上根本不产生可视输出，但如果为虚拟机安装并启用了VirtualBox远程桌面扩展（VRDE），则可以充当RDP服务器。与其他图形界面相反，VBoxHeadless不需要图形支持。这是非常有用的，例如，你要在没有安装X Window系统的Linux服务器上托管虚拟机。

如果上述前端仍然不能满足你的特殊需求，你还可以为VirtualBox创建另一个前端，因为VirtualBox内核将其所有功能整合到一个干净的API中; 请参考[第11章VirtualBox编程接口](https://www.virtualbox.org/manual/ch11.html)。

# 第二章 VirtualBox安装

本章主要说明VirtualBox在不同平台上的安装，主要包括：Windows、Linux、Solaris、Mac OS X。一般只要安装过软件的，VirtualBox的安装与他们没有差别。详细的信息可以参考其官方手册。

<https://www.virtualbox.org/manual/ch02.html>

# 第三章 配置虚拟机

通过第一章，快速了解了VirtualBox并可以运行一台虚拟机，本章详细介绍如何配置虚拟机。

在向客户机操作系统提供虚拟硬件方面你有很大的自由度。虚拟硬件可用于与主机系统或其他客户机系统进行通信。例如，如果你在ISO文件中为VirtualBox提供了CD-ROM的映像，则VirtualBox可将该映像呈现给客户机系统，就像它是物理CD-ROM一样。同样，你可以通过其虚拟网卡让客户机系统访问真正的网络，你也可以让互联网上的主机系统，其他客户机系统访问该客户机系统。

## 3.1。支持的客户操作系统

由于VirtualBox旨在为x86系统提供通用的虚拟化环境，因此可能会运行任何类型的操作系统，但是，VirtualBox重点为某些客机系统进行了优化。具体为哪些客户机操作系统进行了优化，见官方手册：<https://www.virtualbox.org/manual/ch03.html> 由于本笔记主要是学习VirtualBox的使用，因此不在这里列出这些客户机操作系统的列表。通用的Linux客户机系统，VirtualBox支持的也很完善，对于内核版本大于2.6.13的Linux而言，性能会很好，VirtualBox也为其做了优化。

**64位客户机操作系统：**

即使在32位主机操作系统（可以安装在64位的CPU上）上，VirtualBox也支持64位客户机操作系统，但是需要满足以下条件：

1. 你需要具有支持硬件虚拟化的64位处理器（参见[第10.3节“硬件与软件虚拟化”](https://www.virtualbox.org/manual/ch10.html#hwvirt)）。
2. 你必须为需要64位支持的特定虚拟机启用硬件虚拟化; 64位虚拟机不支持软件虚拟化。
3. 如果要在32位主机操作系统上使用64位客户机支持，还必须为特定的VM选择64位客户机操作系统。由于在32位主机上支持64位虚拟机系统会引起额外的开销，VirtualBox仅在明确要求时启用此支持。在64位主机（通常具有硬件虚拟化支持）的情况下，始终支持64位客户机操作系统，无论设置如何，因此你可以在客户虚拟机中简单安装64位操作系统。

**警告：**

在任何主机上，你应该为要在64位模式下使用的虚拟机启用**I / O APIC**。这对于64位Windows VM尤其如此。请参见 第3.3.2节“高级”选项。此外，对于64位Windows客户机系统，你应该确保虚拟机使用 **英特尔网络设备**，因为AMD PCNet卡没有64位驱动程序的支持; 请参见[第6.1节“虚拟网络硬件”](https://www.virtualbox.org/manual/ch06.html#nichardware)。

如果你使用VirtualBox图形用户界面的“创建VM”向导，VirtualBox将为每个选定的64位操作系统自动使用正确的设置。

## 3.2。仿真硬件

VirtualBox几乎虚拟化主机的所有硬件。根据VM的配置，客户虚拟机将看到以下虚拟硬件：

* **输入设备。**默认情况下，VirtualBox模拟标准的PS / 2键盘和鼠标。此外，VirtualBox可以提供虚拟USB输入设备，以避免捕获鼠标和键盘。
* **图形。**VirtualBox图形设备（有时称为VGA设备）几乎与所有其他仿真设备不同，它不以任何物理设备为基础。
* **存储。**VirtualBox目前模拟了Intel PIIX3 / PIIX4芯片，SATA（AHCI）接口和两个SCSI适配器（LSI Logic和BusLogic）上的标准ATA接口; 有关详细信息，请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。而提供其中的一个对于VirtualBox本身就足够了，因此与其他虚拟机管理程序兼容需要这种多种存储适配器。Windows对其引导设备尤其挑剔，如果存储控制器不同，则在虚拟机管理程序之间迁移虚拟机是非常困难或不可能的。
* **联网。**请参见[第6.1节“虚拟网络硬件”](https://www.virtualbox.org/manual/ch06.html#nichardware)。
* **USB。**VirtualBox模拟三个USB主机控制器：xHCI，EHCI和OHCI。虽然xHCI处理所有传输速度的USB设备，但仅在2011年之后发布的客户操作系统支持xHCI。请注意，对于Windows 7客户机系统，为支持xHCI，必须安装第三方驱动程序。较旧的操作系统通常支持OHCI和EHCI。因为OHCI只处理USB低速和全速设备（USB 1.x和2.0），而EHCI只处理高速设备（仅限于USB 2.0），因此需要两个控制器。仿真的USB控制器不直接与主机上的设备进行通信，而是与虚拟USB层进行通信，USB层抽象USB协议并允许使用远程USB设备。
* **音频。**请参见[第3.7节“音频设置”](https://www.virtualbox.org/manual/ch03.html#settings-audio)。

## 3.3。常规设置

在“设置”窗口的“常规”下，您可以配置虚拟机的最基本方面，如内存和必需硬件。有三个选项卡，“基本”，“高级”和“描述”。

**3.3.1。“基本”选项卡**

在“常规”设置类别的“基本”选项卡下，您可以找到以下设置：

名称

虚拟机在主窗口的VM列表中显示的名称。在这个名字下，VirtualBox还可以保存虚拟机的配置文件。通过更改名称，VirtualBox也会重命名这些文件。因此，您只能使用主机操作系统文件名中允许的字符。

请注意，在内部，VirtualBox使用唯一标识符（UUID）来标识虚拟机。您可以显示这些 VBoxManage。

操作系统/版本

虚拟机中（或将要）安装的客户机操作系统的类型。这是在“新建虚拟机”向导中指定的相同设置，如[第1.7节“创建第一个虚拟机”中所述](https://www.virtualbox.org/manual/ch01.html#gui-createvm)。

而新创建的VM的默认设置取决于所选的操作系统类型，稍后更改类型对VM设置没有影响; 这个价值就是纯粹的信息和装饰。

**3.3.2。“高级”选项卡**

快照文件夹

默认情况下，VirtualBox将快照数据与其他VirtualBox配置数据一起保存; 请参见[第10.1节“VirtualBox存储其文件的位置”](https://www.virtualbox.org/manual/ch10.html#vboxconfigdata)。通过此设置，您可以为每个VM指定任何其他文件夹。

共享剪贴板

您可以在此处选择客户机操作系统的剪贴板是否应与主机的剪贴板共享。如果选择“双向”，则VirtualBox将始终确保两个剪贴板都包含相同的数据。如果选择“Host to guest”或“Guest to host”，VirtualBox将只会在一个方向复制剪贴板数据。

剪贴板共享需要安装VirtualBox Guest Additions。因此，此设置不另有效果; 有关详细信息[，](https://www.virtualbox.org/manual/ch04.html)请参阅[第4章“ *访客添加*](https://www.virtualbox.org/manual/ch04.html) ”。

默认情况下，共享剪贴板被禁用。有关[说明，](https://www.virtualbox.org/manual/ch13.html#security_clipboard)请参见 [第13.3.2.3节“剪贴板”](https://www.virtualbox.org/manual/ch13.html#security_clipboard)。可以随时使用虚拟机“设备”菜单中的“共享剪贴板”菜单项更改此设置。

拖放

此设置允许启用拖放支持：从主机或客户端选择一个对象（例如文件），并直接在客户机或主机上复制或打开它。多个每VM拖放模式允许限制任一方向的访问。

要拖放工作，客人添加需要安装在客人身上。

**注意**

默认情况下禁用拖放功能。可以随时使用虚拟机“设备”菜单中的“拖放”菜单项更改此设置。

有关详细信息[，](https://www.virtualbox.org/manual/ch04.html#guestadd-dnd)请参见[第4.4节“拖放”](https://www.virtualbox.org/manual/ch04.html#guestadd-dnd)。 [[11](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1331) ]

**3.3.3。“说明”选项卡**

在这里，您可以输入虚拟机的任何描述（如果需要）。这对机器的功能没有影响，但是您可能会发现这个空间很有用，可以记录虚拟机的配置和已安装的虚拟机的软件。

要在描述文本字段中插入换行符，请按 *Shift + Enter*。

**3.4。系统设置**

“系统”类别分组与呈现给虚拟机的基本硬件相关的各种设置。

**注意**

由于Microsoft Windows的激活机制对硬件更改敏感，如果要更改Windows guest虚拟机的硬件设置，其中一些更改可能会触发与Microsoft的另一次激活请求。

**3.4.1。“主板”选项卡**

在“主板”选项卡上，您可以影响通常在真实计算机主板上的虚拟硬件。

基本内存

这将设置在运行时分配给VM的RAM数量。将从主机操作系统请求指定的内存量，因此在尝试启动虚拟机时，主机必须可用或作为可用内存可用，并且在虚拟机运行时不会对主机使用。这与“新建虚拟机”向导中指定的设置相同，如上面[第1.7节“创建第一个虚拟机”上的指导](https://www.virtualbox.org/manual/ch01.html#gui-createvm)所述。

通常，可以在安装客户机操作系统之后更改内存大小（如果不将内存减少到操作系统将不再启动的量）。

引导顺序

此设置确定客户机操作系统将尝试从各种虚拟引导设备启动的顺序。类似于真正的PC的BIOS设置，VirtualBox可以让客户操作系统从虚拟软盘，虚拟CD / DVD驱动器，虚拟硬盘驱动器（每个由其他VM设置定义），网络或无这些。

如果选择“网络”，虚拟机将尝试通过PXE机制从网络启动。这需要在命令行中进行详细配置; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

芯片组

在这里，您可以选择哪些芯片组将呈现给虚拟机。在VirtualBox 4.0之前，PIIX3是唯一可用的选项。对于Mac OS X等现代客户操作系统，旧芯片组不再受到好评。因此，VirtualBox 4.0引入了更现代的ICH9芯片组的仿真，该芯片组支持PCI Express，三个PCI总线，PCI至PCI桥接器和消息信号中断（MSI）。这允许现代操作系统解决更多的PCI设备，并且不再需要IRQ共享。使用ICH9芯片组，也可以配置多达36个网卡（最多8个网络适配器与PIIX3）。

指点装置

老客人的默认虚拟指点设备是传统的PS / 2鼠标。如果设置为*USB平板电脑*，VirtualBox会向虚拟机报告USB平板设备的存在，并通过此设备将鼠标事件传送到虚拟机。第三个设置是*USB Multi-Touch Tablet*，适合最近的Windows客人。

使用虚拟USB平板电脑具有以绝对坐标（而不是作为相对位置更改）报告运动的优势，这允许VirtualBox将虚拟机窗口中的鼠标事件转换为平板电脑事件，而无需在客户端中“捕获”鼠标在[第1.8.2节“捕获和释放键盘和鼠标”中描述](https://www.virtualbox.org/manual/ch01.html#keyb_mouse_normal)。这使得即使没有安装Guest Additions，使用VM也不那么繁琐。[[12](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1371) ]

启用I / O APIC

高级可编程中断控制器（APIC）是近年来取代旧式可编程中断控制器（PIC）的较新的x86硬件特性。使用I / O APIC，操作系统可以使用超过16个中断请求（IRQ），因此避免IRQ共享，提高可靠性。

**注意**

对于64位客户机操作系统，特别是Windows Vista，*需要*启用I / O APIC ; 如果要在虚拟机中使用多个虚拟CPU，那么也是必需的。

然而，对于I / O APIC的软件支持对于Windows之外的某些操作系统来说是不可靠的。此外，使用I / O APIC稍微增加了虚拟化的开销，因此减慢了客户机操作系统。

**警告**

所有从Windows 2000开始的Windows操作系统根据I / O APIC是否可用安装不同的内核。与ACPI一样，因此*，在安装* Windows客户机操作系统*后*，I / O APIC *不能被关闭*。安装后打开它将不起作用。

启用EFI

这使得可扩展固件接口（EFI）可以替代传统的BIOS，并且可能对某些高级用例很有用。有关详细信息[，](https://www.virtualbox.org/manual/ch03.html#efi)请参阅[第3.13节“替代固件（EFI）”](https://www.virtualbox.org/manual/ch03.html#efi)。

硬件时钟在UTC时间

如果选中，VirtualBox将以UTC格式报告系统时间，而不是本地（主机）时间。这会影响虚拟实时时钟（RTC）的运行方式，并可能对于类似Unix的客户机操作系统有用，通常这些操作系统希望将硬件时钟设置为UTC。

此外，默认情况下，您可以关闭VirtualBox提供给客户机操作系统的**高级配置和电源接口（ACPI）**。ACPI是目前的行业标准，允许操作系统识别硬件，配置主板和其他设备并管理电源。由于所有现代PC都包含此功能，Windows和Linux已经支持多年，默认情况下也会在VirtualBox中启用。它只能在命令行上关闭; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

**警告**

所有从Windows 2000开始的Windows操作系统根据ACPI是否可用安装不同的内核，因此在安装Windows客户机操作系统后，*不得关闭* ACPI 。安装后打开它将不起作用。

**3.4.2。“处理器”选项卡**

在“处理器”选项卡上，可以设置客户机操作系统应该看到的虚拟**CPU核心数**。从版本3.0开始，VirtualBox支持对称多处理（SMP），每个虚拟机最多可以显示32个虚拟CPU内核。

但是，您不应该将虚拟机配置为使用比实际可用的更多CPU内核（实际内核，无超线程）。

在此选项卡上，您还可以设置**“CPU执行上限”**。此设置限制主机CPU花费虚拟CPU的时间。默认设置为100％，表示没有限制。50％的设置意味着单个虚拟CPU可以使用多达50％的单个主机CPU。请注意，限制虚拟CPU的执行时间可能会引起客户计时问题。

此外，**“启用PAE / NX”** 设置可确定主机CPU的PAE和NX功能是否将暴露给虚拟机。PAE代表“物理地址扩展”。通常，如果操作系统启用和支持，那么即使32位x86 CPU也可以访问超过4 GB的RAM。这可以通过在存储器地址中添加另外4位来实现，因此使用36位，最多可以解决64 GB。某些操作系统（如Ubuntu Server）需要CPU支持PAE，无法在没有虚拟机的情况下运行。

使用虚拟机运行现代服务器操作系统，VirtualBox还支持CPU热插拔。有关详细信息，请参见[第9.5节“CPU热插拔”](https://www.virtualbox.org/manual/ch09.html#cpuhotplug)。

**3.4.3。“加速”选项卡**

在此页面上，您可以确定VirtualBox是否以及如何使用主机CPU可能支持的硬件虚拟化扩展。大多数CPU在2006年之后建立的情况就是这样。

您可以单独选择每个虚拟机，VirtualBox是否应该使用软件或硬件虚拟化。[[13](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1414) ]

在大多数情况下，默认设置会很好; VirtualBox将根据您在创建虚拟机时选择的操作系统选择合理的默认值。但是，在某些情况下，您可能希望更改这些预配置的默认值。

高级用户可能对软件与硬件虚拟化的技术细节感兴趣; 请参见[第10.3节“硬件与软件虚拟化”](https://www.virtualbox.org/manual/ch10.html#hwvirt)。

如果主机的CPU支持**嵌套分页**（AMD-V）或**EPT**（Intel VT-x）功能，那么除了硬件虚拟化之外，还可以通过启用嵌套分页来显着提高性能。有关技术细节，请参见[第10.7节“嵌套分页和VPID”](https://www.virtualbox.org/manual/ch10.html#nestedpaging)。

从5.0版开始，VirtualBox提供半虚拟化接口，以提高客户操作系统的时间精度和性能。可用的paravirtprovider选项记录在[第8.8节“VBoxManage modifyvm”中](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)的选项下 。有关半虚拟化提供商的更多详细信息，请参见[第10.4节“](https://www.virtualbox.org/manual/ch10.html#gimproviders)半 [虚拟化提供程序”](https://www.virtualbox.org/manual/ch10.html#gimproviders)。

**3.5。显示设置**

视频内存大小

这样可以以MB为单位设置由guest虚拟机提供的虚拟显卡提供的内存大小。与主内存一样，指定的数量将从主机的驻留内存分配。基于视频存储器的数量，可能会提供更高的分辨率和颜色深度。

如果视频内存量太小，GUI将会显示警告，无法将VM切换到全屏模式。最小值取决于虚拟显示器的数量，屏幕分辨率和主机显示的颜色深度以及*3D加速*和 *2D视频加速*的激活。粗略估计是（*颜色深度* / 8）× *垂直像素* × *水平像素* × *屏幕数* = *字节数*。如上所述，任何激活的显示加速设置可能需要额外的内存。

监控计数

通过此设置，VirtualBox可以为虚拟机提供多个虚拟监视器。如果客户机操作系统（如Windows）支持多个连接的显示器，VirtualBox可以假装存在多个虚拟显示器。[[14](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1446) ]支持多达8个这样的虚拟显示器。

多个监视器的输出将显示在主机上并行运行的多个VM窗口中。

然而，在全屏和无缝模式下，它们将使用连接到主机的可用物理显示器。因此，为了使全屏和无缝模式与多个显示器配合使用，您将至少需要与配置虚拟显示器一样多的物理显示器，否则VirtualBox将报告错误。当您处于全屏或无缝模式时，您可以使用视图菜单通过按主机键+主页来配置客户和主机监视器之间的关系。

请参阅[第14章，*已知限制*](https://www.virtualbox.org/manual/ch14.html)。

启用3D加速

如果虚拟机安装了Guest Additions，则可以在此处选择是否支持加速3D图形。有关详细信息，请参见[第4.5.1节“硬件3D加速（OpenGL和Direct3D 8/9）”](https://www.virtualbox.org/manual/ch04.html#guestadd-3d)。

启用2D视频加速

如果具有Microsoft Windows的虚拟机安装了Guest Additions，则可以在此处选择是否应支持加速的2D视频图形。有关详细信息[，](https://www.virtualbox.org/manual/ch04.html#guestadd-2d)请参阅[第4.5.2节“Windows客户端的硬件2D视频加速”](https://www.virtualbox.org/manual/ch04.html#guestadd-2d)。

远程显示

在“远程显示”选项卡下，如果安装了VirtualBox Remote Display Extension（VRDE），则可以启用VirtualBox内置的VRDP服务器。这允许您使用任何标准RDP查看器（如mstsc.exeMicrosoft Windows随附）远程连接到虚拟机的控制台。在Linux和Solaris系统上，您可以使用标准的开源程序rdesktop 。这些功能在[第7.1节“远程显示（VRDP支持）”](https://www.virtualbox.org/manual/ch07.html#vrde)中有详细描述 。

视频截取

在“视频捕获”选项卡下，您可以为此虚拟机启用视频捕获。请注意，在执行VM时，也可以启用/禁用此功能。

**3.6。存储设置**

虚拟机设置中的“存储”类别允许您将虚拟硬盘，CD / DVD和软盘映像和驱动器连接到虚拟机。

在真正的PC中，所谓的“存储控制器”将物理磁盘驱动器连接到计算机的其余部分。同样，VirtualBox将虚拟存储控制器呈现给虚拟机。在每个控制器下，显示连接到控制器的虚拟设备（硬盘，CD / DVD或软盘驱动器）。

**注意**

本节只能给您快速介绍VirtualBox存储设置。由于VirtualBox为您提供了丰富的这方面的丰富选择，所以我们将整个用户手册的全部章节解释为全部细节：请参见[第5章*虚拟存储*](https://www.virtualbox.org/manual/ch05.html)。

如果您使用“创建虚拟机”向导创建机器，则通常会看到如下所示的内容：

|  |
| --- |
| https://www.virtualbox.org/manual/images/vm-settings-harddisk.png |

根据创建虚拟机时选择的客户机操作系统类型，新VM中存储设备的典型布局如下所示：

* 您将看到一个**IDE控制器，**一个虚拟CD / DVD驱动器已附加到IDE控制器的“辅助主站”端口。
* 您还将看到一个**SATA控制器，**它是一种更现代化的存储控制器，用于更高的硬盘数据吞吐量，虚拟硬盘与之连接。您最初通常会有一个这样的虚拟磁盘，但是您可以在上面的屏幕截图中看到，您可以有多个虚拟磁盘，每个由磁盘映像文件（在这种情况下为VDI文件）表示。

如果您使用较早版本的VirtualBox创建虚拟机，则默认存储布局可能会有所不同。您可能只有一个IDE控制器，CD / DVD驱动器和硬盘都已连接到该控制器。如果您在创建虚拟机时选择了较旧的操作系统类型，这也可能适用。由于较旧的操作系统不支持SATA，无需额外的驱动程序，VirtualBox将确保最初不存在此类设备。有关其他信息，请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

VirtualBox还提供了一个**软盘控制器**，这是特别的：您不能添加软驱以外的设备。虚拟软盘驱动器（如虚拟CD / DVD驱动器）可以连接到主机软盘驱动器（如果有的话）或磁盘映像，在这种情况下，必须使用RAW格式。

您可以自由修改这些媒体附件。例如，如果要从创建的另一个虚拟磁盘复制某些文件，则可以将该磁盘作为第二个硬盘连接，如上图所示。您还可以添加第二个虚拟CD / DVD驱动器，或更改附加这些项目的位置。以下选项可用：

* 要**添加另一个虚拟硬盘或CD / DVD或软盘驱动器，请**选择要添加的存储控制器（IDE，SATA，SCSI，SAS，软盘控制器），然后单击下面的“添加磁盘”按钮树。然后，您可以选择“添加CD / DVD设备”或“添加硬盘”。（如果您单击软盘控制器，则可以添加软盘驱动器。）或者，右键单击存储控制器并在其中选择一个菜单项。

在窗口的右侧，您可以设置以下内容：

* 1. 然后，您可以选择虚拟磁盘应连接到的控制器的哪个**设备插槽**。IDE控制器有四个槽，传统上被称为“主要主站”，“主从站”，“次要主站”和“从站”。相比之下，SATA和SCSI控制器最多可为您提供30个可连接虚拟设备的插槽。
  2. 您可以选择要使用的**图像文件**。
     + 对于虚拟硬盘，右侧会显示一个带有下拉列表的按钮，您可以使用标准文件对话框选择一个**虚拟硬盘**文件，或者 **创建一个新的硬盘** （映像文件） “创建新磁盘”向导，这在[第1.7节“创建您的第一个虚拟机”中有介绍](https://www.virtualbox.org/manual/ch01.html#gui-createvm)。

有关支持的映像文件类型的详细信息，请参见[第5.2节“磁盘映像文件（VDI，VMDK，VHD，HDD）”](https://www.virtualbox.org/manual/ch05.html#vdidetails)。

* + - 对于虚拟CD / DVD驱动器，图像文件通常将采用标准ISO格式。最常见的是，您可以从从Internet获取的ISO文件中安装操作系统时选择此选项。例如，大多数Linux发行版都是以这种方式提供的。

对于虚拟CD / DVD驱动器，可以使用以下附加选项：

* + - * 如果从列表中选择**“主机驱动器”**，则主机的物理设备连接到VM，以便客户机操作系统可以读取和写入您的物理设备。例如，如果要从真正的安装光盘安装Windows，这是有用的。在这种情况下，从显示的下拉列表中选择您的主机驱动器。

如果要使用主机驱动器刻录（刻录）CD或DVD，则还需要启用 **“Passthrough”** 选项; 请参见[第5.9节“CD / DVD支持”](https://www.virtualbox.org/manual/ch05.html#storage-cds)。

* + - * 如果选择**“从虚拟驱动器中删除磁盘”，** VirtualBox将向未插入介质的客户端呈现一个空的CD / DVD驱动器。
* 要**删除附件，请** 选择它，然后单击底部的“删除”图标（或右键单击并选择菜单项）。

在客人运行时，可以更改可移动媒体（CD / DVD和软盘）。由于“设置”对话框当时不可用，您还可以从虚拟机窗口的“设备”菜单中访问这些设置。

**3.7。音频设置**

虚拟机的“设置”窗口中的“音频”部分确定虚拟机是否会看到连接的声卡，以及是否应在主机系统上听到音频输出。

如果客人启用了音频，则可以选择Intel AC'97控制器，Intel HD Audio控制器[[15](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1540) ]或SoundBlaster 16卡的仿真。无论如何，您可以选择什么音频驱动程序VirtualBox将在主机上使用。

在Linux主机上，根据您的主机配置，还可以在OSS，ALSA或PulseAudio子系统之间进行选择。在较新的Linux发行版中，应优先选用PulseAudio子系统。

由于Solaris主机上仅支持VirtualBox 5.0 OSS，Solaris主机上不再支持“Solaris Audio”音频后端。

**3.8。网络设置**

虚拟机的“设置”窗口中的“网络”部分允许您配置VirtualBox如何向虚拟机提供虚拟网卡以及它们的运行方式。

首次创建虚拟机时，VirtualBox默认启用一个虚拟网卡，并为其选择“网络地址转换”（NAT）模式。通过这种方式，客户可以使用主机的网络连接到外部世界，外部世界可以连接到您选择在虚拟机外部可见的客户端上的服务。

这个默认设置对于大概95％的VirtualBox用户来说是很好的。然而，VirtualBox在如何虚拟化网络方面非常灵活。它支持每个虚拟机的许多虚拟网卡，前四个可以在“管理器”窗口中进行详细配置。使用VBoxManage命令行可以配置其他网卡。

由于可用的大量选项，我们专门介绍本手册的整个章节来讨论网络配置; 请参阅[第6章*虚拟网络*](https://www.virtualbox.org/manual/ch06.html)。

**3.9。串口**

VirtualBox以易于使用的方式完全支持虚拟机中的虚拟串行端口。[[16](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1554) ]

自从原来的IBM PC以来，个人计算机已经配备了一个或两个串行端口（也称DOS和Windows的COM端口）。串行端口通常与调制解调器一起使用，一些计算机鼠标在USB变得普遍之前曾经连接到串行端口。

虽然串行端口不再像以前那样普遍存在，但仍然有一些重要的用途。例如，串行端口可用于通过零调制解调器电缆设置原始网络，以防以太网不可用。此外，串行端口对于需要进行内核调试的系统程序员来说是不可或缺的，因为内核调试软件通常通过串行端口与开发人员进行交互。使用虚拟串行端口，系统程序员可以在虚拟机上进行内核调试，而不需要真正的计算机连接。

如果启用了虚拟串行端口，客户机操作系统会看到一个标准的16550A兼容UART设备。支持接收和发送数据。这个虚拟串口如何连接到主机是可配置的，细节取决于你的主机操作系统。

您可以使用图形用户界面或命令行 VBoxManage工具来设置虚拟串行端口。对于后者，请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm) ; 在该部分，寻找 --uart和 --uartmode选项。

在这两种情况下，每个虚拟机最多可以配置四个虚拟串行端口。对于每个这样的设备，您将需要确定

1. 通过选择I / O基址和中断（IRQ），虚拟机应该看到什么样的串口。对于这些，我们建议使用传统的值[[17](https://www.virtualbox.org/manual/ch03.html" \l "ftn.idm1568) ]，它们是：
   1. COM1：I / O base 0x3F8，IRQ 4
   2. COM2：I / O base 0x2F8，IRQ 3
   3. COM3：I / O base 0x3E8，IRQ 4
   4. COM4：I / O base 0x2E8，IRQ 3
2. 然后，您将需要确定该虚拟端口应该连接到哪个端口。对于每个虚拟串行端口，您有以下选项：
   1. 您可以选择将虚拟串行端口“断开连接”，这意味着客人将看到设备，但它的行为就好像没有连接电缆一样。
   2. 您可以将虚拟串行端口连接到主机上的物理串行端口。（在Windows主机上，这将是一个名字COM1;在Linux或Solaris主机上，它将是一个设备节点 /dev/ttyS0）。然后，VirtualBox将简单地将从虚拟串行端口接收到并发送到虚拟串行端口的所有数据重定向到物理设备。
   3. 您可以告诉VirtualBox将虚拟串行端口连接到主机上的软件管道。这取决于您的主机操作系统：
      * 在Windows主机上，将通过命名管道发送和接收数据。管道名称必须采用标识虚拟机的格式 \\.\pipe\<name> ，<name>但可以自由选择。
      * 在Mac，Linux或Solaris主机上，使用本地域套接字。必须选择套接字文件名，以便运行VirtualBox的用户具有足够的权限来创建和写入它。该 /tmp目录通常是一个很好的候选人。

在Linux上有各种可以连接到本地域套接字的工具，也可以在服务器模式下创建一个。最灵活的工具是 socat许多发行版的一部分。

* 1. 在这种情况下，您可以配置VirtualBox是否应创建命名管道（或非Windows主机，本地域套接字）本身，还是VirtualBox应该假定管道（或套接字）已存在。使用 VBoxManage命令行选项，这被分别称为“服务器”或“客户端”模式。
  2. 对于两个虚拟机（对应于零调制解调器电缆）之间的直接连接，只需配置一个虚拟机即可创建管道/套接字，另一个虚拟机附加到另一个虚拟机。
  3. 您可以将虚拟串行端口输出发送到文件。此选项对于捕获来宾的诊断输出非常有用。只要运行VirtualBox的用户有足够的权限创建和写入文件，就可以使用任何文件。
  4. TCP套接字：用于通过TCP / IP转发串行流量，充当服务器，或者可以作为连接到其他服务器的TCP客户端。它允许远程机器通过TCP直接连接到客户的串行端口。
     + TCP服务器：取消选中 *连接到现有管道/套接字*复选框并指定**port** 号码。通常为23或2023.请注意，在类UNIX系统上，普通用户必须使用大于1024的端口。

客户端可以使用的软件，如PuTTY 或telnet命令行工具来访问TCP服务器。

* + - TCP客户端：要通过Internet或LAN创建虚拟空调制解调器电缆，另一端可以通过TCP连接指定。如果检查*连接到现有管道/套接字* 复选框，TCP套接字将以客户端模式运行。**hostname:port**

每个虚拟机最多可以配置四个串行端口，但您可以从上面选择任何端口号。但串口不能可靠地共享中断; 如果同时使用两个端口，则它们必须使用不同的中断级别，例如COM1和COM2，而不是COM1和COM3。

**3.10。USB支持**

**3.10.1。USB设置**

虚拟机的“设置”窗口中的“USB”部分允许您配置VirtualBox的复杂USB支持。

VirtualBox可以允许虚拟机直接访问主机上的USB设备。为了实现这一点，VirtualBox向宾客操作系统提供了一个虚拟的USB控制器。一旦guest虚拟机开始使用USB设备，它将在主机上显示为不可用。

**注意**

1. 请注意主机上正在使用的USB设备！例如，如果您允许客人连接到当前安装在主机上的USB硬盘，则启动客人时，它将与主机断开连接，无需正常关机。这可能会导致数据丢失。
2. Solaris主机对USB支持有一些已知的限制; 请参阅[第14章*已知限制*](https://www.virtualbox.org/manual/ch14.html)。

除了允许客人访问您的本地USB设备之外，VirtualBox甚至允许您的客人通过使用VirtualBox远程桌面扩展（VRDE）连接到远程USB设备。有关详细信息，请参见[第7.1.4节“远程USB”](https://www.virtualbox.org/manual/ch07.html#usb-over-rdp)。

在“设置”对话框中，您可以首先配置客户端是否可用USB，然后选择USB支持级别：OHCI for USB 1.1，EHCI（也将启用OHCI），用于USB 2.0或xHCI，用于所有USB速度。

**注意**

xHCI和EHCI控制器作为VirtualBox扩展包发货，必须单独安装。有关详细信息[，](https://www.virtualbox.org/manual/ch01.html#intro-installing)请参见[第1.5节“安装VirtualBox和扩展包”](https://www.virtualbox.org/manual/ch01.html#intro-installing)。

当为虚拟机启用USB支持时，您可以确定哪些设备将自动附加到访客。为此，您可以通过指定USB设备的某些属性来创建所谓的“过滤器”。具有匹配过滤器的USB设备在连接到主机后将自动传递给客户端。没有匹配过滤器的USB设备可以手动传递给客户机，例如使用Devices / USB设备菜单。

单击“USB设备过滤器”窗口右侧的“+”按钮将创建一个**新的过滤器。** 您可以给过滤器一个名称（稍后引用），并指定过滤条件。您指定的标准越多，将选择的设备越精确。例如，如果您仅指定供应商ID为046d，则Logitech生产的所有设备将可供客人使用。另一方面，如果您填写所有字段，则过滤器将仅适用于特定供应商的特定设备型号，甚至不适用于具有不同版本和序列号的其他同类型设备。

具体如下：

1. **供应商和产品编号。**使用USB，每个供应商的USB产品都带有全球独一无二的识别号码，“供应商ID”。同样地，每一行产品都被分配了一个“产品ID”号码。这两个数字通常以十六进制形式（即它们由数字0-9和字母AF组成），冒号将供应商与产品ID分隔开。例如， 046d:c016代表Logitech作为供应商，以及“M-UV69a光轮鼠标”产品。

或者，您也可以按名称指定**“制造商”**和**“产品”**。

要列出连接到主机的所有USB设备及其各自的供应商和产品ID，可以使用以下命令（请参见[第8章*VBoxManage*](https://www.virtualbox.org/manual/ch08.html)）：

VBoxManage列出usbhost

在Windows上，您还可以在设备管理器中查看连接到系统的所有USB设备。在Linux上，您可以使用该 lsusb命令。

1. **序列号。**虽然供应商和产品ID已经非常具体来识别USB设备，但如果您有两个相同品牌和产品线的相同设备，则还需要其序列号才能正确地过滤出来。
2. **远程。**此设置指定设备是仅本地设备，还是仅限于远程设备（通过VRDP），或者。

在Windows主机上，您需要拔下并重新连接USB设备才能为其创建过滤器。

例如，您可以创建一个新的USB过滤器，并指定供应商ID为046d（Logitech，Inc），制造商索引为1，“not remote”。然后，Logitech公司生产的主机系统上的制造商索引为1的任何USB设备对客机系统都是可见的。

几个过滤器可以选择单个设备 - 例如，选择所有罗技设备的过滤器，以及选择特定网络摄像头的设备。

您可以通过单击过滤器名称旁边的复选框来**停用**过滤器而不删除它们。

**3.10.2。Windows和Linux主机的实现说明**

在Windows主机上，内核模式设备驱动程序提供USB代理支持。它实现了一个USB监视器，允许VirtualBox在插入设备时捕获设备，以及USB设备驱动程序来为特定虚拟机声明USB设备。与1.4.0之前的VirtualBox版本相反，在安装驱动程序后，系统重新启动不再需要。此外，您不再需要重新插入VirtualBox的设备来声明它们。

在较新的Linux主机上，VirtualBox通过文件系统中的特殊文件访问USB设备。当安装VirtualBox时，这些可用于vboxusers系统组中的所有用户 。为了能够从访客系统访问USB，请确保您是该组的成员。

在较旧的Linux主机上，使用usbfs文件系统访问USB设备 。因此，执行VirtualBox的用户需要USB文件系统的读写权限。大多数发行版提供了一组（例如 usbusers）VirtualBox用户需要添加的组。此外，VirtualBox只能代理未被Linux主机USB驱动程序声明的虚拟机USB设备。该Driver=条目 /proc/bus/usb/devices将显示您当前声明哪些设备。有关详细信息 [，](https://www.virtualbox.org/manual/ch12.html#ts_usb-linux)请参阅[第12.8.7节“USB不工作”](https://www.virtualbox.org/manual/ch12.html#ts_usb-linux)usbfs。

**3.11。共享文件夹**

共享文件夹允许您轻松地在虚拟机和主机之间交换数据。此功能要求VirtualBox Guest Additives安装在虚拟机中，并在[第4.3节“共享文件夹”](https://www.virtualbox.org/manual/ch04.html#sharedfolders)中有详细描述 。

**3.12。用户界面**

“用户界面”部分允许您更改此VM的用户界面的某些方面。

菜单栏

这个小部件允许您禁用某些菜单（单击菜单以释放它），某些菜单条目（取消选中该条目的禁用复选框）和完整的菜单栏（取消选中最右边的复选框）。

迷你工具栏

在全屏或无缝模式下，VirtualBox可以显示一个小工具栏，其中包含虚拟机菜单栏中通常可用的某些项目。除非您将鼠标悬停在该工具栏上，否则此工具栏会自动缩小到一个小的灰色线条。使用工具栏，您可以从全屏或无缝模式返回，控制机器执行或启用某些设备。如果您不想看到工具栏，请禁用此设置。

第二个设置允许在屏幕顶部显示工具栏，而不是在底部显示。

状态栏

该小部件允许您禁用状态栏的某些图标（取消选中图标的复选框以禁用它），重新排列图标（拖放图标），并禁用完整的状态栏（取消选中最左侧的复选框）。

**3.13。替代固件（EFI）**

从版本3.1开始，VirtualBox包括对可扩展固件接口（EFI）的实验支持，这是一种新的行业标准，旨在最终将传统BIOS替代为稍后引导计算机和某些系统服务的主要接口。

默认情况下，VirtualBox使用虚拟机的BIOS固件。要为给定的虚拟机使用EFI，可以在设备的“设置”对话框中启用EFI（请参见[第3.4.1节“主板”选项卡“](https://www.virtualbox.org/manual/ch03.html#settings-motherboard)）。或者，使用VBoxManage命令行界面，如下所示：

VBoxManage modifyvm“VM名称”--firmware efi

要切换回使用BIOS，请使用：

VBoxManage modifyvm“VM名称”--firmware bios

EFI的一个显着的用户是苹果的Mac OS X，但是最近的Linuxes和Windows（从Vista开始）提供了可以使用EFI启动的特殊版本。

EFI在VirtualBox中的另一个可能用途是开发和测试EFI应用程序，而无需引导任何操作系统。

请注意，VirtualBox EFI支持是实验性的，随着EFI的成熟而变得越来越广泛，将会得到加强。Mac OS X，Linux和较新的Windows客户端已知可以正常工作。Windows 7 guest虚拟机无法使用VirtualBox EFI实现启动。

**3.13.1。EFI中的视频模式**

EFI提供两个不同的视频接口：GOP（图形输出协议）和UGA（通用图形适配器）。Mac OS X使用GOP，而Linux则倾向于使用UGA。VirtualBox提供了一个配置选项来控制两个接口的帧缓冲区大小。

要控制GOP，请使用以下 VBoxManage命令：

VBoxManage setextradata“VM名称”VBoxInternal2 / EfiGopMode N

其中N可以是0,1,2,3,4,5中的一个，分别指的是640x480,800x600,1024x768,1280x1024,1440x900，1920x1200屏幕分辨率。

要更改UGA分辨率：

VBoxManage setextradata“VM名称”VBoxInternal2 / UgaHorizo​​ntalResolution 1440 VBoxManage setextradata“VM名称”VBoxInternal2 / UgaVerticalResolution 900

只有当VM关闭电源时，GOP和UGA的视频模式才能更改，并保持持续，直到更改为止。

**3.13.2。指定引导参数**

目前无法从正在运行的访客中操纵EFI变量（例如，通过nvram在Mac OS X中运行该工具来设置“boot-args”变量，guest虚拟机将无法正常工作）。作为替代方法，可以将“VBoxInternal2 / EfiBootArgs”extradata传递给VM，以便设置“boot-args”变量。更改“boot-args”EFI变量：

VBoxManage setextradata“VM name”VBoxInternal2 / EfiBootArgs <value>

[[9](https://www.virtualbox.org/manual/ch03.html" \l "idm1203) ]参见[第14章，*已知限制*](https://www.virtualbox.org/manual/ch14.html)。

[[10](https://www.virtualbox.org/manual/ch03.html" \l "idm1236) ] VirtualBox 2.0添加了64位客户支持; 在VirtualBox 2.1中添加了对32位主机上的64位客户端的支持。

[[11](https://www.virtualbox.org/manual/ch03.html" \l "idm1331) ] VirtualBox 4.2添加了拖放实验支持。

[[12](https://www.virtualbox.org/manual/ch03.html" \l "idm1371) ]虚拟USB平板电脑添加了VirtualBox 3.2。根据所选的客户机操作系统，默认情况下，新的虚拟机启用此操作系统。

[[13](https://www.virtualbox.org/manual/ch03.html" \l "idm1414) ]在VirtualBox 2.2版之前，软件虚拟化是默认的; 从版本2.2开始，VirtualBox将为您创建的新虚拟机默认启用硬件虚拟化。（由于兼容性原因，现有虚拟机不会自动更改，并且当然可以为每个虚拟机更改默认虚拟机。）

[[14](https://www.virtualbox.org/manual/ch03.html" \l "idm1446) ] VirtualBox 3.2添加了多显示器支持。

[[15](https://www.virtualbox.org/manual/ch03.html" \l "idm1540) ]由于Windows 7及更高版本（以及64位Windows Vista）不支持Intel AC'97控制器，因此，VirtualBox 4.0已添加了Intel HD Audio支持。

[[16](https://www.virtualbox.org/manual/ch03.html" \l "idm1554) ] VirtualBox 1.5添加了串口支持。

[[17](https://www.virtualbox.org/manual/ch03.html" \l "idm1568) ]参见，例如[http://en.wikipedia.org/wiki/COM\_(hardware\_interface）](http://en.wikipedia.org/wiki/COM_(hardware_interface)" \t "_top)。

# 第四章 客户机增强（Guest Additions）

通过前面的学习，已经可以运行一台虚拟机了，但是你会发现在使用过程中，键盘、鼠标这些在你的主机系统和客户机系统之间的切换非常麻烦，你需要经常通过host key来切换，同时你会发现你的客户机系统的屏幕与你的主机显示器不匹配，这些使用起来都不方便。本章介绍的Guest Additions就是解决这些使用不便的问题的。

## 4.1、简介

Guest Additions是用于安装在客户机操作系统中的客户机增强工具。它们由设备驱动程序和应用程序组成，可以优化客户机操作系统，以获得更好的性能和可用性。所有受支持的客户机操作系统的VirtualBox Guest Additions作为单个CD-ROM映像文件提供 ，文件为VBoxGuestAdditions.iso。该映像文件位于VirtualBox的安装目录中。要安装特定客户机系统虚拟机的Guest Additions，请将此ISO文件作为虚拟CD-ROM安装在虚拟机中，然后从中安装。

Guest Additions提供以下功能：

* 1. 鼠标指针集成：在客户机系统和主机系统之间的鼠标切换不需要host key了，只需要移动鼠标即可。
  2. 共享文件夹：提供了一种在主机系统和客户机系统之间交换数据的简单方法。
  3. 更好的视频支持：虚拟机的窗口大小自动调节、更多的视频模式、加速视频和图形性能。
  4. 无缝窗户：通过此功能，虚拟机桌面上显示的各个窗口可以映射到主机的桌面上，底层应用程序就像在主机上运行一样，用处不大，后面会简单描述。
  5. 构建主机系统/虚拟机系统通信通道：提供了管理客户机系统的更多方式，甚至可以通过主机管理客户机系统。
  6. 时间同步：安装了Guest Additions后，VirtualBox可以确保客户机系统时间与主机的时间更好地同步。
  7. 共享剪贴板：安装Guest Additions后，客户机操作系统的剪贴板可以与主机操作系统共享，这个非常实用和方便。
  8. 自动登录。

每个版本的VirtualBox，甚至是次要版本，都附带有自己的Guest Additions。为获得最佳效果，最好将Guest Additions与VirtualBox的版本保持一致。从VirtualBox 3.1开始，Windows和Linux的Guest Additions会自动检查更新。如果主机正在运行的VirtualBox比Guest Additions的版本更新，则在客户机系统中将显示一条通知。

要禁用此更新检查，请将客户机属性/VirtualBox/GuestAdd/CheckHostVersion的值设置 为0。

## 4.2、安装和维护Guest Additions

Guest Additions可用在Windows，Linux，Solaris或OS/2的虚拟机上。由于最为常用的为Windows和Linux因此，接下来主要说明Linux和Windows Guest Additions的添加和卸载。

**4.2.1。Windows Guest Additions**

VirtualBox Windows Guest Additions旨在安装在运行Windows操作系统的虚拟机中。具体支持哪些Windows版本，这里不再列出，参考VirtualBox的官方手册：<https://www.virtualbox.org/manual/ch04.html>。

**4.2.1.1。安装**

在虚拟机菜单栏中的“设备（Devices）”菜单中，VirtualBox有一个菜单项，名为“插入Guest Additions CD映像”，这将Guest Additions ISO文件安装在虚拟机中。然后，Windows客户机不出意外会自动启动Guest Additions安装程序，该安装程序将Guest Additions安装到Windows guest虚拟机中。其他客户操作系统（或者如果自动启动CD上的软件被禁用）需要手动启动安装程序。

如果您希望手动安装新增功能，则可以执行以下步骤：

1. 启动已安装Windows的虚拟机。
2. 从虚拟机菜单栏的“设备”菜单中选择“安装CD / DVD-ROM”，然后选择“CD / DVD-ROM映像”。这将显示[第5.3节“虚拟媒体管理器”中](https://www.virtualbox.org/manual/ch05.html#vdis)介绍[的虚拟媒体管理器](https://www.virtualbox.org/manual/ch05.html#vdis)。
3. 在虚拟媒体管理器中，按“添加”按钮，浏览主机文件系统的 VBoxGuestAdditions.iso 文件：
   * 在Windows主机上，您可以在VirtualBox安装目录中找到此文件（通常在下 C:\Program files\Oracle\VirtualBox）。
   * 在Mac OS X主机上，您可以在VirtualBox的应用程序包中找到此文件。（右键单击Finder中的VirtualBox图标，然后选择*显示包内容，*它位于该 Contents/MacOS 文件夹中。）
   * 在Linux主机上，您可以在additions安装VirtualBox（通常/opt/VirtualBox/）的文件夹中 找到此文件 。
   * 在Solaris主机上，您可以在additions安装VirtualBox（通常/opt/VirtualBox）的文件夹中 找到此文件 。
4. 返回虚拟媒体管理器，选择该ISO文件，然后按“选择”按钮。这将安装ISO文件并将其作为CD-ROM呈现给您的Windows客户端。

除非您的Windows guest虚拟机已禁用自动启动功能，否则Windows现在将从Additions ISO自动启动VirtualBox Guest Additions安装程序。如果自动启动功能已关闭，请VBoxWindowsAdditions.exe从guest虚拟机内的CD / DVD驱动器中选择 启动安装程序。

安装程序将向Windows驱动程序数据库添加多个设备驱动程序，然后调用硬件检测向导。

根据您的配置，它可能会显示驱动程序未经数字签名的警告。您必须确认这些，以继续安装并正确安装添加。

安装后，重新启动客户机操作系统以激活“添加”。

**4.2.1.2。更新Windows客户添加**

如前所述，可以再次运行安装程序来更新Windows Guest添加。这将替换以前的更新版本的Additions驱动程序。

或者，您也可以打开Windows设备管理器，并为两个设备选择“更新驱动程序...”

1. VirtualBox图形适配器和
2. VirtualBox系统设备。

对于每一个，选择提供自己的驱动程序，并使用“从磁盘”将向导指向带有访客添加的CD-ROM驱动器。

**4.2.1.3。无人值守安装**

这个不常用，需要了解的话，请看其官方手册：<https://www.virtualbox.org/manual/ch04.html>。

**4.2.1.4。手动文件提取**

**前面介绍的都是自动安装**Guest Additions中的驱动和应用，如果要手动安装其中的应用和驱动程序，可以通过键入以下内容从Windows Guest Additions中提取文件：

VBoxWindowsAdditions.exe /extract

需要详细了解的话，请看其官方手册：<https://www.virtualbox.org/manual/ch04.html>。

**4.2.2。Linux Guest Additions**

像Windows Guest Addions一样，VirtualBox Guest Addions for Linux也是可以安装在客户机操作系统中的一组设备驱动程序和系统应用程序。具体支持哪些Linux版本，这里不再列出，参考VirtualBox的官方手册：<https://www.virtualbox.org/manual/ch04.html>。

**注意：**

一些Linux发行版已经附带了全部或部分VirtualBox Guest Additions。您可以选择保留发行版的Guest Additions版本，但这些版本通常不是最新版本，而且功能受限制，因此我们建议您将其与VirtualBox附带的Guest添加替换。VirtualBox Linux Guest Additions安装程序会尝试检测现有的安装并替换它们，但是根据分发如何集成Guest Additions，这可能需要一些手动的交互。强烈建议您在更换预先安装的Guest Additions之前，先获取虚拟机的快照。

**4.2.2.1。安装Linux客户添加**

VirtualBox Guest Linux for Linux提供在与上述Windows的Guest Additions相同的虚拟CD-ROM文件中。他们还附带一个引导您完成安装过程的安装程序，尽管由于Linux发行版之间的显着差异，安装可能会稍微复杂一些。

安装通常包括以下步骤：

1. 在安装访客添加之前，您将必须准备您的访客系统来构建外部内核模块。这与[第2.3.2节“VirtualBox驱动程序模块”中](https://www.virtualbox.org/manual/ch02.html#externalkernelmodules)所述类似，但现在必须在Linux *guest*虚拟机中执行此步骤，而不是在Linux主机系统上执行，如此处所述。

如果您怀疑发生错误，请检查您的客人设置是否正确，并尝试执行该命令

rcvboxadd设置

作为根。

1. 将 VBoxGuestAdditions.isoCD文件插入Linux客户机的虚拟CD-ROM驱动器，与[第4.2.1.1节“安装”中](https://www.virtualbox.org/manual/ch04.html#mountingadditionsiso) Windows客户机的描述完全相同。
2. 更改到安装CD-ROM驱动器的目录，并以root身份执行：

sh ./VBoxLinuxAdditions.run

**4.2.2.2。图形和鼠标集成**

在Linux和Solaris客户端，VirtualBox图形和鼠标集成通过X Window系统。VirtualBox可以使用系统的X.Org变体（或与第一个X.Org版本相同的XFree86版本4.3）。在安装过程中，X.Org显示服务器将被设置为使用Guest Additions附带的图形和鼠标驱动程序。

将客户添加安装到受支持的Linux发行版或Solaris系统的新安装（许多不受支持的系统也将正常工作）后，guest虚拟机的图形模式将会更改，以适应主机上的VirtualBox窗口的大小。您还可以通过使用该VBoxManage工具发送“视频模式提示”来请求客户系统切换到特定分辨率。

使用X.Org服务器版本1.3（它是X Window系统版本11的版本7.3的一部分）或更高版本的客户端支持多个访客监视器。客人屏幕的布局可以根据需要使用客户操作系统随附的工具进行调整。

如果您想了解更多关于X.Org驱动程序设置的详细信息（特别是如果您希望在安装程序无法正确处理的设置中使用它们），则应阅读 [第9.4.2节“访客图形和鼠标驱动程序设置深入“](https://www.virtualbox.org/manual/ch09.html#guestxorgsetup)。

**4.2.2.3。更新Linux客户添加**

通过再次使用更新的CD-ROM映像，可以更新安装过程。这将替换更新版本的驱动程序。更新访客添加后，您应该重新启动。

**4.2.2.4。卸载Linux访客添加**

如果您的虚拟机上安装了一个版本的Guest Additions，并且希望在不安装新版本的情况下将其删除，则可以通过将Guest Additions CD映像插入虚拟CD-ROM驱动器中，如上所述，并运行安装程序来自guest虚拟机的CD映像路径中的“uninstall”参数的当前Guest添加：

sh ./VBoxLinuxAdditions.run卸载

虽然这通常可以没有问题，但在某些情况下，您可能需要对客户机（特别是XFree86Config或xorg.conf文件）进行一些手动清理，特别是如果安装的Additions版本或客户机操作系统很旧，或者如果您在安装后对Guest Additions设置进行了自己的更改。

从版本3.1.0开始，您可以通过调用卸载添加

/opt/VBoxGuestAdditions-5.1.22/uninstall.sh

请/opt/VBoxGuestAdditions-5.1.22 用正确的Guest Additions安装目录替换 。

## 4.3、共享文件夹

使用VirtualBox的“共享文件夹”功能，您可以从访客系统访问主机系统的文件。这类似于如何在Windows网络中使用网络共享 - 除了共享文件夹不需要网络，只有访客添加。Windows（2000或更新版本），Linux和Solaris guest虚拟机支持共享文件夹。

共享文件夹必须物理上驻留在 *主机上*，然后与guest虚拟机共享，guest虚拟机使用Guest Addition中的特殊文件系统驱动程序与主机通信。对于Windows客户端，共享文件夹实现为伪网络重定向器; 对于Linux和Solaris guest虚拟机，Guest Additions提供了一个虚拟文件系统。

要在VirtualBox中与虚拟机共享主机文件夹，您必须指定该文件夹的路径，并为其选择一个客户端可以使用它访问的“共享名称”。因此，首先在主机上创建共享文件夹; 然后，在客人中连接到它。

可以通过几种方式为特定的虚拟机设置共享文件夹：

* 在运行的虚拟机的窗口中，您可以从“设备”菜单中选择“共享文件夹”，或单击右下角状态栏上的文件夹图标。
* 如果虚拟机当前未运行，则可以在每个虚拟机的“设置”对话框中配置共享文件夹。
* 从命令行可以使用VBoxManage创建共享文件夹，如下所示：

VBoxManage共享文件夹添加“VM名称” - 名称“sharename”--hostpath“C：\ test”

有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-sharedfolder)请参见[第8.32节“VBoxManage共享文件夹添加/删除”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-sharedfolder)。

有两种类型的股份：

1. 只有VM已经定义的VM共享;
2. 暂时的VM共享，可以在运行时添加和删除，并且在VM停止后不会持久; 为此，将--transient选项添加 到上述命令行。

默认情况下，共享文件夹对主机路径上的文件具有读/写访问权限。要限制guest虚拟机具有只读访问权限，请创建只读共享文件夹。这可以使用GUI或通过--readonly使用VBoxManage创建共享文件夹时附加参数来实现。

从4.0版开始，VirtualBox共享文件夹还支持符号链接（**符号链接**），在以下条件下：

1. 主机操作系统必须支持符号链接（即需要Mac，Linux或Solaris主机）。
2. 目前只有Linux和Solaris Guest Additions支持符号链接。
3. 出于安全考虑，客户机操作系统默认情况下不允许创建符号链接。如果您信任客户机操作系统以不滥用该功能，则可以使用以下命令为“sharename”创建符号链接：

VBoxManage setextradata“VM name”VBoxInternal2 / SharedFoldersEnableSymlinksCreate / sharename 1

这个还是比较重要的，我在共享文件夹下编译MySQL遇到如下问题：就是通过如上方式解决的。

问题：

cmake\_symlink\_library: System Error: Read-only file system

错误的意思就是：文件系统是只读的，不允许创建符号链接，其实文件系统只是不允许创建符号链接而已，可以创建其他文件，但是不管怎么说就是权限问题。

解决办法：

关闭客户机系统；

退出VirtualBox；

在命令行下执行命令：

VBoxManage setextradata VM\_NAME VBoxInternal2/SharedFoldersEnableSymlinksCreate/SHARE\_NAME 1

当时对于我而言：VM\_NAME=CentOS7，SHARE\_NAME=CentOS7。

启动客户机系统

之后再接着编译，不再出现如上问题了。

**4.3.1。手动安装**

您可以像安装普通网络共享一样安装VM内部的共享文件夹：

* 在Windows客户机中，共享文件夹是可浏览的，因此在Windows资源管理器中可见。因此，要将主机的共享文件夹附加到Windows客户端，请打开Windows资源管理器，并在“我的网络位置”→“整个网络”→“VirtualBox共享文件夹”下查找。通过右键单击共享文件夹并从弹出的菜单中选择“映射网络驱动器”，可以为该共享文件夹分配驱动器号。

或者，在Windows命令行上，使用以下命令：

net use x：\\ vboxsvr \ sharename

虽然vboxsvr是固定名称（注意vboxsrv也可以使用），将“x：”替换为要用于共享的驱动器号，并sharename 使用指定的共享名称替换 VBoxManage。

* 在Linux客户机中，使用以下命令：

mount -t vboxsf [-o OPTIONS] sharename mountpoint

要在引导过程中挂载共享文件夹，请将以下条目添加到/ etc / fstab中：

sharename mountpoint vboxsf默认为0 0

* 在Solaris guest虚拟机中，使用以下命令：

mount -F vboxfs [-o OPTIONS] sharename mountpoint

用sharename（VBoxManage或）使用指定的共享名称替换（使用小写） ，以及 mountpoint要将共享装载到客户端的路径（例如 /mnt/share）。通常的挂载规则适用，也就是先创建此目录（如果尚不存在）。

以下是在Solaris上安装用户“jack”的共享文件夹的示例：

$ id uid = 5000（jack）gid = 1（其他） $ mkdir / export / home / jack / mount $ pfexec mount -F vboxfs -o uid = 5000，gid = 1 jackshare / export / home / jack / mount $ cd〜/ mount $ ls sharedfile1.mp3 sharedfile2.txt $

除mount命令提供的标准选项之外 ，还提供以下功能：

iocharset CHARSET

设置用于I / O操作的字符集。请注意，在Linux guest虚拟机上，如果未指定“iocharset”选项，则Guest Additions驱动程序将尝试使用CONFIG\_NLS\_DEFAULT内核选项指定的字符集。如果未设置此选项，则将使用UTF-8。也，

convertcp CHARSET

可用于指定用于共享文件夹名称的字符集（默认为utf8）。

通用安装选项（在安装手册页中记录）也适用。特别有用的是选项 uid， gid并且 mode，由于它们允许普通用户访问（读/写模式，取决于设置），即使root已挂载文件系统。

**4.3.2。自动安装**

从4.0版开始，VirtualBox可以自动挂载共享文件夹。如果为特定共享文件夹启用了自动装载功能，则Guest用户将在用户登录到客户机操作系统后自动挂载该文件夹。详细信息取决于客户机操作系统类型：

* 对于**Windows客户端，**任何自动挂载的共享文件夹都将收到其自己的驱动器号（例如 E:），具体取决于guest虚拟机中剩余的空闲驱动器号。

如果没有空闲的驱动器字母，自动安装将失败; 因此，Windows客户端的自动安装共享文件夹的数量通常限制在22以下。

* 使用**Linux客户端，** 自动挂载的共享文件夹将/media与该前缀一起安装到 目录中sf\_。例如，共享文件夹myfiles将被安装到/media/sf\_myfilesLinux和/mnt/sf\_myfilesSolaris上。

guest属性 /VirtualBox/GuestAdd/SharedFolders/MountPrefix 确定使用的前缀。将该guest属性更改为“sf”以外的值以更改该前缀; 有关详细信息[，](https://www.virtualbox.org/manual/ch04.html#guestadd-guestprops)请参见[第4.7节“访客属性”](https://www.virtualbox.org/manual/ch04.html#guestadd-guestprops)。

**注意**

对自动安装的共享文件夹的访问权限仅授予vboxsf由VirtualBox Guest Additions安装程序创建的用户组 。因此，访客用户必须是该组的成员才具有读/写访问权限，或者具有只读访问权限，以防文件夹不可映射写入。

要将安装目录更改为其他目录 /media，可以设置guest虚拟机属性 /VirtualBox/GuestAdd/SharedFolders/MountDir。

* **Solaris客户机的**行为类似于Linux客户机，除了/mnt用作默认安装目录，而不是 /media。

要在VM运行时应用自动安装的共享文件夹进行任何更改，需要重新启动客户机操作系统。（这仅适用于自动安装的共享文件夹，而不是手动挂载的共享文件夹。）

## 4.4、拖放

从5.0版开始，VirtualBox支持将内容从主机拖放到访客，反之亦然。为了使这个工作，客人必须安装最新的客人添加。

拖放透明地允许从一端到另一端复制或打开文件，目录甚至某些剪贴板格式，例如从主机到访客或从访客到主机。然后，可以在主机和VM之间执行拖放操作，因为它将是主机操作系统上的本地拖放操作。

目前，在主机和客户端都实现了基于Windows和X-Windows的系统的拖放操作。由于X-Windows运行不同的拖放协议，所以现在仅支持最常用的XDND。使用其他协议（如Motif或OffiX）的应用程序不会被VirtualBox识别。

在使用拖放的上下文中，数据的起源称为 **源**，即实际数据来自并指定的位置。另一方面， **目标**是指定来源的数据的位置。将数据从源传输到目标可以以各种方式进行，例如复制，移动或链接。[[19](https://www.virtualbox.org/manual/ch04.html" \l "ftn.idm2171) ]

将数据从主机传输到客户机操作系统时，主机在这种情况下是源，而客户操作系统则是目标。但是，当这样做时，也就是将数据从客户操作系统传输到主机，客户操作系统这一次成为源，主机是目标。

为了安全起见，可以使用虚拟机或VBoxManage的“设备”菜单中的“拖放”菜单项，在每个虚拟机的运行时配置拖放：以下四种模式可用：

|  |
| --- |
| https://www.virtualbox.org/manual/images/dnd-modes.png |

* **禁用**禁用完全拖放。这是创建新VM时的默认值。
* **主机到访客**使得只能执行从主机到访客的拖放操作。
* **“访客到主机”仅**允许从**客户机到主机**执行拖放操作。
* **双向**可以对两个方向执行拖放操作，例如从主机到客户端，反之亦然。

**注意**

拖放支持取决于正在使用的前端; 目前只有VirtualBox Manager前端提供此功能。

要使用VBoxManage来控制当前的拖放模式，请参见[第8章*VBoxManage*](https://www.virtualbox.org/manual/ch08.html)。命令modifyvm 并controlvm允许通过命令行设置虚拟机的当前拖放模式。

**4.4.1。支持的格式**

由于VirtualBox可以在各种主机操作系统上运行，并且还支持广泛的客户端，必须在传送数据格式之后转换某些数据格式，以便目标操作系统（即接收数据的一方）能够处理他们以适当的方式。

**注意**

拖动文件时，无法以任何方式进行数据转换，例如将文件从Linux客户端传输到Windows主机时，Linux特定的行结束将不会转换为Windows。

以下格式由VirtualBox拖放服务处理：

* **纯文本**，从文本编辑器，网络浏览器和终端窗口等应用程序
* **文件**，从文件管理器，如Windows资源管理器，鹦鹉螺和Finder
* **目录**，与文件相同

**4.4.2。已知的限制**

以下限制已知：

* 在Windows主机上，现在可以将内容从**UAC升级（用户帐户控制）**程序拖放 到非UAC升级的程序，反之亦然。所以当使用管理员权限启动VirtualBox时，拖放将无法使用默认情况下以常规用户权限运行的Windows资源管理器。

**4.5。硬件加速图形**

**4.5.1。硬件3D加速（OpenGL和Direct3D 8/9）**

VirtualBox Guest Additions包含对Windows，Linux和Solaris客户端的实验硬件3D支持。[[20](https://www.virtualbox.org/manual/ch04.html" \l "ftn.idm2226) ]

使用此功能，如果虚拟机中的应用程序通过OpenGL或Direct3D 8/9编程接口使用3D功能，而不是使用软件（这将很慢）进行仿真，VirtualBox将尝试使用主机的3D硬件。这适用于所有支持的主机平台（Windows，Mac，Linux，Solaris），前提是主机操作系统可以首先使用加速的3D硬件。

目前，3D加速具有以下先决条件：

1. 它仅适用于某些Windows，Linux和Solaris guest虚拟机。尤其是：
   * 使用Windows客户端的3D加速需要Windows 2000，Windows XP，Vista或Windows 7.支持OpenGL和Direct3D 8/9（不支持Windows 2000）（实验性）。
   * Linux上的OpenGL需要内核2.6.27及更高版本以及X.org服务器1.5及更高版本。Ubuntu 10.10和Fedora 14已经过测试和确认工作。
   * Solaris客户机上的OpenGL需要X.org服务器1.5及更高版本。
2. 必须安装Guest添加。

**注意**

要使基本的Direct3D加速工作在Windows客户机中工作，VirtualBox需要替换虚拟机中的Windows系统文件。因此，Guest Additions安装程序提供Direct3D加速功能作为必须明确启用的选项。此外，您必须在“安全模式”下安装访客添加。但这**并不** 适用于适用于Vista和Windows 7的客人实验WDDM Direct3D的视频驱动程序，请参见[第14章，*已知限制*](https://www.virtualbox.org/manual/ch14.html) 的详细信息。

1. 由于此时3D支持仍然是实验性的，因此默认情况下将被禁用，并且必须在VM设置中**手动启用**（请参见[第3.3节“常规设置”](https://www.virtualbox.org/manual/ch03.html#generalsettings)）。

**注意**

不受信任的客户系统不应该允许使用VirtualBox的3D加速功能，就像不受信任的主机软件不应该被允许使用3D加速。3D硬件的驱动程序通常太复杂，无法正确安全，任何允许访问它们的软件可能会损害运行它们的操作系统。此外，启用3D加速可让客户直接访问VirtualBox主机进程中的大量其他程序代码，这可能可以用来使虚拟机崩溃。

使用VirtualBox 4.1，为Windows Vista和Windows 7客户端添加了Windows Aero主题支持。要启用Aero主题支持，必须安装实验的VirtualBox WDDM视频驱动程序，这可以在Guest Additions安装中使用。由于WDDM视频驱动程序目前仍然是实验性的，因此默认情况下不会安装，并且必须在Guest Additions安装程序中**手动选择** “否”，当Direct3D显示“您是否要安装基本的Direct3D支持”对话框时功能被选中。

**注意**

不同于目前基本的Direct3D支持，WDDM视频驱动程序安装时**不**要求“安全模式”。

默认情况下，Aero主题未启用。启用它

* 在Windows Vista guest中：右键单击桌面，在上下文菜单中选择“个性化”，然后在“个性化”窗口中选择“Windows颜色和外观”，在“外观设置”对话框中选择“Windows Aero”，然后按“好”
* 在Windows 7客户端：右键单击桌面，在上下文菜单中选择“个性化”，并在“个性化”窗口中选择任何Aero主题

从技术上讲，VirtualBox通过在客户机中安装附加的硬件3D驱动程序来实现这一点。该驱动程序充当硬件3D驱动程序，并向客户机操作系统报告（虚拟）硬件能够进行3D硬件加速。当客户端的应用程序通过OpenGL或Direct3D编程接口请求硬件加速时，它们将通过VirtualBox实现的特殊通信隧道发送到主机，然后*主机* 通过主机的编程接口执行请求的3D操作。

**4.5.2。用于Windows客户端的硬件2D视频加速**

从版本3.1开始，VirtualBox Guest Additions包含Windows客户端的实验硬件2D视频加速支持。

使用此功能，如果Windows VM内的应用程序（例如视频播放器）使用2D视频叠加来播放影片剪辑，则VirtualBox将尝试使用主机的视频加速硬件，而不是在软件中执行覆盖拉伸和颜色转换会慢）。它目前适用于Windows，Linux和Mac主机平台，前提是您的主机操作系统可以首先使用2D视频加速。

目前，2D视频加速具有以下先决条件：

1. 它仅适用于Windows客户机（XP或更高版本）。
2. 必须安装Guest添加。
3. 由于此时2D支持仍然是实验性的，因此默认情况下禁用，并且必须在VM设置中**手动启用**（请参见[第3.3节“常规设置”](https://www.virtualbox.org/manual/ch03.html#generalsettings)）。

在技​​术上，VirtualBox通过在Guest Additions视频驱动程序中公开视频叠加DirectDraw功能来实现此功能。驱动程序通过VirtualBox实现的特殊通信隧道向主机发送所有覆盖命令。在主机方面，OpenGL用于实现颜色空间转换和缩放

**4.6。无缝窗户**

使用VirtualBox的“无缝窗口”功能，您可以使虚拟机中显示的窗口并排显示在主机窗口旁边。以下客户机操作系统支持此功能（前提是安装了Guest Additions）：

* Windows客户端（支持添加VirtualBox 1.5）;
* 支持运行X Window系统的Linux或Solaris guest虚拟机（添加VirtualBox 1.6）。

启用无缝窗口（见下文）后，VirtualBox禁止显示客人的桌面背景，允许您在主机的窗口旁边无缝地运行客户机操作系统的窗口：

|  |
| --- |
| https://www.virtualbox.org/manual/images/seamless.png |

要启用无缝模式，启动虚拟机后，按“主机”键（通常是正确的控制键）和“L”。这将扩大虚拟机显示的大小到主机屏幕的大小，并掩盖客户操作系统的背景。要返回“正常”VM显示（即禁用无缝窗口），请再次按主机键和“L”。

**4.7。宾客属性**

从2.1版开始，VirtualBox允许从正在运行的guest虚拟机请求某些属性，前提是已安装VirtualBox Guest Additions并且VM正在运行。这对两件事情是有好处的：

1. 一些预定义的虚拟机特性由VirtualBox自动维护，可以在主机上检索，例如监视虚拟机性能和统计信息。
2. 此外，任意字符串数据可以在客户端和主机之间进行交换。这在两个方向都有效。

为了实现这一点，VirtualBox在VirtualBox客户添加和主机之间建立一个私有通信通道，双方的软件都可以使用这个通道来交换字符串数据，用于任意目的。访客属性只是附加值的字符串键。它们可以由主机和客户端设置（写入），也可以从两侧读取。

除了建立读取和写入值的一般机制之外，VirtualBox Guest Additions还会自动维护一组预定义的访客属性，以便检索有趣的访客数据，如客人的确切操作系统和服务包级别，安装版本客户添加，当前登录到客户操作系统的用户，网络统计信息等。这些预定义的属性都是前缀 /VirtualBox/和组织成一个分层树的树。

当从虚拟机的“机器”菜单中选择“会话信息对话”时，会显示一些此运行时信息。

使用此通道的更灵活的方式是通过 VBoxManage guestproperty命令集; 有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-guestproperty)请参见 [第8.33节“VBoxManage guestproperty”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-guestproperty)。例如，要使给定运行的VM的*所有*可用guest虚拟机具有各自的值列出，请使用以下命令：

$ VBoxManage guestproperty枚举“Windows Vista III” VirtualBox命令行管理界面版本5.1.22 （C）2005-2017甲骨文公司 版权所有。 名称：/ VirtualBox / GuestInfo / OS / Product，值：Windows Vista Business Edition， 时间戳：1229098278843087000，标志： 名称：/ VirtualBox / GuestInfo / OS / Release，value：6.0.6001， 时间戳：1229098278950553000，标志： 名称：/ VirtualBox / GuestInfo / OS / ServicePack，值：1， 时间戳：1229098279122627000，标志： 名称：/ VirtualBox / GuestAdd / InstallDir， 值：C：/ Program Files / Oracle / VirtualBox 客人添加，时间戳：1229098279269739000，标志： 名称：/ VirtualBox / GuestAdd / Revision，值：40720， 时间戳：1229098279345664000，标志： 名称：/ VirtualBox / GuestAdd / Version，value：5.1.22， 时间戳：1229098279479515000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxControl.exe，值：5.1.22r40720， 时间戳：1229098279651731000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxHook.dll，值：5.1.22r40720， 时间戳：1229098279804835000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxDisp.dll，值：5.1.22r40720， 时间戳：1229098279880611000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxMRXNP.dll，值：5.1.22r40720， 时间戳：1229098279882618000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxService.exe，值：5.1.22r40720， 时间戳：1229098279883195000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxTray.exe，值：5.1.22r40720， 时间戳：1229098279885027000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxGuest.sys，值：5.1.22r40720， 时间戳：1229098279886838000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxMouse.sys，值：5.1.22r40720， 时间戳：1229098279890600000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxSF.sys，值：5.1.22r40720， 时间戳：1229098279893056000，标志： 名称：/VirtualBox/GuestAdd/Components/VBoxVideo.sys，值：5.1.22r40720， 时间戳：1229098279895767000，标志： 名称：/ VirtualBox / GuestInfo / OS / LoggedInUsers，value：1， 时间戳：1229099826317660000，标志： 名称：/ VirtualBox / GuestInfo / OS / NoLoggedInUsers，value：false， 时间戳：1229098455580553000，标志： 名称：/ VirtualBox / GuestInfo / Net / Count，value：1， 时间戳：1229099826299785000，标志： 名称：/ VirtualBox / HostInfo / GUI / LanguageID，value：C， 时间戳：1229098151272771000，标志： 名称：/ VirtualBox / GuestInfo / Net / 0 / V4 / IP，值：192.168.2.102， 时间戳：1229099826300088000，标志： 名称：/ VirtualBox / GuestInfo / Net / 0 / V4 / Broadcast，值：255.255.255.255， 时间戳：1229099826300220000，标志： 名称：/ VirtualBox / GuestInfo / Net / 0 / V4 /网络掩码，值：255.255.255.0， 时间戳：1229099826300350000，标志： 名称：/ VirtualBox / GuestInfo / Net / 0 /状态，值：向上， 时间戳：1229099826300524000，标志： 名称：/ VirtualBox / GuestInfo / OS / LoggedInUsersList，value：username， 时间戳：1229099826317386000，标志：

要查询单个属性的值，请使用“get”子命令，如下所示：

$ VBoxManage guestproperty获取“Windows Vista III”“/ VirtualBox / GuestInfo / OS / Product” VirtualBox命令行管理界面版本5.1.22 （C）2005-2017甲骨文公司 版权所有。 价值：Windows Vista商业版

要从guest虚拟机添加或更改guest虚拟机属性，请使用该工具 VBoxControl。此工具包含在VirtualBox 2.2或更高版本的Guest添加。从Linux客户端启动时，出于安全考虑，此工具需要root权限

$ sudo VBoxControl guestproperty枚举 VirtualBox客户添加命令行管理界面版本5.1.22 （C）2009-2017甲骨文公司 版权所有。 名称：/ VirtualBox / GuestInfo / OS / Release，value：2.6.28-18-generic， 时间戳：1265813265835667000，flags：<NULL> 名称：/ VirtualBox / GuestInfo / OS / Version，值：＃59-Ubuntu SMP Thu Jan 28 01:23:03 UTC 2010， 时间戳：1265813265836305000，标志：<NULL> ...

对于更复杂的需求，您可以使用VirtualBox编程接口; 请参阅[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。

**4.8。客人控制**

从版本3.2开始，VirtualBox的Guest添加允许从主机系统启动VM内的应用程序。

为了这个工作，应用程序需要安装在客户端内; 主机上不需要安装其他软件。此外，文本模式输出（到stdout和stderr）可以显示在主机上进行进一步处理以及用于指定用户凭据和超时值（以毫秒为单位）的选项来限制应用程序运行的时间。

此功能可用于自动部署客户端内的软件。

从版本4.0开始，Windows的Guest Additions允许自动更新（只有已安装的Guest Additions 4.0或更高版本）。此外，还可以从主机复制到guest虚拟机以及远程创建guest虚拟机目录。

要使用这些功能，请使用VirtualBox命令行，请参见[第8.34节“VBoxManage guestcontrol”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-guestcontrol)。

**4.9。记忆过量**

在具有许多VM的服务器环境中 Guest Additions可用于在多个虚拟机之间共享物理主机内存，从而减少VM使用的内存总量。如果内存使用是限制因素，并且CPU资源仍然可用，这可以帮助在每个主机上打包更多的虚拟机。

**4.9.1。内存膨胀**

从版本3.2开始，VirtualBox的Guest添加可以更改VM在主机运行时使用的主机内存量。由于如何实现，这个功能被称为“内存膨胀”。

**注意**

VirtualBox仅在64位主机上支持内存扩充，Mac OS X主机不支持内存扩充功能。

**注意**

内存膨胀不能启用大页面。要关闭大页面支持虚拟机，请运行 VBoxManage modifyvm <VM name> --largepages off

通常，要更改分配给虚拟机的内存量，必须完全关闭虚拟机并修改其设置。随着内存膨胀，分配给虚拟机的内存可以被给予另一个虚拟机，而不必关闭机器。

当请求内存膨胀时，VirtualBox Guest Additions（在客户端中运行）会从内核级别的客户机操作系统分配物理内存，并将该内存锁定在guest虚拟机中。这样可以确保guest虚拟机不再使用该内存：客户端应用程序不能分配它，而客户端内核也不会使用它。VirtualBox然后可以重新使用这个内存，并将其提供给另一个虚拟机。

通过气泡机制可用的内存仅可供VirtualBox重复使用。它*不会* 作为可用内存返回主机。因此，从正在运行的guest虚拟机请求气球存储器将不会增加主机上的空闲，未分配内存量。因此，内存膨胀是多台虚拟机运行时的内存过度配置机制。这对于暂时启动另一台机器，或在更复杂的环境中，可以根据客户端使用的内存如何并行运行的许多虚拟机进行复杂的内存管理。

此时，通过VBoxManage支持内存扩充。使用以下命令增加或减少运行的虚拟机中安装了Guest Additions的内存气球的大小：

VBoxManage controlvm“VM名称”guestmemoryballoon <n>

"VM name"相关虚拟机的名称或UUID 在哪里， 是 <n>从客户机分配的内存量（以兆字节为单位）。有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm)请参见[第8.13节“VBoxManage controlvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm)。

您还可以设置默认气球，它将在启动后使用以下命令从VM自动请求：

VBoxManage modifyvm“VM名称”--guestmemoryballoon <n>

默认情况下，没有分配气球内存。这是与其他modifyvm设置一样的VM设置，因此只能在机器关闭时进行设置; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

**4.9.2。页面融合**

而内存扩张只是减少了可用于VM的RAM的数量，Page Fusion的工作方式不同：它避免了几个类似运行的虚拟机之间的内存重复。

在同一主机上运行多个类似虚拟机（例如，具有相同操作系统）的服务器环境中，大量内存页是相同的。VirtualBox 3.2版本介绍的VirtualBox页面融合技术是一种新颖的技术，可以有效识别这些相同的内存页面，并在多个虚拟机之间进行共享。

**注意**

VirtualBox仅在64位主机上支持页面融合，Mac OS X主机不支持页面融合。页面融合目前仅适用于Windows客户端（2000及更高版本）。

给定主机上的VM越相似，则Page Fusion可以降低正在使用的主机内存量。因此，如果主机上的所有虚拟机运行相同的操作系统（例如Windows XP Service Pack 2），它将会发挥最佳效果。而不是在每个虚拟机中拥有每个操作系统的完整副本，Page Fusion识别这些操作系统使用的相同内存页面，并消除了重复的内容，在多台计算机之间共享主机内存（“重复数据删除”）。如果VM尝试修改已经与其他VM共享的页面，则会为该虚拟机再次分配一个新页面，其中包含共享页面的副本（“写入时复制”）。所有这些对于虚拟机是完全透明的。

您可能会熟悉其他管理程序产品的这种内存过量配置，称为“页面共享”或“相同页面合并”功能。然而，Page Fusion与其他解决方案显着不同，其解决方案有几个缺点：

1. 传统管理程序扫描*所有*客户机内存，并为每个单个内存页面计算校验和（哈希值）。然后，他们查找具有相同散列的页面，并比较这些页面的整个内容; 如果两个页面产生相同的散列，那么这些页面的内容很可能是相同的。这当然可以花费相当长的时间，特别是如果系统不空闲的话。因此，额外的内存仅在大量时间（这可能是几个小时甚至几天！）之后才可用。更糟糕的是，这种页面共享算法通常会消耗大量CPU资源，并将虚拟化开销提高10-20％。

VirtualBox中的页面融合使用VirtualBox Guest Additions中的逻辑来快速识别虚拟机之间最可能相同的内存单元。因此，它几乎可以立即实现页面共享的大部分可能节省，几乎没有开销。

1. 页面融合也不太可能被相同的内存所困惑，它将消除只是为了学习秒钟，现在内存将会发生变化，并且必须执行非常昂贵的，经常是服务中断的重新分配。

此时，页面融合只能通过VBoxManage进​​行控制，只能在虚拟机关闭时进行。要为VM启用页面融合，请使用以下命令：

VBoxManage modifyvm“VM名称”--pagefusion on

您可以使用某些指标来观察Page Fusion操作。 RAM/VMM/Shared显示了融合页面的总量，而每个VM度量 Guest/RAM/Usage/Shared将返回给定VM的融合内存量。有关如何查询指标的信息[，](https://www.virtualbox.org/manual/ch08.html#metrics)请参阅[第8.35节“VBoxManage指标”](https://www.virtualbox.org/manual/ch08.html#metrics)。

**注意**

启用页面融合可能会间接增加恶意客户端成功攻击在同一主机上运行的其他虚拟机的机会，请参见[第13.3.4节“潜在的不安全操作”](https://www.virtualbox.org/manual/ch13.html#pot-insecure)。

[[18](https://www.virtualbox.org/manual/ch04.html" \l "idm1829) ]实验WDDM驱动程序添加了VirtualBox 4.1。

[[19](https://www.virtualbox.org/manual/ch04.html" \l "idm2171) ]目前只支持复制数据。移动或链接尚未实现。

[[20](https://www.virtualbox.org/manual/ch04.html" \l "idm2226) ] OpenBox对Windows客户端的支持被添加到VirtualBox 2.1中; 支持Linux和Solaris，后跟VirtualBox 2.2。使用VirtualBox 3.0，为Windows guest虚拟机添加了Direct3D 8/9支持。现在也支持OpenGL 2.0。使用VirtualBox 4.1 Windows Vista和Windows 7客户端添加了Windows Aero主题支持（实验性）

# 第五章 存储虚拟化

由于虚拟机最有希望看到其虚拟计算机内置的硬盘，VirtualBox必须能够作为虚拟硬盘向guest虚拟机呈现“真实”存储。目前有三种实现这一目标的方法：

1. 最常见的是，VirtualBox将在真正的硬盘上使用大型图像文件，并将它们作为虚拟硬盘呈现给guest虚拟机。这在[第5.2节“磁盘映像文件（VDI，VMDK，VHD，HDD）”中有介绍](https://www.virtualbox.org/manual/ch05.html#vdidetails)。
2. 或者，如果您有iSCSI存储服务器，也可以将这样的服务器附加到VirtualBox; 这在[第5.10节“iSCSI服务器”中](https://www.virtualbox.org/manual/ch05.html#storage-iscsi)有描述。
3. 最后，作为高级功能，您可以允许虚拟机直接访问一个主机磁盘; 此高级功能在[第9.9.1节“从guest虚拟机使用原始主机硬盘”中进行了描述](https://www.virtualbox.org/manual/ch09.html#rawdisk)。

每个这样的虚拟存储设备（映像文件，iSCSI目标或物理硬盘）将需要连接到VirtualBox向虚拟机呈现的虚拟硬盘控制器。这将在下一节中进行说明。

**5.1。硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe**

在真实的PC中，硬盘和CD / DVD驱动器连接到一个称为硬盘控制器的设备，这些设备可以驱动硬盘操作和数据传输。VirtualBox可以模拟当今PC中通常存在的五种最常见类型的硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，基于USB和NVMe的大容量存储设备。[[21](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2391) ]

* **IDE（ATA）**控制器是IBM PC / AT（1984）中磁盘控制器的向后兼容但非常高级的扩展。最初，此界面仅与硬盘一起工作，但后来也被扩展到还支持CD-ROM驱动器和其他类型的可移动介质。在物理PC中，本标准使用带40或80线的扁平带状平行电缆。每个这样的电缆可以将两个设备连接到传统上被称为“主”和“从”的控制器。典型的PC有两个连接器用于这种电缆; 因此，最多支持多达四个IDE设备。

在VirtualBox中，每个虚拟机可能启用了一个IDE控制器，最多可以为您提供四个可以附加到机器的虚拟存储设备。（默认情况下，这四个中的一个 - 辅助主机 - 被预配置为机器的虚拟CD / DVD驱动器，但可以更改。[[22](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2398) ]）

因此，即使您的客户机操作系统不支持SCSI或SATA设备，也应该始终能够看到IDE控制器。

您还可以选择要向虚拟机（PIIX3，PIIX4或ICH6）提供的IDE控制器硬件VirtualBox的确切类型。这在性能方面没有任何区别，但是如果从另一个虚拟化产品导入虚拟机，则该机器中的操作系统可能会期望特定的控制器类型，如果没有找到则会崩溃。

在使用图形用户界面的“新建虚拟机”向导创建新的虚拟机之后，通常会在机器的“存储”设置中看到一个IDE控制器，虚拟CD / DVD驱动器将附加到其中一个该控制器的四个端口。

* **串行ATA（SATA）**是2003年推出的新标准。与IDE相比，它支持更高的速度和更多的控制器设备。而且，通过物理硬件，可以在系统运行时添加和删除设备。SATA控制器的标准接口称为高级主机控制器接口（**AHCI**）。

像一个真正的SATA控制器，VirtualBox的虚拟SATA控制器运行速度更快，并且比虚拟IDE控制器消耗更少的CPU资源。此外，与VirtualBox IDE控制器（已连接DVD驱动器）一样，您可以将最多30个虚拟硬盘连接到一个机器，而不是三个虚拟硬盘。

因此，从版本3.2开始，根据所选的客户机操作系统，VirtualBox将使用SATA作为新创建的虚拟机的默认值。默认情况下会创建一个虚拟SATA控制器，并使用新VM创建的默认磁盘连接到此控制器。

**警告**

整个SATA控制器和连接到它的虚拟磁盘（包括那些在IDE兼容模式下）将不会被没有设备支持AHCI的操作系统所看到。特别是，**在Windows Vista之前，Windows不支持AHCI**，因此Windows XP（甚至SP3）将不会看到此类磁盘，除非您安装其他驱动程序。通过安装SATA驱动程序并在VM设置对话框中更改控制器类型，可以在安装后从IDE切换到SATA。[[23](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2412) ]

要将SATA控制器添加到默认情况下尚未启用的计算机（由于是由早期版本的VirtualBox创建的，或者由于所选客户机操作系统默认不支持SATA），请转到“存储“页面，点击”存储树“框中的”添加控制器“按钮，然后选择”添加SATA控制器“。此后，附加控制器将作为虚拟机中的单独PCI设备显示，您可以向其中添加虚拟磁盘。

要更改SATA控制器的IDE兼容模式设置，请参见[第8.19节“VBoxManage storagectl”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storagectl)。

* **SCSI**是另一个成熟的行业标准，代表“小型计算机系统接口”。SCSI早在1986年就被标准化为通用接口，用于各种设备（包括存储设备）之间的数据传输。今天SCSI仍然用于连接硬盘和磁带设备，但它大多在商品硬件中被替换。它仍然在高性能工作站和服务器中常用。

主要用于与其他虚拟化软件兼容，VirtualBox可选择支持LSI Logic和BusLogic SCSI控制器，每个最多可连接15个虚拟硬盘。

要启用SCSI控制器，请在虚拟机设置对话框的“存储”页面上单击“存储树”框下的“添加控制器”按钮，然后选择“添加SCSI控制器”。之后，附加控制器将在虚拟机中显示为单独的PCI设备。

**警告**

与其他控制器类型一样，SCSI控制器只能由具有设备支持的操作系统看到。Windows 2003及更高版本附带LSI Logic控制器的驱动程序，而Windows NT 4.0和Windows 2000附带BusLogic控制器的驱动程序。Windows XP附带驱动程序。

* **串行连接SCSI（SAS）**是使用SCSI命令集的另一个总线标准。然而，与SCSI相反，使用物理设备，使用串行电缆代替并行电缆，这简化了物理设备连接。因此，在某些方面，SAS是SCSI到SATA的IDE，它允许更可靠和更快速的连接。

为了支持需要SAS控制器的高端客户机，VirtualBox模拟了一个LSI Logic SAS控制器，可以像SCSI控制器那样启用。此时，最多可以将八个设备连接到SAS控制器。

**警告**

与SATA一样，SAS控制器只能在具有设备支持的操作系统中看到。特别是， **在Windows Vista之前Windows不支持SAS**，因此Windows XP（甚至SP3）将不会看到此类磁盘，除非您安装其他驱动程序。

* 的**USB大容量存储设备** 是通过USB到如硬盘或闪存驱动器的外部存储设备连接到主机的一个标准。所有主要的操作系统长期以来都支持这些设备，并发送通用驱动程序，使第三方驱动程序多余。特别是不支持SATA控制器的传统操作系统可能会受益于USB大容量存储设备。

VirtualBox提供的虚拟USB存储控制器与其他存储控制器类型不同：当存储控制器作为单个PCI设备显示给具有多个磁盘的客户端时，基于USB的存储控制器不会显示为虚拟存储控制器。连接到控制器的每个磁盘将作为专用USB设备显示给访客。

**警告**

通过USB连接的驱动器启动时，使用EFI作为BIOS缺少USB支持。

* **非易失性存储器快速（NVMe）** 是2011年出现的一种非常新的标准，它直接通过PCI express连接非易失性存储器（NVM），以提升以前使用的SSD协议对SSD的带宽限制。与其他标准不同，命令集非常简单，实现最大吞吐量，与ATA或SCSI不兼容。操作系统需要支持NVMe设备才能使用它们。例如Windows 8.1添加了本机NVMe支持，对于Windows 7本机支持，添加了更新。 [[24](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2441) ]

**警告**

仅当使用EFI作为BIOS缺少适当的驱动程序时，才支持通过NVMe连接的驱动器进行引导。

总而言之，VirtualBox为您提供以下类别的虚拟存储插槽：

1. 四个插槽连接到传统IDE控制器，它们始终存在（其中一个通常是虚拟CD / DVD驱动器）;
2. 如果客户机操作系统启用和支持，则连接到SATA控制器的30个插槽;
3. 15个插槽连接到SCSI控制器，如果启用并由客户机操作系统支持;
4. 八个插槽连接到SAS控制器，如果客户机操作系统启用和支持。
5. 八个插槽连接到虚拟USB控制器，如果由客户机操作系统启用和支持。
6. 如果由客户机操作系统启用和支持，则连接到NVMe控制器的最多255个插槽。

鉴于这种大量的存储控制器，您可能会问自己选择哪一个。一般来说，您应该避免使用IDE，除非它是您的客人所支持的唯一控制器。无论您使用SATA，SCSI还是SAS，都没有任何实际的区别。控制器的种类仅供VirtualBox用于与现有硬件和其他管理程序兼容。

**5.2。磁盘映像文件（VDI，VMDK，VHD，HDD）**

磁盘映像文件驻留在主机系统上，并由guest虚拟机将其视为特定几何体的硬盘。当客户机操作系统读取或写入硬盘时，VirtualBox将请求重定向到图像文件。

像物理磁盘一样，虚拟磁盘具有大小（容量），必须在创建映像文件时指定。而与物理磁盘相反，VirtualBox允许您在创建后扩展映像文件，即使它已经有数据; 有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)请参见[第8.23节“VBoxManage modifymedium”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)。[[25](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2465) ]

VirtualBox支持磁盘映像文件的四种变体：

* 通常，VirtualBox使用自己的容器格式的客制硬盘 - 虚拟磁盘映像（VDI）文件。特别地，当您使用新磁盘创建新的虚拟机时，将使用此格式。
* VirtualBox还完全支持许多其他虚拟化产品（特别是由VMware）使用的受欢迎和开放的VMDK容器格式。[[26](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2473) ]
* VirtualBox还完全支持Microsoft使用的VHD格式。
* 还支持Parallels版本2（HDD格式）的映像文件。[[27](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2479) ]由于缺少格式的文档，不支持较新的格式（3和4）。然而，您可以使用Parallels提供的工具将此类图像文件转换为版本2格式。

无论磁盘容量和格式如[第1.7节“创建您的第一个虚拟机”中所述](https://www.virtualbox.org/manual/ch01.html#gui-createvm)，无论磁盘容量和格式是如何创建磁盘映像的两个选项：固定大小或动态分配。

* 如果创建**固定大小的映像**，将在主机系统上创建与虚拟磁盘容量大致相同大小的映像文件。因此，对于10G磁盘，您将拥有一个10G的文件。请注意，根据图像的大小和硬盘的写入性能，创建固定大小的映像可能需要很长时间。
* 为了更灵活的存储管理，请使用**动态分配的映像**。这将首先非常小，不占用未使用的虚拟磁盘扇区的空间，但是在磁盘扇区首次写入时将会增长，直到驱动器达到驱动器创建时选择的最大容量。虽然此格式最初占用较少的空间，但VirtualBox需要扩展映像文件的事实会消耗额外的计算资源，因此，直到磁盘文件大小稳定后，写入操作可能比使用固定大小的磁盘更慢。然而，一段时间后，增长速度将会放缓，写作操作的平均罚款将可以忽略不计。

**5.3。虚拟媒体管理器**

VirtualBox跟踪虚拟机正在使用的所有硬盘，CD / DVD-ROM和软盘映像。这些通常被称为“已知媒体”，来自两个来源：

* 目前所有媒体都附加到虚拟机上;
* “注册”媒体兼容于4.0版之前的VirtualBox版本。有关媒体注册如何更改版本4.0的详细信息，请参阅[第10.1节“VirtualBox存储其文件的位置”](https://www.virtualbox.org/manual/ch10.html#vboxconfigdata)。

可以在**虚拟媒体管理器中**查看和更改已知媒体，您可以从VirtualBox主窗口中的“文件”菜单访问它们。

|  |
| --- |
| https://www.virtualbox.org/manual/images/virtual-disk-manager.png |

已知的媒体可以方便地分组在三个可选格式的三个选项卡中。这些格式是：

* VirtualBox自己的虚拟磁盘映像（VDI）格式或上一章列出的第三方格式的硬盘映像;
* 标准ISO格式的CD / DVD图像;
* 标准RAW格式的软盘映像。

如上图截图所示，对于每个图像，Virtual Media Manager会显示图像文件和其他信息的完整路径，例如图像当前附加到的虚拟机（如果有的话）。

虚拟媒体管理器允许您

* **除去**从注册表的图像（和当这样做时选择性地删除的图像文件）;
* **“释放”**图像，即将其从虚拟机分离，如果它当前作为虚拟硬盘连接到一个虚拟机。
* **复制**一个虚拟硬盘，到另一个目标类型可以不同，选项是 - VDI，VHD或VMDK。
* **修改**磁盘映像文件的属性 - 可用选项有：正常，不可变，写通，可共享，多连接。
* **刷新**当前在窗口中选择的磁盘映像的显示属性的值。

通过选择窗口顶部显示的选项，或右键单击介质并从下拉菜单中显示的选项中选择介质，可以访问这些命令。

从版本4.0开始，要**创建新的磁盘映像，**请使用虚拟机设置对话框中的“存储”页面，因为磁盘映像现在默认存储在每台机器自己的文件夹中。

硬盘映像文件可以复制到其他主机系统上并导入到虚拟机中，尽管某些客户系统（特别是Windows 2000和XP）将要求以与旧版本相似的方式设置新的虚拟机。

**注意**

不要简单地复制虚拟磁盘映像。如果将这样的第二个副本导入到虚拟机中，VirtualBox会抱怨出错，因为VirtualBox会为每个磁盘映像分配一个唯一的标识符（UUID），以确保它只被使用一次。有关此事项的说明[，](https://www.virtualbox.org/manual/ch05.html#cloningvdis)请参见[第5.6节“克隆磁盘映像”](https://www.virtualbox.org/manual/ch05.html#cloningvdis)。此外，如果要将虚拟机复制到另一个系统，VirtualBox具有可能更适合您需要的导入/导出工具; 请参见 [第1.14节“导入和导出虚拟机”](https://www.virtualbox.org/manual/ch01.html#ovf)。

**5.4。特殊图像写入模式**

对于VirtualBox支持的每个虚拟磁盘映像，您可以分别确定如何受虚拟机和快照操作的写操作影响。这适用于所有上述图像格式（VDI，VMDK，VHD或HDD），而不管图像是固定大小还是动态分配。

默认情况下，图像处于“正常”模式。要使用以下列出的非标准模式之一来标记现有图像，请使用 VBoxManage modifyhd; 见[第8.23节“VBoxManage modifymedium”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)。或者，使用VBoxManage将图像附加到VM并使用--mtype 参数; 请参见[第8.18节“VBoxManage storageattach”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storageattach)。

1. 对于**正常的图像**（默认设置），客户端可以读取和写入磁盘的方式没有限制。

当您按照[第1.10节“快照”中](https://www.virtualbox.org/manual/ch01.html#snapshots)所述拍摄虚拟机的 [快照时](https://www.virtualbox.org/manual/ch01.html#snapshots)，这种“正常硬盘”的状态将与快照一起记录，并且当还原到快照时，其状态将完全重置。

（技术上严格来说，图像文件本身不是“重置”，而是在拍摄快照时，VirtualBox“冻结”图像文件，不再写入，对于VM的写操作，第二个“创建差异“图像文件，仅接收对原始图像的更改;有关详细信息，请参阅下一节。）

虽然可以将相同的“正常”映像附加到多个虚拟机，但是只能连接到同一个映像文件的这些虚拟机中的一个可以同时执行，否则如果多个机器写入同一个映像文件，则会出现冲突。[[28](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2554) ]

1. 相比之下，**直写硬盘**完全不受快照的影响：拍摄快照时*不会*保存其状态，并且在还原快照时不恢复。
2. **可共享硬盘**是直写**硬盘**的一种变体。原则上，它们的行为完全相同，即在拍摄快照时，它们的状态*不会*保存，并且在恢复快照时不恢复。只有将这些磁盘附加到多个虚拟机时，差异才会显示出来。可共享磁盘可能会连接到可能同时运行的多个虚拟机。这使得它们适合于VM和类似应用程序之间的集群文件系统使用，这些应用程序是明确准备同时访问磁盘的。只能以这种方式使用固定大小的图像，并且动态分配的图像被拒绝。

**警告**

这是一个专家功能，滥用可能会导致数据丢失 - 常规文件系统未准备好处理多方同时进行的更改。

1. 接下来，**不可变图像**只能在虚拟机运行时暂时写入访问; 当虚拟机下次启动时，所有更改都将丢失。因此，与“正常”图像相反，相同的不可变图像可以与几个虚拟机无缝使用。

*创建*一个不可变的图像没有任何意义，因为它将最初是空的，并且在每台机器重新启动时丢失内容（除非你真的希望有一台在机器启动时总是未格式化的磁盘）。因此，通常，您将首先创建一个“正常”图像，然后，当您认为其内容有用时，稍后将其标记为不可变。

如果您拍摄具有不可变图像的机器的快照，则在每台机器上电时，这些图像将重置为最后（当前）快照的状态（而不是原始不可变图像的状态）。

**注意**

作为一个特殊的例外，不可变图像如果连接到保存状态的计算机或在计算机运行时（最后一个快照）（所谓的“在线”快照）），则 *不会*重置。因此，如果机器的当前快照是这样的“在线”快照，其不可变图像的行为与前面描述的“正常”图像完全相同。要重新启用此类图像的自动复位，请删除机器的当前快照。

在技​​术上，VirtualBox从来没有直接写入不可变图像。来自机器的所有写入操作将被引导到差分图像; VM下次启动时，差分图像将被复位，以便每次VM启动时，其不可变图像都具有完全相同的内容。[[29](https://www.virtualbox.org/manual/ch05.html" \l "ftn.idm2577) ]差异化图像仅在机器从VirtualBox内通电时重新启动，而不是通过从机器中请求重新启动而重新启动。这也是为什么不变图像的行为如上所述当快照也存在时，也使用差异图像。

如果在虚拟机启动时自动丢弃差分映像不符合您的需要，则可以使用以下autoreset参数 将其关闭 VBoxManage modifyhd：有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)请参见[第8.23节“VBoxManage modifymedium”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvdi)。

1. 即使这些机器同时运行，**多重模式下**的图像可以同时连接到多个虚拟机。对于连接此类图像的每个虚拟机，都会创建差异图像。结果，由一台机器写入这样的虚拟磁盘的数据不被附加图像的其他机器看到; 每个机器创建自己的多重图像的写入历史记录。

在技​​术上，“multiattach”图像的行为与“不可变”图像相同，除了差分图像在每次机器启动时都不会重置。

此模式对于共享几乎从未写入的文件（例如图片库）很有用，每个客户机只更改少量数据，大多数磁盘内容保持不变。修改的块被存储在保持相对小的差分图像中，并且共享内容仅在主机存储一次。

1. 最后，**只读映像**自动用于CD / DVD映像，因为CD / DVD不能写入。

说明各种类型与快照之间的差异：假设您已经在虚拟机中安装了客户机操作系统，并且已经拍摄了快照。想象一下，您已经意外地感染了您的虚拟机病毒，并希望返回快照。使用正常的硬盘映像，您只需恢复快照，并且硬盘映像的早期状态也将被还原（并且您的病毒感染将被撤消）。使用不可变硬盘，只需关闭并打开虚拟机电源，病毒感染将被丢弃。然而，使用直写映像，您不能通过虚拟化轻松地撤消病毒感染，但必须像虚拟机一样对虚拟机进行消毒。

但是，如果您希望保留关键数据而不考虑快照，您可能会发现直写图像很有用，并且由于可以将多个映像附加到虚拟机，因此您可能希望有一个不可变的操作系统和一个写入为您的数据文件。

**5.5。差异图像**

上一节暗示了区分图像以及如何使用快照，不可变图像和多个磁盘附件。对于好奇的VirtualBox用户，本节将更详细地介绍它们的工作原理。

差分图像是一个特殊的磁盘映像，它只保留与另一个映像的差异。差异图像本身是无用的，它必须总是引用另一个图像。差分图像通常被称为“小孩”，它与“父”（parent）保持差异。

当差分图像处于活动状态时，它将从虚拟机而不是其父代接收所有写入操作。差分图像仅包含自创建差分图像以来已更改的虚拟硬盘的扇区。当机器从这样的虚拟硬盘读取扇区时，首先查看差分图像。如果该部门存在，则从那里返回; 如果没有，VirtualBox会查看父级。换句话说，父母变成“只读”; 它永远不会写入，但是如果某个扇区没有改变，则会被读取。

差异图像可以链接。如果为已经具有差异图像的虚拟磁盘创建了另一个差异图像，则它将成为原始父项的“孙子”。然后，第一个差分图像也变为只读，并且写入操作仅转到第二级差分图像。当从虚拟磁盘读取数据时，VirtualBox需要首先查看第二个差分图像，然后再进入第一个，如果没有找到扇区，然后再进入原始图像。

可以有无限数量的差异图像，每个图像可以有多个孩子。因此，根据您的机器配置的复杂程度，差异图像可以与父母，“兄弟姐妹”和孩子形成一个复杂的树。写操作总是转到附加到机器上的一个“活动”差分图像，并且对于读操作，VirtualBox可能需要查找链中的所有父母，直到找到所讨论的扇区。您可以在虚拟媒体管理器中查看这样的树：

|  |
| --- |
| https://www.virtualbox.org/manual/images/virtual-disk-manager2.png |

在所有这些情况下，从虚拟机的角度来看，虚拟硬盘的行为与任何其他磁盘一样。虚拟机正在运行时，由于VirtualBox可能需要多次查找扇区，因此存在轻微的运行时I / O开销。然而，这并不明显，因为具有扇区信息的表始终保存在内存中并且可以快速查找。

差异图像在以​​下情况下使用：

1. **快照。**当您创建快照时，如上一节所述，VirtualBox“冻结”附加到虚拟机的图像，并为每个图像创建差异图像（确切地说，每个图像不是“直写”模式）。从虚拟机的角度来看，虚拟磁盘继续运行，但所有的写操作都进入差分映像。每次创建另一个快照时，对于每个硬盘附件，将创建并附加另一个差异图像，形成链或树。

在上述屏幕截图中，您将看到原始磁盘映像现在附加到快照，表示拍摄快照时磁盘的状态。

如果现在**还原**一个快照 - 也就是说，如果要回到快照中存储的确切机器状态，则会发生以下情况：

* 1. VirtualBox将复制到快照中的虚拟机设置复制回虚拟机。因此，如果从拍摄快照后对机器配置进行了更改，则它们将被撤销。
  2. 如果在机器运行时拍摄快照，它将包含已保存的机器状态，并且还恢复该状态; 恢复快照后，机器将处于“已保存”状态，并在下次启动时从那里继续执行。否则机器将处于“关机”状态，并进行全面启动。
  3. 对于连接到机器的每个磁盘映像，保存自拍摄当前快照以来所有写入操作的差分图像被丢弃，原始父图像将再次生效。（如果恢复了“根”快照，那么这将是每个附件的根磁盘映像;否则，其他一些差异映像将从它下降。）这有效地恢复了旧的机器状态。

如果稍后**删除**快照以释放磁盘空间，则对于每个磁盘附件，其中一个差异映像将变得过时。在这种情况下，不能简单地删除磁盘附件的差分图像。相反，VirtualBox需要查看差异图像的每个部分，并需要将其复制回其父代; 这被称为“合并”图像，并且可以是可能冗长的过程，这取决于差分图像的大小。在合并操作过时的差异化图像被删除之前，也可能暂时需要大量额外的磁盘空间。

1. **不变的图像。**当图像切换到“不可变”模式时，也会创建差分图像。与快照一样，父图像然后变为只读，差分图像接收所有写入操作。每次启动虚拟机时，所有连接到其上的不可变图像都会丢弃各自的差异图像，每次重新启动时都可以有效地重置虚拟机的虚拟磁盘。

**5.6。克隆磁盘映像**

您可以在同一主机上复制硬盘映像文件，以便快速生成具有相同操作系统设置的第二个虚拟机。但是，您*只*应使用VirtualBox附带的实用程序制作虚拟磁盘映像副本; 见[第8.24节“VBoxManage clonemedium”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-clonevdi)。这是因为VirtualBox为每个磁盘映像分配一个唯一的标识号（UUID），它也存储在映像内，VirtualBox将拒绝使用两个使用相同数字的映像。如果您不小心尝试重新导入正常复制的磁盘映像，则可以使用VirtualBox的实用程序进行第二个副本，并导入。

请注意，较新的Linux发行版从驱动器的ID标识引导硬盘。虚拟磁盘映像的UUID确定驱动器的ID VirtualBox报告。因此，如果克隆磁盘映像并尝试引导复制的映像，则guest虚拟机可能无法在UUID更改时确定其自己的引导磁盘。在这种情况下，您必须修改引导加载程序脚本中的磁盘ID（例如 /boot/grub/menu.lst）。磁盘ID如下所示：

SCSI-SATA\_VBOX\_HARDDISK\_VB5cfdb1e2-c251e503

复制图像的ID可以用

hdparm -i / dev / sda

**5.7。主机I / O缓存**

从版本3.2开始，VirtualBox可以选择禁用主机操作系统在磁盘映像文件上执行的I / O缓存。

传统上，VirtualBox已将磁盘映像文件作为普通文件打开，导致主机操作系统像任何其他文件一样缓存。其主要优点是速度：当客户操作系统写入磁盘并且主机操作系统缓存使用延迟写入时，写操作可以快速报告给客户操作系统，同时主机操作系统可以异步执行操作。此外，当您第二次启动VM并且有足够的内存可用于操作系统以用于缓存时，虚拟磁盘的大部分可能位于系统内存中，并且VM可以更快地访问数据。

请注意，这仅适用于图像文件; 驻留在远程iSCSI存储上的虚拟磁盘从不发生缓冲，这是企业级设置中更常见的情况（请参见[第5.10节“iSCSI服务器”](https://www.virtualbox.org/manual/ch05.html#storage-iscsi)）。

虽然缓冲是虚拟桌面计算机上几台机器的有用的默认设置，但这种方法有一些缺点：

1. 通过主机操作系统缓存的延迟写入不太安全。客户操作系统写入数据时，即使尚未到达物理磁盘，也会考虑写入的数据。如果由于某种原因写入不会发生（电源故障，主机崩溃），数据丢失的可能性增加。
2. 磁盘映像文件往往非常大。因此，缓存它们可以快速耗尽整个主机操作系统缓存。取决于主机操作系统缓存的效率，这可能会严重影响主机，尤其是在同一时间运行多个虚拟机时。例如，在Linux主机上，主机缓存可能导致Linux延迟所有写入，直到主机缓存几乎完整，然后立即写出所有这些更改，可能会使VM执行停顿几分钟。当I / O请求超时时，这可能会导致客户端的I / O错误。
3. 物理内存通常由于客户机操作系统通常具有自己的I / O高速缓存而浪费，这可能导致数据被缓存两次（在客户端和主机缓存中），效果不大。

如果由于上述原因决定禁用主机I / O缓存，VirtualBox会使用自己的小型缓存来缓冲写入，但没有读取高速缓存，因为这通常由访客操作系统执行。此外，VirtualBox通过多个I / O线程完全支持其虚拟SATA，SCSI和SAS控制器的异步I / O。

由于IDE控制器不支持异步I / O，出于性能原因，您可能希望为VM的虚拟IDE控制器启用主机缓存。

因此，VirtualBox允许您配置主机I / O缓存是否分别用于每个I / O控制器。请勿在给定的虚拟存储控制器的“存储”设置中选中“使用主机I / O缓存”框，或者使用以下VBoxManage命令禁用虚拟存储控制器的主机I / O缓存：

VBoxManage storagectl“VM名称”--name <controllername> --hostiocache关闭

有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storagectl)请参见[第8.19节“VBoxManage storagectl”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storagectl)。

由于上述原因，VirtualBox现在默认使用SATA控制器进行新的虚拟机。

**5.8。限制磁盘映像的带宽**

从4.0版开始，VirtualBox允许限制用于异步I / O的最大带宽。此外，它支持通过多个图像的带宽组共享限制。有可能有一个以上这样的限制。

限制通过配置 VBoxManage。下面的示例创建一个名为“限制”的带宽组，将限制设置为20 MB / s，并将组分配给VM的连接磁盘：

VBoxManage bandwidthctl“VM名称”添加限制 - 类型磁盘 - 限制20M VBoxManage storageattach“VM名称”--storagectl“SATA”--port 0 - 设备0 - 类型hdd --medium disk1.vdi --bandwidthgroup限制 VBoxManage storageattach“VM名称”--storagectl“SATA” - 端口1 - 设备0 - 类型hdd --medium disk2.vdi --bandwidthgroup限制

组中的所有磁盘共享带宽限制，这意味着在上面的示例中，组合的两个图像的带宽永远不会超过20 MB / s。但是，如果一个磁盘不需要带宽，则另一个磁盘可以使用其组中的剩余带宽。

虚拟机正在运行时可以更改每个组的限制，并立即进行更改。以下示例将上述示例中创建的组的限制更改为10 MB / s：

VBoxManage bandwidthctl“VM名称”设置Limit --limit 10M

**5.9。CD / DVD支持**

虚拟CD / DVD驱动器默认情况下仅支持阅读。介质配置在运行时可更改。您可以在三个选项之间选择提供中等数据：

* **主机驱动器**定义客户端可以从主机驱动器中的介质读取。
* **图像文件**（通常是ISO文件）使访客只读访问图像中的数据。
* **空的**代表没有插入介质的驱动器。

在上述之间进行更改或更改主机驱动器中由机器访问的介质或更改图像文件将会向客户机操作系统发出中等更改，从而可以对更改进行响应（例如，启动安装程序）。

客户端可以防止中等的更改，VirtualBox可以通过锁定主机驱动器来反映出来。您可以通过VirtualBox GUI或VBoxManage命令行工具在这种情况下强制进行中等移除。有效地，这相当于许多CD / DVD驱动器提供的紧急弹出，具有所有相关的副作用：客户操作系统可以像真实的硬件一样发出错误消息，客户应用程序可能会不正当行为。请谨慎使用。

**注意**

提供给guest虚拟机的驱动器的标识字符串（在客户端，将由诸如Windows设备管理器之类的配置工具显示）始终为“VBOX CD-ROM”，与虚拟驱动器的当前配置无关。这是为了防止每次更改配置时在客户机操作系统中触发硬件检测。

标准CD / DVD仿真仅允许读取标准数据CD和DVD格式。作为实验功能，对于其他功能，可以通过启用“直通”模式，让客户端直接访问CD / DVD主机驱动器。根据主机硬件，这可能会使三件事情发生：潜在的：

* 如果主机DVD驱动器是CD / DVD刻录机，则从客户端进行CD / DVD写入;
* 播放音频CD;
* 播放加密的DVD。

在GUI对话框中有一个“Passthrough”复选框，用于配置连接到存储控制器的媒体，或者您可以使用该 --passthrough选项 VBoxManage storageattach; 详见[8.18节“VBoxManage storageattach”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storageattach)。

即使启用了传递，不安全的命令（如更新驱动器固件）将被阻止。视频CD格式从不支持，即使在直通模式下，也不能从虚拟机播放。

在Solaris主机上，由于主机执行的安全措施，传递需要运行具有真正root权限的VirtualBox。

**5.10。iSCSI服务器**

iSCSI代表“Internet SCSI”，是允许通过Internet（TCP / IP）连接使用SCSI协议的标准。特别是随着千兆以太网的出现，将iSCSI存储服务器简单地作为远程硬盘连接到计算机网络已成为可负担的。在iSCSI术语中，提供存储资源的服务器称为“iSCSI目标”，而连接到服务器并访问其资源的客户端称为“iSCSI启动器”。

VirtualBox可以透明地将iSCSI远程存储作为虚拟硬盘呈现给虚拟机。客户机操作系统在虚拟磁盘映像（VDI文件）和iSCSI目标之间将看不到任何区别。为了实现这一点，VirtualBox具有集成的iSCSI启动器。

VirtualBox的iSCSI支持已经根据iSCSI标准进行开发，并且应该符合所有符合标准的iSCSI目标。要在VirtualBox中使用iSCSI目标，必须使用命令行; 请参见 [第8.18节“VBoxManage storageattach”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storageattach)。

[[21](https://www.virtualbox.org/manual/ch05.html" \l "idm2391) ] VirtualBox 1.6增加了SATA支持; 增加了2.1的实验性SCSI支持，并完全实现了2.2。通常，VirtualBox 3.1使存储附件变得更加灵活; 见下文。支持LSI Logic SAS控制器添加了VirtualBox 3.2; 自VirtualBox 5.0支持USB大容量存储设备; 添加了VirtualBox 5.1的NVMe控制器支持。

[[22](https://www.virtualbox.org/manual/ch05.html" \l "idm2398) ]将机器的CD / DVD驱动器分配给辅助主机在VirtualBox 3.1之前已修复; 现在可以更改，驱动器可以在IDE控制器的其他插槽中，并且可以有多个此类驱动器。

[[23](https://www.virtualbox.org/manual/ch05.html" \l "idm2412) ] VirtualBox推荐了可从[http://downloadcenter.intel.com/Product\_Filter.aspx?ProductID=2101](http://downloadcenter.intel.com/Product_Filter.aspx?ProductID=2101" \t "_top)下载的Intel Matrix Storage驱动程序。

[[24](https://www.virtualbox.org/manual/ch05.html" \l "idm2441) ] NVMe控制器是扩展包的一部分。

[[25](https://www.virtualbox.org/manual/ch05.html" \l "idm2465) ]使用VirtualBox 4.0添加了图像调整大小。

[[26](https://www.virtualbox.org/manual/ch05.html" \l "idm2473) ] VirtualBox 1.4添加了对VMDK的初始支持; 自2.1版本以来，VirtualBox完全支持VMDK，这意味着您可以创建快照，并使用上面介绍的所有其他高级功能的VDI图像与VMDK。

[[27](https://www.virtualbox.org/manual/ch05.html" \l "idm2479) ] VirtualBox 3.1中添加了支持。

[[28](https://www.virtualbox.org/manual/ch05.html" \l "idm2554) ]现在这个限制比VirtualBox 2.2更宽松了。以前，每个“正常”磁盘映像只能 *附加*到一台机器上。只要只有其中一台机器正在运行，它就可以连接到多台机器上。

[[29](https://www.virtualbox.org/manual/ch05.html" \l "idm2577) ]这个行为也改变了VirtualBox 2.2。以前，机器会话*结束*时，差分图像被丢弃 ; 现在每次机器上电时都会丢弃它们。

# 第六章 网络虚拟化

如[第3.8节“网络设置”中](https://www.virtualbox.org/manual/ch03.html#settings-network)简要提到的，VirtualBox为每个虚拟机提供多达八个虚拟PCI以太网卡。对于每张这样的网卡，你可以单独选择：

1. 虚拟化的网络硬件类型；
2. 虚拟网卡将的网络模式。

可以在VirtualBox图形用户界面中的设置对话框的“网络”部分中配置四个网卡（当然了，要在虚拟机停止后设置）。你可以通过VBoxManage modifyvm在命令行上配置所有8个网卡; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。本章详细地介绍了各种网络设置。

## 6.1、虚拟网络硬件

对于每张虚拟网卡，你可以单独选择将呈现给虚拟机的*硬件*类型 。VirtualBox可以虚拟化以下六种类型的网络硬件：

* AMD PCNet PCI II (Am79C970A);
* AMD PCNet FAST III (Am79C973, the default);
* Intel PRO/1000 MT Desktop (82540EM);
* Intel PRO/1000 T Server (82543GC);
* Intel PRO/1000 MT Server (82545EM);
* Paravirtualized network adapter (virtio-net)（半虚拟化网络适配器）.

PCNet FAST III是默认选择，因为几乎所有操作系统以及GNU GRUB启动管理器都支持。Intel PRO/1000 MT Desktop类型适用于Windows Vista及更高版本。Intel PRO/1000 T Server可被Windows XP客户机系统识别，无需安装额外的驱动程序。MT Server变体便于从其他平台导入OVF这种格式的虚拟机。

"Paravirtualized network adapter (virtio-net)" 网络硬件是特殊的，如果选择这一项，VirtualBox将不会虚拟一般网络硬件（即被一般客户机原始系统所支持）。VirtualBox期望虚拟化环境的特殊软件由客户机系统提供，从而避免了虚拟网络硬件和提高网络性能的复杂性。从版本3.1开始，VirtualBox为业界标准的“virtio”网络驱动程序提供支持，这些驱动程序是开源项目KVM的一部分。

“virtio”网络驱动程序可用于以下客户机操作系统：

* Linux内核版本2.6.25或更高版本可以配置为virtio。
* 对于Windows 2000，XP和Vista，可以从KVM项目网页下载和安装virtio驱动程序。地址如下：[http://www.linux-kvm.org/page/WindowsGuestDrivers](http://www.linux-kvm.org/page/WindowsGuestDrivers" \t "_top)。

VirtualBox还支持**巨型帧**，即具有超过1500字节数据的网络数据包，具体怎么支持可以查阅VirtualBox官方手册，默认情况下，巨型帧是禁用的。

## 6.2、网络模式简介

八个网络适配器中的每一个均可以单独配置为以下模式中的一个：

1、Not attached

有网卡，但是没有连接网线。这种情况类似于服务器没有连接网线的情况，因此也就无法访问网络。一般不会使用这种情况，也没有需要特殊说明的地方。

2、网络地址转换（NAT）

如果你想要浏览网页，下载文件、查看电子邮件等，那么此默认模式对你来说应该就够用了，您可以跳过本节的其余部分内容。请注意，使用Windows文件共享时有一些限制。具体限制接下来会说。

3、NAT网络

NAT网络是VirtualBox 4.3中引入的新NAT风格。6.4节具体介绍

4、桥接网络

这是一种为了满足更高级的网络需求的模式，比如在客户机上模拟一个局域网或者运行服务器。启用后，VirtualBox将连接到你的一个物理网卡，并使用物理网卡直接交换网络数据包，从而绕过主机操作系统的网络栈。

5、内部联网

这可以用于创建不同类型的基于软件的网络，这对于所选择的虚拟机可见，但不适用于在主机或外部世界上运行的应用程序。

6、仅主机网络

这可以用于创建包含主机和一组虚拟机的网络，即：主机和各个虚拟机组成一个网络。而不需要主机的物理网络接口。相反，虚拟网络接口（类似于环回接口）在主机上创建，从而提供虚拟机与主机之间的连接。

7、通用网络

很少使用的模式，目前有两种可用的子模式：

UDP隧道：可以用于通过现有网络基础架构直接轻松、透明地互连运行在不同主机上的虚拟机。

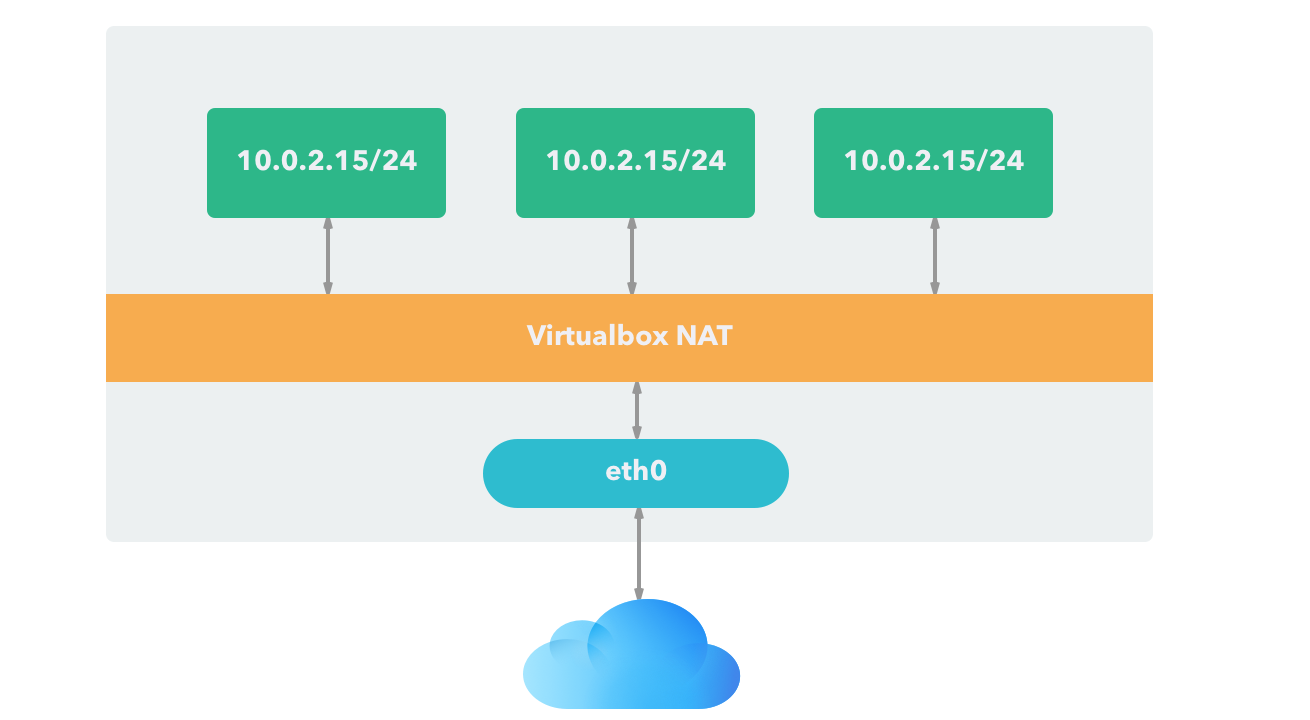
VDE（虚拟分布式以太网）网络：此选项可用于连接到Linux或FreeBSD主机上的虚拟分布式以太网交换机。

目前，这需要从源代码编译VirtualBox，因为Oracle软件包不包含它。

接下来详细介绍可用的网络模式。

## 6.3、网络地址转换（NAT）

网络地址转换（NAT）是从虚拟机访问外部网络最简单的方式。通常，它不需要对主机网络和客户机系统网络做任何配置。它是VirtualBox中的默认网络模式。启用NAT模式的虚拟机与通过路由器连接到Internet的真实计算机非常相似。在这种情况下，“路由器”是VirtualBox网络引擎，会自动转发虚拟的报文。虚拟机发出去的报文，会被 virtualbox 网络引擎拦截，抽取其中的 TCP/IP 数据，然后用主机的网络进行发送，对于外部网络来说，它们看到这个报文是主机发送的。virtualbox 还会自动监听响应报文，然后把目标地址修改成虚拟机的地址，这样虚拟机就能收到应答报文。默认情况下，**虚拟机之间不能相互通信**。NAT模式的缺点在于，就像路由器背后的私有网络一样，**虚拟机是不可见**的，即：不能从外部网络访问，除非设置了端口转发，否则你不能以这种方式在虚拟机上运行服务器。图解如下：**VirtualBox NAT的角色类似一个路由器。**



本图片来自：<http://cizixs.com/2017/03/09/virtualbox-network-mode-explained>

该模式下，虚拟机从集成到VirtualBox的DHCP服务器在专用网络上接收其网络地址和配置。因此分配给虚拟机的IP地址通常与主机的网络完全不同。虚拟机的多张网卡都可以设置为NAT，第一张网卡连接到专用网络10.0.2.0，第二张网卡连接到网络10.0.3.0等。如果由于某些原因需要更改为客户机系统分配的IP范围，请参阅官方手册[第9.11节“微调VirtualBox NAT引擎”](https://www.virtualbox.org/manual/ch09.html)。

我在mac下设置NAT模式之后依然无法上网，这时候，不要急要检查一下Linux网络的配置，当时一看是客户机系统Linux的配置文件配置的不对，即：配置文件为：

cat /etc/sysconfig/network-scripts/ifcfg-enp0s3

........

ONBOOT=no

改为：

ONBOOT=yes

就OK了。

因此遇到问题之后不要慌乱着急，要利用知识来判断问题出在了哪里，然后解决问题。

**因此，没有设置端口转发的情况下：**

**虚拟机与主机的关系：**只能单向访问，虚拟机可以通过网络访问到主机，主机无法通过网络访问到虚拟机。

**虚拟机与网络中其他主机的关系：**只能单向访问，虚拟机可以访问到网络中其他主机，其他主机不能通过网络访问到虚拟机。

**虚拟机与虚拟机的关系：**相互不能访问，虚拟机与虚拟机各自完全独立，相互间无法通过网络访问彼此。

**6.3.1。使用NAT配置端口转发**

由于NAT模式下虚拟机连接到VirtualBox内部的专用网络并且对主机不可见，因此主机或与主机同一网络上的其他计算机无法访问虚拟机上的网络服务。然而，像物理路由器一样，VirtualBox可以通过**端口转发来**使选定的服务对外可见**。**这意味着VirtualBox会监听主机上的某些端口，并将所有到达客户端的数据包重新发送到相同或不同的端口。**对于网络上的主机或其他物理（或虚拟）机器上的应用程序，虚拟机上被代理的服务看起来就像在真实主机上运行一样，因此，要访问虚拟机的话就要访问主机系统**。这也意味着你不能在主机的相同端口上运行相同的服务。

可以使用图形管理工具配置端口转发。在这里，可以将主机端口映射到客户机端口。也可以使用VBoxManage命令行工具。详情请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html)。在设置端口转发时，不必要保证主机和客户机系统的端口是一致的，你可以使用主机上尚未由服务进程使用的任何端口。

您将需要知道服务使用的客户机中的哪些端口，并决定在主机上使用哪些端口（通常但并不总是您希望在guest虚拟机和主机上使用相同的端口）。你可以使用主机上尚未由服务使用的任何端口。例如：

**VBoxManage** modifyvm "VM name" --natpf1 "guestssh,tcp,,2222,,22"

比如这条命令就是给虚拟机添加一条规则，所有发到主机 2222 端口的报文都要转发到虚拟机的 22 端口。--natpf 后面的数字代表这个规则是作用于第一块网卡的，最后一个参数的各个字段含义是：

1. 转发规则的名字。如果没有指定，virtualbox 会自动生成一个名字
2. 报文类型。支持 tcp、udp
3. 主机地址。如果不填写，不论主机哪个网卡收到 2222端口的数据都要转发；如果有填写，则指定某个网卡收到 2222 端口的报文才转发。比如 127.0.0.1,2222 说明只转发本地的报文。
4. 主机端口。监听的主机端口
5. 虚拟机地址。如果虚拟机不是 DHCP server 分配的，需要告诉 virtualbox 虚拟机网卡的 ip 地址
6. 虚拟机端口。虚拟机内部提供服务的端口，报文会转发到这个端口

删除端口转发规则也是类似，区别是提供规则名称就行：

VBoxManage modifyvm "VM name" --natpf1 **delete** "guestssh"

也可以在VM运行时配置NAT，详见[第8.13节“VBoxManage controlvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm)。

**6.3.2。使用NAT启动PXE**

不常用，略去。详细信息，可以参考VirtualBox的官方手册：<https://www.virtualbox.org/manual/ch06.html>。

**6.3.3。NAT限制**

NAT模式有四种**限制**：

1、ICMP协议限制：一些经常使用的网络调试工具（例如 ping或tracerouting）依赖于ICMP协议来发送/接收消息。虽然使用VirtualBox 2.1（ping现在可以工作）改进了对ICMP的支持，但其他一些工具可能仍然无法可靠地工作。

2、接收UDP广播是不可靠的：客户家不能可靠地接收广播。

3、有些协议如GRE不受支持：不支持TCP和UDP以外的协议。

4、转发的主机端口不能小于1024：因为在基于UNIX系统的主机上只有 root 权限才能使用这些端口。容易造成冲突，而且权限问题会导致虚拟机无法启动

## 6.4、网络地址转换服务（NAT网络）

网络地址转换（NAT）服务（即：NAT网络）的工作方式与家用路由器的工作方式类似，虚拟机可以访问外网，外网不可以访问虚拟机（除非配置了端口转发）。你可以认为NAT模式和内网模式的组合，即：NAT服务附带了一个内部网络。当创建NAT服务时，可以选择内部网络的名称，如果内部网络不存在，将创建内部网络。创建NAT网络的示例命令如下：

VBoxManage natnetwork add --netname natnet1 --network "192.168.15.0/24" --enable

其中：“natnet1”是要使用的内部网络的名称，“192.168.15.0/24”是NAT服务接口的网络地址和掩码。默认情况下为静态配置，网关将分配地址192.168.15.1（网络接口地址之后的地址），这可能会发生变化。要将DHCP服务器添加到内部网络，使用如下命令：

VBoxManage natnetwork add --netname natnet1 --network“192.168.15.0/24”--enable --dhcp on

或者在创建后向内部网络添加DHCP服务器：

VBoxManage natnetwork modify --netname natnet1 --dhcp on

要再次禁用它，请使用：

VBoxManage natnetwork modify --netname natnet1 --dhcp off

DHCP服务器提供了已注册的名字服务器的列表，但不会将服务器映射到127/8网络。

要启动NAT服务（NAT网络），请使用以下命令：

VBoxManage natnetwork start --netname natnet1

如果网络连接了DHCP服务器，则它将与NAT网络服务一起启动。

VBoxManage natnetwork stop --netname natnet1

停止NAT网络服务，连同DHCP服务器（如果有的话）。

删除NAT网络服务使用：

VBoxManage natnetwork remove --netname natnet1

如果在内部网络上启用了此命令，则不会删除DHCP服务器。

NAT网络也支持端口转发（对于IPv4 使用--port-forward-4，对于IPv6使用 --port-forward-6）：

VBoxManage natnetwork modify --netname natnet1 --port-forward-4 "ssh:tcp:[]:1022:[192.168.15.5]:22"

这将使得到达主机1022端口的所有流量转发到虚拟机192.168.15.5的22号端口上。其中：主机端口、虚拟机端口和虚拟机IP不可省略。具体的含义和解释，请参考6.3.1节。

要删除转发规则，请使用：

VBoxManage natnetwork modify --netname natnet1 --port-forward-4 delete ssh

也可以将NAT服务（NAT网络）绑定到指定的接口：

VBoxManage setextradata global "NAT/win-nat-test-0/SourceIp4" 192.168.1.185

要查看已经注册的NAT网络列表，请使用：

VboxManage list natnetworks

## 6.5、桥接网络

通过桥接网络，VirtualBox将使用*主机*系统上的设备驱动程序来过滤主机系统网卡的数据。因此，此驱动程序称为“网络过滤器”驱动程序。这种模式下，你把客户机看做一台网络上真正的计算机一样就可以了，客户机系统和主机系统之间通过网桥（又叫二层交换机）连接，**这时候VirtualBox的角色是网桥，使虚拟机与主机建立在同一网络上**。但是真正的上网还是通过主机系统的网卡来访问的，因此客户机的网络要和主机系统的一样，包括网络号、子网掩码、广播地址这些。一般来说选择默认模式就好，虚拟机是通过主机所在网络中的DHCP服务得到ip地址的。典型的配置如下：

**主机：**

en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500

ether 18:65:90:d3:2b:15

inet6 fe80::1c7e:78a8:e0a2:d864%en0 prefixlen 64 secured scopeid 0x4

inet 192.168.0.108 netmask 0xffffff00 broadcast 192.168.0.255

......

**虚拟机：**

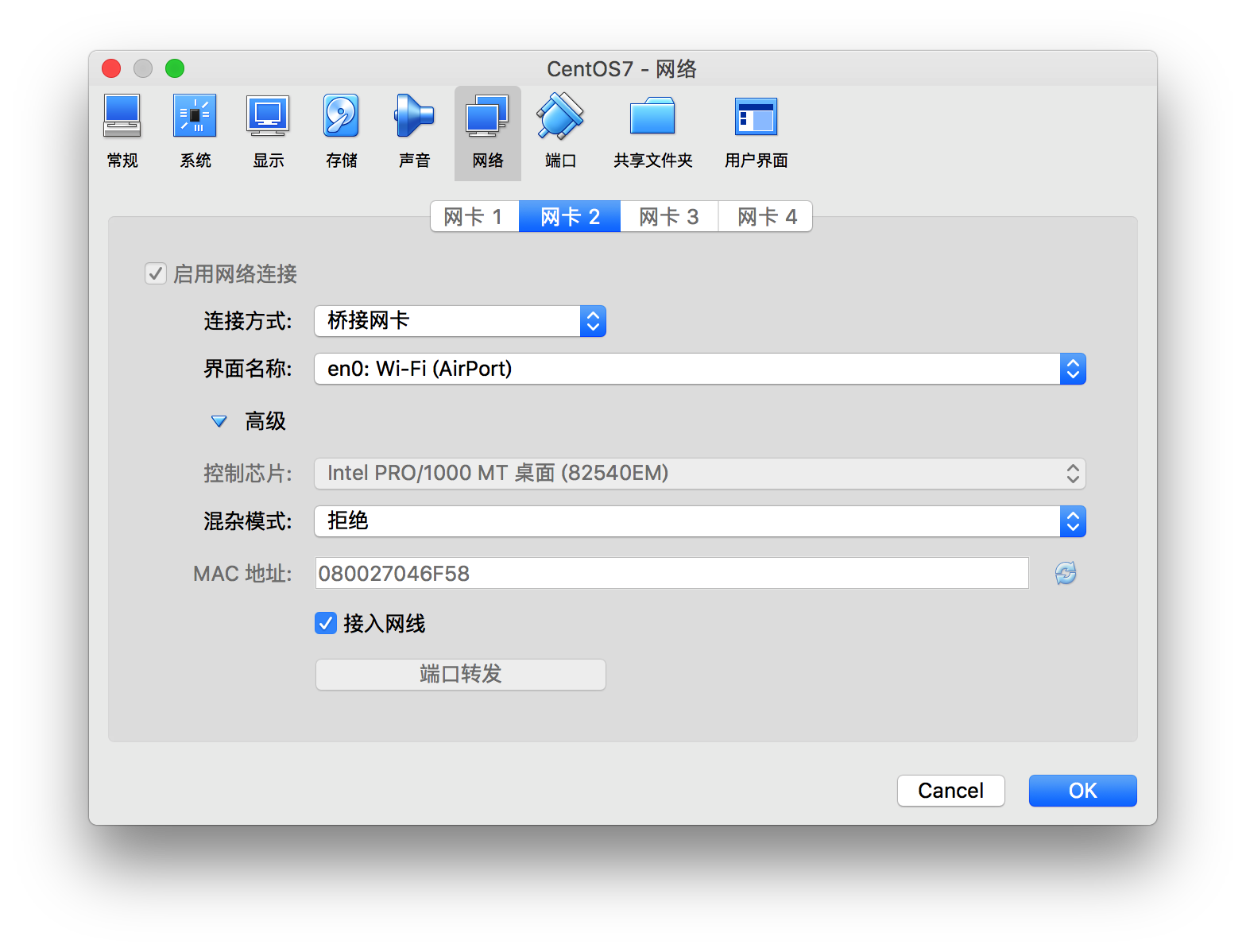
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.0.114 netmask 255.255.255.0 broadcast 192.168.0.255

inet6 fe80::ebaa:fe7a:14ca:24be prefixlen 64 scopeid 0x20<link>

ether 08:00:27:04:6f:58 txqueuelen 1000 (Ethernet)

要启用桥接网络模式，你需要做的就是打开虚拟机的“设置”对话框，进入“网络”页面，并在“链接方式”字段的下拉列表中选择“桥接网络”。最后，从底部的列表中选择所需的主机接口名字，其中包含主机系统的物理网络接口名字。例如，在典型的MacBook上，您可以在“en1：AirPort”（即无线接口）和“en0：Ethernet”（表示与网络电缆的接口）之间进行选择。如下图：



**注意：**

由于大多数无线适配器不支持混杂模式，因此桥接到无线接口的方式与桥接到有线接口不同。

根据你的主机操作系统，应注意以下限制：

* 在**Macintosh**主机上，使用AirPort（Mac的无线网络）进行桥接网络功能受到限制。目前，VirtualBox仅支持通过AirPort的IPv4和IPv6。对于其他协议（如IPX），必须选择有线接口。
* 在**Linux**主机上，当使用无线接口进行桥接网络时，功能受到限制。目前，VirtualBox仅支持IPv4和IPv6无线。对于其他协议（如IPX），必须选择有线接口。
* 其他的很多限制请参考官方手册，这里就不在列出了。<https://www.virtualbox.org/manual/ch06.html>。

**桥接模式下：**

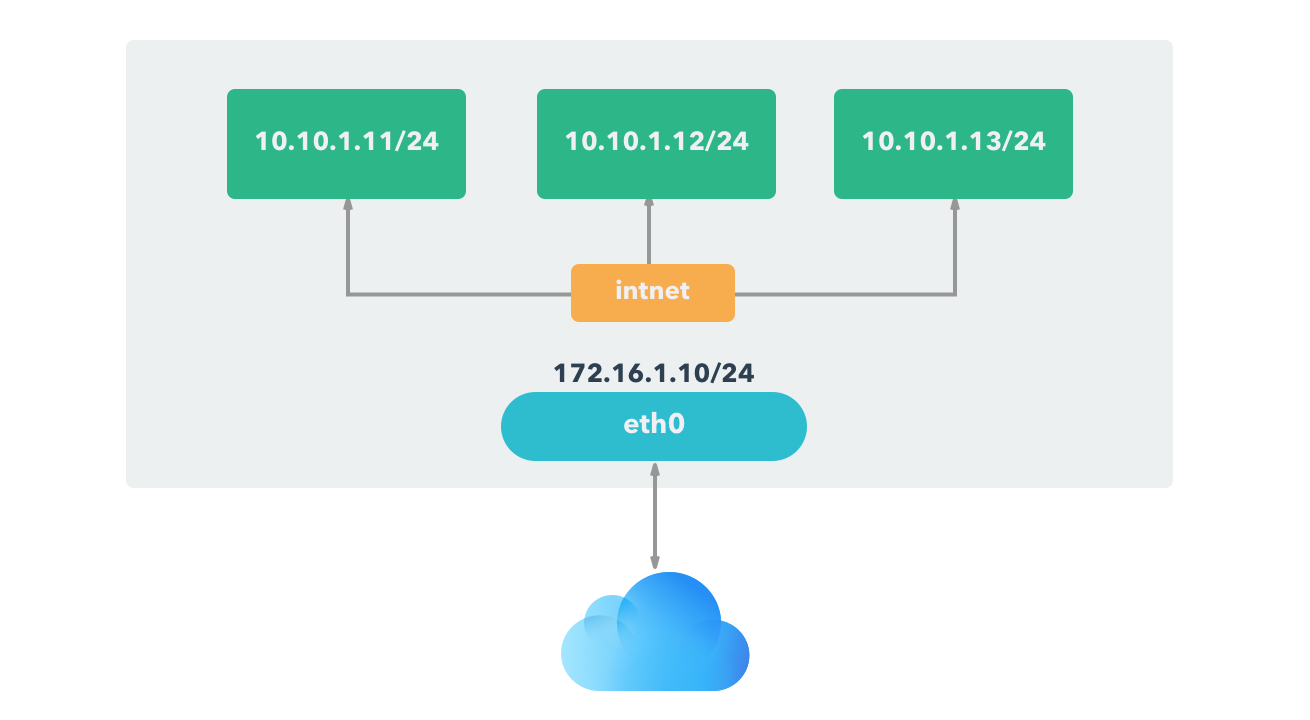
**虚拟机与主机的关系：**可以相互访问，因为虚拟机在真实网络段中有独立IP，主机与虚拟机处于同一网络段中，彼此可以通过各自IP相互访问。

**虚拟机于网络中其他主机的关系：**可以相互访问，同样因为虚拟机在真实网络段中有独立IP，虚拟机与所有网络其他主机处于同一网络段中，彼此可以通过各自IP相互访问。

**虚拟机与虚拟机的关系：**可以相互访问，原因同上。

## 6.6、内部联网

Internal 网络模式和 Bridged 模式很相似，只不过这些虚拟机只能和同主机通模式下的其他虚拟机联通，无法访问外网。技术上来说，这个模式下的所有功能都可以通过桥接模式实现，但是这种模式有个很好的优势就是安全性比较高，主机是无法直接看到虚拟机网络报文的。图示如下：



virtualbox 就是个交换机，不过去桥接模式不同的是这个交换机不与主机系统建立联系，这一点与桥接模式不同。virtualbox连接着主机上该网络模式下连接到同个 Internal 网络的虚拟机。虚拟机之间通过 virtualbox 通信，virtualbox 不会把报文发送到外部。网络 Internal 是通过名字识别的，添加网卡的时候必须选择网络名字，网络名称相同的虚拟机会自动连接到同一个 Internal 网络。VirtualBox支持驱动程序实现完整的以太网交换机，并支持广播/多播帧和混杂模式。

为了将虚拟机的网卡连接到内部网络，有两种方法可以实现：

* 你可以在VirtualBox图形用户界面中使用虚拟机的“设置”对话框。在设置对话框的“网络”类别中，从网络模式的下拉列表中选择“内部网​​络”。现在从下面的下拉列表中选择现有的内部网络的名称，或者在输入框中输入一个新的名称。
* 你也可以使用命令行方式：

VBoxManage modifyvm "VM name" --nic<x> intnet

这个命令会把虚拟机的第 x 块网卡加入到默认的 internal 网络 intnet 中。

或者你也可以指定一个 internal network 名字

VBoxManage modifyvm "VM name" --intnet<x> "networkname"

这个命令把虚拟机的第 x 块网卡设置为 internal 网络模式，并加入到 networkname 的网络中。

虚拟机的 ip 地址可以使用静态IP地址的方式，也可以通过VirtualBox的 DHCP Server来管理内部网络的IP地址。。有关详细信息，参见[第8.38节“VBoxManage dhcpserver”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-dhcpserver)。

作为一种安全措施，默认情况下，内部网络的客户机Linux只允许在相同用户ID下运行的虚拟机建立内部网络。不过，可以创建一个共享的内部网络接口，可以由具有不同UUI的用户访问。

静态IP策略下内部网络客户机系统的配置案例：

对于linux，手动配置ip和子网掩码，而且手动配置时需保证各个虚拟机ip在同一网段），如果是centos7，可依照如下操作：

centos7中手动添加ip和子网掩码的方法：

ifconfig：查看虚拟机上有哪些网卡

vi /etc/sysconfig/network-scripts/ifcfg-xxx　　xxx为具体的网卡名

修改BOOTPROTO＝“none”或者“static”，这样设置成手动，默认是“dhcp“是动态获取ip

最后增加IPADDR＝”192.168.1.1“，NETMASK＝”255.255.255.0“

:wq　退出保存

service network restart 重启服务，生效

以同样的方法，设置另1台centos7虚拟机.

内部网络模式下：

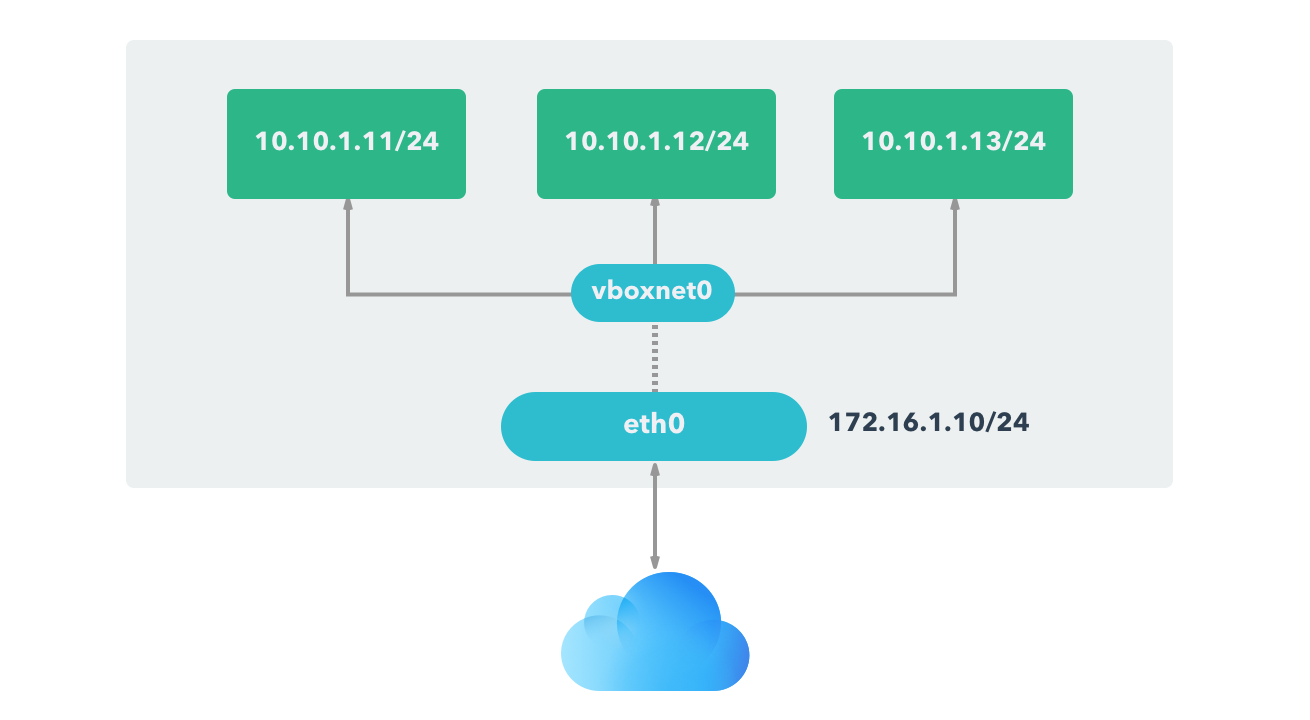
**虚拟机与主机的关系：**不能相互访问，彼此不属于同一个网络，无法相互访问。

**虚拟机与网络中其他主机的关系：**不能相互访问，理由同上。

**虚拟机与虚拟机的关系：**可以相互访问，前提是在设置网络时，两台虚拟机设置同一网络名称。

## 6.7、仅主机网络

Host-only网络是VirtualBox 2.2之后添加的另一种网络模式。它可以被认为是桥接和内部网络模式之间的混合：与桥接网络一样，虚拟机之间可以通信，虚拟机也可以与主机通信。但是注意：与内部网络一样，虚拟机不能与主机之外的世界通信，因为它们没有连接到物理网络接口。当使用仅主机联网时，VirtualBox会在主机上创建一个新的软件界面（其实就是在主机上创建了一个虚拟网卡）。换句话说，通过桥接网络，现有的物理网络接口用于将虚拟机连接到主机。而通过内部联网，外边又看不到虚拟机的流量。桥接模式下，虚拟机和主机处于同一网络（网络号和掩码相同），仅主机网络下：虚拟机之间处于同一网络下，但是虚拟机和主机不在同一网络下。图示如下：



要将虚拟机的虚拟网络接口更改为“仅主机”模式有如下两种做法：

* 到图形用户界面中的虚拟机设置中的“网络”页面，然后选择“仅主机联网”；
* 在命令行中键入VBoxManage modifyvm "VM name" --nic<x> hostonly；详见[8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

在将VM附加到仅主机网络之前，你必须至少创建一个仅限于主机的接口，有两种方法：

* GUI：“文件”→“首选项”→“网络”→“仅主机网络”→“（+）添加主机网络“，MacBook下的入口为VirtualBox偏好设置；
* 命令行：VBoxManage hostonlyif create，详见[第8.37节“VBoxManage hostonlyif”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-hostonlyif)。

对于仅主机网络，像内部网络一样，你会发现内置于VirtualBox中的DHCP服务器非常有用。这样可以启用主机专用网络中的IP地址管理，否则你将需要静态配置所有IP地址。

* 在VirtualBox图形用户界面中，你可以通过“文件”→“首选项”→“网络”在全局设置中配置所有这些项目，其中列出了目前正在使用的所有仅主机网络。单击网络名称，然后点击右侧的“编辑”按钮，你可以修改适配器和DHCP设置。MacBook下入口为VirtualBox偏好设置。
* 在命令行中你可以使用VBoxManage dhcpserver；有关详细信息，请参见[第8.38节“VBoxManage dhcpserver”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-dhcpserver)。

**注意：**在Linux和Mac OS X主机上，仅主机接口的数量限制为128。Solaris和Windows主机没有这样的限制。

**在此小结以下各个常用网络的访问情况：**

|  | **VM ↔ Host** | **VM1 ↔ VM2** | **VM → Internet** | **VM ← Internet** |
| --- | --- | --- | --- | --- |
| **Host-only** | **+** | **+** | **–** | **–** |
| **Internal** | **–** | **+** | **–** | **–** |
| **Bridged** | **+** | **+** | **+** | **+** |
| **NAT** | **–** | **–** | **+** | [**Port forwarding**](https://www.virtualbox.org/manual/ch06.html#natforward) |
| **NAT Network** | **–** | **+** | **+** | [**Port forwarding**](https://www.virtualbox.org/manual/ch06.html#network_nat_service) |

## 6.8、UDP隧道网络

这种联网模式允许互连在不同主机上运行的虚拟机。这个模式也不常用，具体细节参考官方手册即可：

<https://www.virtualbox.org/manual/ch06.html#network_udp_tunnel>。

## 6.9、VDE网络

虚拟分布式以太网（VDE）是一种灵活的虚拟网络基础设施，以安全的方式跨越多台主机。它允许L2 / L3交换，包括生成树协议，VLAN和WAN仿真。它是VirtualBox的一个可选部分，它只包含在源代码中。VDE仅在Linux和FreeBSD主机上可用。由于VDE很少使用，因此有关设置VDE网络的更多信息，请参阅软件随附的文档。[http://wiki.virtualsquare.org/wiki/index.php/VDE\_Basic\_Networking](http://wiki.virtualsquare.org/wiki/index.php/VDE_Basic_Networking" \t "_top)。

## 6.10、限制网络I / O的带宽

从4.2版开始，VirtualBox允许限制用于网络传输的最大带宽。一个虚拟机的几个网络适配器可以通过带宽组共享限制。

**注意：**VirtualBox仅限制虚拟机发送数据包，不限制虚拟机接收的流量。

限制通过 VBoxManage配置。下面的示例创建一个名为“Limit”的带宽组，将限制设置为20 Mbit/s，并将组分配给VM的第一个和第二个适配器：

VBoxManage bandwidthctl "VM name" add Limit --type network --limit 20m

VBoxManage modifyvm "VM name" --nicbandwidthgroup1 Limit

VBoxManage modifyvm "VM name" --nicbandwidthgroup2 Limit

组中的所有适配器共享带宽限制，这意味着在上面的示例中，两个适配器的带宽和永远不能超过20 Mbit/s。但是，如果一个适配器不需要带宽，则另一个适配器可以使用带宽组的剩余带宽。虚拟机正在运行时也可以更改每个带宽组的限制，并立即生效。以下示例将上述示例中创建的组的限制更改为100 Kbit/s：

VBoxManage bandwidthctl "VM name" set Limit --limit 100k

要完全禁用VM的第一个适配器的流量，请使用以下命令：

VBoxManage modifyvm“VM name”--nicbandwidthgroup1 none

在VM运行时，你也可以通过指定带宽组的限制为0，来禁止所有分配给带宽组的虚拟机网络适配器访问网络。例如，对于名为“Limit”的带宽组，请使用：

VBoxManage bandwidthctl "VM name" set Limit --limit 0

## 6.11、提高网络性能

VirtualBox提供了多种虚拟网络适配器，也提供了多种联网方式。这些适配器之间，以及联网方式的不同，网络性能也会有所不同。性能方面，Virtio网络适配器要优于Intel PRO / 1000仿真适配器，也优于PCNet系列适配器。三种联网类型：内部、桥接和仅主机，几乎具有相同的性能，内部类型更快一点，并且使用较少的CPU周期，因为数据包不会到达主机的网络堆栈。NAT是最慢的（最安全的）。

分配给VM的CPU数量不会提高网络性能，在某些情况下可能会由于客户机系统中的并发性增加而对性能有害。

为了提高网络性能通常注意如下几点：

1. 只要有可能就使用*virtio*网络适​​配器，否则使用*Intel PRO / 1000* 适配器之一;
2. 使用桥接模式，而不是 NAT ;
3. 确保在客户机操作系统中启用分段卸载。默认启用。你可以使用ethtool Linux客户机中的命令检查和修改卸载设置。
4. 使用第三方工具（如Wireshark），对VM网络适配器上的网络流量进行全面详细的分析。为此，需要在VM的网络适配器上使用混杂模式。只能在以下网络模式上使用：NAT网络，桥接模式，内部网络模式和仅主机模式。

要设置混杂模式，请从网络适配器的“网络设置”对话框中的下拉列表中选择，或使用命令行工具VBoxManage; 有关详细信息，请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

混杂模式策略是：

* 1. deny （默认设置）：隐藏任何不适用于此虚拟机网卡的流量。
  2. allow-vms ：隐藏来自该虚拟机网卡的所有主机的流量，允许其查看虚拟机之间的所有流量。
  3. allow-all 该虚拟机的网络适配器可以看到所有流量。

# 第七章 远程虚拟机

**7.1。远程显示（VRDP支持）**

VirtualBox可以远程显示虚拟机，这意味着虚拟机可以在一台计算机上执行，即使机器将显示在第二台计算机上，并且机器也将从那里进行控制，就好像虚拟机正在运行的那样电脑。

为了最大的灵活性，从VirtualBox 4.0开始，VirtualBox通过通用扩展接口VirtualBox Remote Desktop Extension（VRDE）实现远程机器显示。基本的开源VirtualBox包仅提供此接口，而第三方可以使用VirtualBox扩展包提供实现，这些扩展包必须与基础包分开安装。有关详细信息[，](https://www.virtualbox.org/manual/ch01.html#intro-installing)请参见[第1.5节“安装VirtualBox和扩展包”](https://www.virtualbox.org/manual/ch01.html#intro-installing)。

Oracle 在VirtualBox扩展包中提供对**VirtualBox远程显示协议（VRDP）的**支持。当安装此软件包时，VirtualBox 4.0及更高版本支持VRDP的方式与VirtualBox 4.0之前的二进制（非开源）版本相同。

VRDP是Microsoft的远程桌面协议（RDP）的向后兼容扩展。因此，您可以使用任何标准RDP客户端来控制远程VM。

即使安装了扩展，默认情况下也禁用VRDP服务器。它可以在每个虚拟机的基础上，在VirtualBox Manager中的“显示”设置中轻松启用（请参见[第3.5节“显示设置”](https://www.virtualbox.org/manual/ch03.html#settings-display)）或与 VBoxManage：

VBoxManage modifyvm“VM名称”--vrde打开

缺省情况下，VRDP服务器使用TCP端口 3389。如果运行多个VRDP服务器，则需要更改默认端口，因为端口只能由一个服务器一次使用; 您可能还需要在Windows主机上进行更改，因为默认端口可能已被Windows本身内置的RDP服务器使用。端口5000至5050通常不被使用，可能是一个不错的选择。

可以在图形用户界面的“显示”设置或命令--vrdeport选项中 更改端口 VBoxManage modifyvm。您可以指定逗号分隔的端口列表或端口范围。在两个端口号之间使用破折号来指定范围。VRDP服务器将绑定到指定列表中的**一个**可用端口。例如，VBoxManage modifyvm "VM name" --vrdeport 5000,5010-5012将配置服务器绑定到其中一个端口5000,5010,5011或5012.有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm-vrde)请参见[第8.8.5节“远程机器设置”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm-vrde)。

运行的VM使用的实际端口可以使用VBoxManage showvminfo命令查询 或在“会话信息对话框”的“运行时”选项卡上的GUI中查看，可以通过VM窗口的“机器”菜单访问。

VirtualBox 4.3中已经实现了对IPv6的支持。如果主机操作系统支持IPv6，VRDP服务器将自动侦听除IPv4之外的IPv6连接。

**7.1.1。普通的第三方RDP观众**

由于VRDP与RDP向后兼容，您可以使用任何标准RDP查看器连接到这样的远程虚拟机（下面的示例）。对于这个工作，你必须指定 **IP地址，**您的 *主机*（不是虚拟机的！）作为服务器的地址连接到系统，以及该**端口号**的VRDP服务器使用。

以下是最常见的RDP观众的示例：

* 在Windows上，可以使用mstsc.exeWindows附带的Microsoft Terminal Services Connector（）。您可以启动“运行”对话框（按Windows键和“R”）并键入“mstsc”。您还可以在“开始”→“所有程序”→“附件”→“远程桌面连接”下找到它。如果您使用“运行”对话框，您可以直接键入选项：

mstsc 1.2.3.4:3389

更换1.2.3.4主机IP地址，并3389在必要时使用不同的端口。

**注意**

IPv6地址必须用方括号括起来指定端口。例如：mstsc [fe80::1:2:3:4]:3389

**注意**

当连接以测试连接，地址为localhost localhost和 127.0.0.1使用可能无法正常工作 mstsc.exe。而是127.0.0.2[:3389]必须使用地址 。

* 在其他系统上，您可以使用标准的开源程序 rdesktop。这与大多数Linux发行版一起提供，但VirtualBox还附带了rdesktop的修改版本，用于远程USB支持（请参见下面的[第7.1.4节“远程USB”](https://www.virtualbox.org/manual/ch07.html#usb-over-rdp)）。

使用rdesktop，使用以下命令行：

rdesktop -a 16 -N 1.2.3.4:3389

如上面的Microsoft查看器所述，替换1.2.3.4 主机IP地址，并3389在必要时使用不同的端口。该-a 16选项要求每像素16位的颜色深度，我们建议。（为了获得最佳性能，安装客户机操作系统后，应将其显示颜色深度设置为相同的值）。该-N选项可以使用NumPad键。

* 如果运行KDE桌面，可能更喜欢 krdcKDE RDP查看器。命令行将如下所示：

krdc rdp：//1.2.3.4：3389

再次，替换1.2.3.4主机IP地址，并3389在必要时使用不同的端口。krdc需要使用“rdp：//”位才能将其转换为RDP模式。

* 您可以使用Sun Ray瘦客户机 uttsc，这是Sun Ray Windows Connector软件包的一部分。有关详细信息，请参阅相应的文档。

**7.1.2。VBoxHeadless，远程桌面服务器**

当从VirtualBox Manager启动的任何VM都能够远程运行虚拟机时，如果您不想首先在本地显示虚拟机，则必须运行完整的GUI是不方便的。特别是，如果您正在运行的服务器硬件，其唯一目的是托管虚拟机，并且所有虚拟机都应该通过VRDP远程运行，则在服务器上完全没有图形用户界面 - 尤其是在一个Linux或Solaris主机，VirtualBox管理器附带了对Qt和SDL库的依赖。如果您宁愿在服务器上没有X Window系统，这是不方便的。

因此，VirtualBox附带了另一个前端 VBoxHeadless，它在主机上根本不产生可视输出，但仍然可以传送VRDP数据。这个前端没有依赖于Linux和Solaris主机上的X Window系统。[[35](https://www.virtualbox.org/manual/ch07.html" \l "ftn.idm3180) ]

要启动虚拟机 VBoxHeadless，您有三个选项：

* 您可以使用

VBoxManage startvm“VM名称” - 型无头

额外的--type选项使VirtualBox VBoxHeadless用作内部虚拟化引擎的前端，而不是Qt前端。

* 一种替代方法是VBoxHeadless直接使用 ，如下所示：

VBoxHeadless --startvm <uuid | name>

这种启动VM的方法有助于排除报告的问题，VBoxManage startvm ... 因为您可以看到有时更详细的错误消息，特别是在VM执行开始之前的早期故障。在正常情况下VBoxManage startvm 是优选的，因为它直接运行VM作为后台进程，必须在直接启动时明确完成 VBoxHeadless。

* 另一个选择是 VBoxHeadless从VirtualBox Manager GUI开始，通过在启动虚拟机或Headless Start 从Machine菜单中选择时按住Shift键。

由于VirtualBox版本5.0，当您使用 VBoxHeadless启动虚拟机时，将根据虚拟机配置启用VRDP服务器。您可以使用--vrde 命令行参数覆盖虚拟机的设置。要使VRDP服务器启动虚拟机，请执行以下操作：

VBoxHeadless --startvm <uuid | name> --vrde on

并禁用它：

VBoxHeadless --startvm <uuid | name> --vrde关闭

要使VRDP服务器根据VM配置启用，与其他前端一样，您仍然可以使用：

VBoxHeadless --startvm <uuid | name> --vrde config

但这是一样的

VBoxHeadless --startvm <uuid | name>

如果您启动虚拟机，VBoxManage startvm ... 那么始终使用虚拟机的配置设置。

**7.1.3。一步一步：在无头服务器上创建虚拟机**

以下说明可能会让您了解如何通过网络连接在无头服务器上创建虚拟机。我们将创建一个虚拟机，建立一个RDP连接并安装客户机操作系统，而无需触摸无头服务器。所有你需要的是以下内容：

1. 具有受支持主机操作系统的服务器机器上的VirtualBox。必须安装VRDP服务器的VirtualBox扩展包（请参阅上一节）。对于以下示例，我们将假设一个Linux服务器。
2. 可以从服务器访问的ISO文件，其中包含要安装的客户机操作系统的安装数据（我们将在以下示例中假定Windows XP）。
3. 与该主机的终端连接，您可以通过该连接访问命令行（例如通过 ssh）。
4. 远程客户端上的RDP查看器; 有关示例[，](https://www.virtualbox.org/manual/ch07.html#rdp-viewers)请参见[第7.1.1节“常见的第三方RDP查看器”](https://www.virtualbox.org/manual/ch07.html#rdp-viewers)。

再次请注意，在服务器机器上，由于我们只使用无头服务器，所以不需要Qt，SDL和X Window系统。

1. 在无头服务器上，创建一个新的虚拟机：

VBoxManage createvm - “Windows XP” - WindowsXP注册 - 注册

请注意，如果不指定 --register，则必须registervm 稍后手动使用该命令。

请注意，您不需要指定 --ostype，但是这样做会为某些VM参数选择一些合理的默认值，例如RAM大小和虚拟网络设备的类型。要获得可以使用的支持的操作系统的完整列表

VBoxManage列表ostypes

1. 确保此VM的设置适用于我们将要安装的客户机操作系统。例如：

VBoxManage modifyvm“Windows XP” - 内存256 --acpi on --boot1 dvd --nic1 nat

1. 为虚拟机创建虚拟硬盘（在这种情况下，大小为10 GB）：

VBoxManage createhd --filename“WinXP.vdi”--size 10000

1. 向新VM添加一个IDE控制器：

VBoxManage storagectl“Windows XP” - 名称“IDE控制器” - 控制器PIIX4

1. 将上面创建的VDI文件设置为新VM的第一个虚拟硬盘：

VBoxManage storageattach“Windows XP”--storagectl“IDE Con​​troller” --port 0 --device 0 --type hdd --medium“WinXP.vdi”

1. 将包含要稍后安装的操作系统安装的ISO文件附加到虚拟机，以便机器可以从其启动：

VBoxManage storageattach“Windows XP”--storagectl“IDE Con​​troller” --port 0 --device 1 - type dvddrive --medium /full/path/to/iso.iso

1. 启用VirtualBox远程桌面扩展（VRDP服务器）：

VBoxManage modifyvm“Windows XP”--vrde on

1. 使用VBoxHeadless启动虚拟机：

VBoxHeadless --startvm“Windows XP”

如果一切正常，您应该看到版权声明。相反，如果你返回到命令行，那么出现了错误。

1. 在客户端计算机上，启动RDP查看器并尝试连接到服务器（有关如何使用各种常见RDP查看器，请参阅上面的[第7.1.1节“常见第三方RDP查看器”](https://www.virtualbox.org/manual/ch07.html#rdp-viewers)）。

您现在应该在RDP查看器中远程查看客户机操作系统的安装例程。

**7.1.4。远程USB**

作为VRDP支持之上的一个特殊功能，VirtualBox还支持远程USB设备。也就是说，在一台计算机上运行的VirtualBox guest虚拟机可以访问正在显示VRDP数据的远程计算机的USB设备，与连接到实际主机的USB设备相同。这允许在充当服务器的VirtualBox主机上运行虚拟机，客户端可以从仅需要网络适配器的其他地方和能够运行RDP查看器的显示器进行连接。当USB设备插入客户端时，远程VirtualBox服务器可以访问它们。

对于这些远程USB设备，与其他USB设备相同的过滤规则适用，如[第3.10.1节“USB设置”所述](https://www.virtualbox.org/manual/ch03.html#settings-usb)。所有您需要做的是在设置这些规则时指定“远程”（或“任何”）。

仅当RDP客户端支持此扩展时，才能访问远程USB设备。在Linux和Solaris主机上，VirtualBox安装提供了一个合适的VRDP客户端 rdesktop-vrdp。uttsc专为使用Sun Ray瘦客户机定制的客户端的最新版本 也支持访问远程USB设备。其他平台的RDP客户端将在未来的VirtualBox版本中提供。

要使远程USB设备可用于VM， rdesktop-vrdp应启动如下：

rdesktop-vrdp -r usb -a 16 -N my.host.address

有关如何正确设置USB设备的权限的更多详细信息[，](https://www.virtualbox.org/manual/ch12.html#ts_usb-linux) 请参阅[第12.8.7节“USB不工作”](https://www.virtualbox.org/manual/ch12.html#ts_usb-linux)。此外，建议禁用在远程主机上自动加载可能在USB设备上工作的任何主机驱动程序，以确保RDP客户端可以访问这些设备。如果在远程主机上正确完成设置，则在VM的VBox.log文件上可以看到插拔事件。

**7.1.5。RDP认证**

对于通过RDP远程访问的每个虚拟机，您可以单独确定客户端连接是否以及如何进行身份验证。为此，使用VBoxManage modifyvm带有--vrdeauthtype选项的命令 ; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)一般介绍。有三种身份验证方法可用：

* “null”方法意味着根本没有认证; 任何客户端都可以连接到VRDP服务器，从而连接虚拟机。这当然是非常不安全的，只有建议私人网络使用。
* “外部”方法通过特殊认证库提供外部认证。VirtualBox附带两个这样的认证库：
  1. 默认身份验证库 VBoxAuth根据主机的用户凭据进行身份验证。根据主机平台，这意味着：
     + 在Linux主机上， VBoxAuth.so 根据主机的PAM系统对用户进行身份验证。
     + 在Windows主机上， VBoxAuth.dll 根据主机的WinLogon系统对用户进行身份验证。
     + 在Mac OS X主机上， VBoxAuth.dylib 根据主机的目录服务验证用户。[[36](https://www.virtualbox.org/manual/ch07.html" \l "ftn.idm3302) ]

换句话说，每个默认的“外部”方法都对主机系统上存在的用户帐户执行身份验证。任何具有有效身份验证凭证的用户都被接受，即用户名不必与运行该虚拟机的用户对应。

* 1. 一个额外的库称为 VBoxAuthSimple对虚拟机的XML设置文件的“extradata”部分中配置的凭据执行身份验证。这可能是获取身份验证的最简单的方法，该认证不依赖于正在运行和受支持的客户端（见下文）。需要以下步骤：
     + VBoxAuthSimple使用以下命令启用 ：

VBoxManage setproperty vrdeauthlibrary“VBoxAuthSimple”

* + - 要为特定VM启用库，必须将身份验证切换到外部：

VBoxManage modifyvm“VM名称”--vrdeauthtype外部

替换 <vm>为VM名称或UUID。

* + - 然后，您需要通过将项目写入机器的extradata来配置用户和密码。由于XML机器设置文件，其中需要写入密码的“extradata”部分是纯文本文件，VirtualBox使用哈希加密密码。必须使用以下命令：

VBoxManage setextradata“VM name”“VBoxAuthSimple / users / <user>”<hash>

替换 <vm>为虚拟机名称或UUID， <user>用户名应该被允许登录，并 <hash>加密密码。例如，要获取密码“secret”的哈希值，可以使用以下命令：

VBoxManage内部命令passwordhash“secret”

这将打印

2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b

然后，您可以使用VBoxManage setextradata将该值存储在计算机的“extradata”部分中。

例如，组合在一起，要将用户“john”和机器“My VM”的密码设置为“secret”，请使用以下命令：

VBoxManage setextradata“我的VM”“VBoxAuthSimple / users / john” 2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b

* 最后，“访客”身份验证方法使用Guest Additions附带的特殊组件执行身份验证; 因此，不会在主机上执行身份验证，而是使用*访客*用户帐户。

此方法目前仍在测试中，尚未支持。

除了上述方法之外，您还可以使用任何其他模块替换默认的“外部”身份验证模块。为此，VirtualBox提供了一个定义良好的界面，可让您编写自己的身份验证模块。这在VirtualBox软件开发工具包（SDK）参考中有详细描述; 有关详细信息[，](https://www.virtualbox.org/manual/ch11.html)请参阅[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。

**7.1.6。RDP加密**

RDP具有基于RC4对称密码（密钥高达128位）的数据流加密。固定间隔（每4096个数据包）更换RC4密钥。

RDP提供不同的认证方式：

1. 历史上，使用RDP4身份验证，RDP客户端不进行任何检查，以验证其连接到的服务器的身份。由于可以使用“中间人”（MITM）攻击获得用户凭据，所以RDP4认证是不安全的，通常不应该使用。
2. RDP5.1认证使用客户端拥有公钥的服务器证书。这样保证服务器拥有相应的私钥。然而，由于这些硬编码的私钥几年前公开，RDP5.1认证也是不安全的。
3. RDP5.2认证使用增强型RDP安全性，这意味着使用外部安全协议来保护连接。RDP4和RDP5.1使用标准RDP安全。VRDP服务器支持使用TLS协议的增强型RDP安全性，作为TLS握手的一部分，将服务器证书发送给客户端。

所述Security/MethodVRDE属性设置所期望的安全方法，其用于一个连接。有效值为：

* + Negotiate - 允许增强（TLS）和标准RDP安全连接。安全方法与客户端协商。这是默认设置。
  + RDP - 仅接受标准RDP安全。
  + TLS - 仅接受增强型RDP安全性。客户端必须支持TLS。

例如，以下命令允许客户端使用标准或增强型RDP安全连接：

vboxmanage modifyvm“VM名称”--vrdeproperty“Security / Method = negotiate”

如果Security/Method属性设置为Negotiate或者 TLS，TLS协议将被服务器自动使用，如果客户端支持TLS。但是，为了使用TLS，服务器必须拥有服务器证书，服务器私钥和证书颁发机构（CA）证书。以下示例显示如何生成服务器证书。

* + 创建CA自签名证书：

openssl req -new -x509 -days 365 -extensions v3\_ca \ -keyout ca\_key\_private.pem -out ca\_cert.pem

* + 生成服务器私钥和签名请求：

openssl genrsa -out server\_key\_private.pem openssl req -new -key server\_key\_private.pem -out server\_req.pem

* + 生成服务器证书：

openssl x509 -req -days 365 -in server\_req.pem \ -CA ca\_cert.pem -CAkey ca\_key\_private.pem -set\_serial 01 -out server\_cert.pem

服务器必须配置为访问所需的文件：

vboxmanage modifyvm“VM名称”\ --vrdeproperty“Security / CACertificate = path / ca\_cert.pem”

vboxmanage modifyvm“VM名称”\ --vrdeproperty“Security / ServerCertificate = path / server\_cert.pem”

vboxmanage modifyvm“VM名称”\ --vrdeproperty“Security / ServerPrivateKey = path / server\_key\_private.pem”

由于连接到服务器的客户端将决定使用哪种类型的加密，使用rrdktop，Linux RDP查看器，使用 -4或 -5选项。

**7.1.7。与VRDP服务器的多个连接**

VirtualBox的VRDP服务器支持与来自不同客户端的相同运行的VM的多个并发连接。所有连接的客户端看到相同的屏幕输出，并共享鼠标指针和键盘焦点。这与几个同时使用同一台电脑的用户类似，轮流在键盘上。

以下命令启用多连接模式：

VBoxManage modifyvm“VM名称”--vrdemulticon on

**7.1.8。多个远程显示器**

要访问两个或多个远程VM显示器，必须启用VRDP多连接模式（请参见[第7.1.7节“与VRDP服务器的多个连接”](https://www.virtualbox.org/manual/ch07.html#vrde-multiconnection)）。

RDP客户端可以使用domain登录参数（-d）来选择要连接的虚拟监视器号。如果参数以后 @跟一个数字，VirtualBox将此数字解释为屏幕索引。选择主访客屏幕@1，第一个辅助屏幕@2等

Microsoft RDP6客户端不允许您指定单独的域名。而是domain\username在 Username:现场使用 - 例如 @2\name。 name必须提供，如果VRDP服务器设置为需要凭据，则必须是用于登录的名称。如果不是，您可以使用任何文本作为用户名。

**7.1.9。VRDP视频重定向**

从VirtualBox 3.2开始，VRDP服务器可以将客户端的视频流重定向到RDP客户端。视频帧使用JPEG算法进行压缩，允许比标准RDP位图压缩方法更高的压缩比。通过降低视频质量可以提高压缩率。

VRDP服务器自动检测来宾中的视频流，因为频繁更新的矩形区域。因此，该方法适用于任何客户操作系统，而无需在客户端安装其他软件; 特别是客人添加不是必需的。

然而，在客户端，目前只有Windows 7远程桌面连接客户端支持此功能。如果客户端不支持视频重定向，VRDP服务器将恢复正常的位图更新。

以下命令启用视频重定向：

VBoxManage modifyvm“VM名称”--vrdevideochannel on

视频的质量定义为10至100％的值，表示JPEG压缩级别（其中较低的数字意味着更低的质量，但更高的压缩率）。可以使用以下命令更改质量：

VBoxManage modifyvm“VM名称”--vrdevideochannelquality 75

**7.1.10。VRDP定制**

使用VirtualBox 4.0可以在VRDP服务器中单独禁用显示输出，鼠标和键盘输入，音频，远程USB或剪贴板。

以下命令更改对应的服务器设置：

VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableDisplay = 1 VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableInput = 1 VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableUSB = 1 VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableAudio = 1 VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableClipboard = 1 VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableUpstreamAudio = 1

要重新启用功能，请使用类似的命令，而不使用尾部1.例如：

VBoxManage modifyvm“VM名称”--vrdeproperty Client / DisableDisplay =

VirtualBox 3.2.10介绍了这些属性。但是，在3.2.x系列中，需要使用以下命令来改变这些设置：

VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableDisplay”1 VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableInput”1 VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableUSB”1 VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableAudio”1 VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableClipboard”1

要重新启用功能，请使用类似的命令，而不使用尾部1.例如：

VBoxManage setextradata“VM名称”“VRDP / Feature / Client / DisableDisplay”

**7.2。隐形传态**

从版本3.1开始，VirtualBox支持“传送” - 即虚拟机正在运行时，通过网络将虚拟机从一个VirtualBox主机移动到另一个虚拟机。无论主机上运行的主机操作系统如何，都可以这样工作：例如，可以在Solaris和Mac主机之间传送虚拟机。

传送要求机器当前正在一台主机上运行，​​然后称为**“源”**。虚拟机将被传送到的主机将被称为**“目标”** ; 然后将目标上的计算机配置为等待源接触目标。然后，机器的运行状态将以最少的停机时间从源传输到目标。

传送在任何TCP / IP网络上发生; 源和目标只需要在传送设置中指定的TCP / IP端口上同意。

在这个时候，这个工作有几个先决条件，但是：

1. 在目标主机上，您必须在VirtualBox中配置与要传送的源上的计算机完全相同的硬件设置。这不适用于仅仅是描述性的设置，例如VM名称，但是显然对于远程传送工作，目标机器必须具有相同的内存量和其他硬件设置。否则传送将失败并显示错误消息。
2. 源和目标上的两个虚拟机必须共享相同的存储（硬盘以及软盘和CD / DVD映像）。这意味着它们使用相同的iSCSI目标或存储位于网络上的某个位置，并且两个主机都可以通过NFS或SMB / CIFS访问它们。

这也意味着源和目标机器都不能有任何快照。

然后执行以下步骤：

1. 在*目标*主机上，配置虚拟机等待传送请求在启动时到达，而不是实际尝试启动计算机。这是通过以下VBoxManage命令完成的：

VBoxManage modifyvm <targetvmname> --teleporter on --teleporterport <port>

<targetvmname>目标主机上虚拟机的名称在哪里，并且 <port>是要在源主机和目标主机上使用的TCP / IP端口号。例如，使用6000.有关详细信息，请参见[第8.8.6节“传送设置”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm-teleport)。

1. 在目标主机上启动虚拟机。你会看到，而不是实际运行，它将显示一个进度对话框。表示正在等待传送请求到达。
2. 像往常一样在*源*主机上启动机器。当它运行并希望将其传送时，请在源主机上发出以下命令：

VBoxManage controlvm <sourcevmname> teleport --host <targethost> --port <port>

其中<sourcevmname>是源主机（即当前正在运行的机器）上的虚拟机的名称， <targethost>是在其上机正在等待瞬间移动请求的目标主机的主机或IP名，和<port> 按照指定必须相同数目在目标主机上的命令。有关详细信息，请参见[第8.13节“VBoxManage controlvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm)。

对于测试，您还可以在同一主机上传送机器; 在这种情况下，请在源和目标主机上使用“localhost”作为主机名。

**注意**

在极少数情况下，如果源和目标的CPU非常不同，传送可能会失败并显示错误消息，否则目标可能会挂起。如果VM正在运行高度优化以在特定CPU上运行但未正确检查某些CPU功能实际存在的应用程序软件，则可能会发生这种情况。VirtualBox将客户机操作系统中显示的CPU功能进行过滤。高级用户可以尝试使用VBoxManage --modifyvm --cpuid命令限制这些虚拟CPU功能; 请参见[第8.8.6节“传送设置”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm-teleport)。

[[35](https://www.virtualbox.org/manual/ch07.html" \l "idm3180) ]在VirtualBox 1.6之前，无头服务器被调用 VBoxVRDP。为了向后兼容，VirtualBox安装仍然使用该名称安装可执行文件。

[[36](https://www.virtualbox.org/manual/ch07.html" \l "idm3302) ] 3.2版中添加了对Mac OS X的支持。

# 第八章 VBoxManage

# 第九章 高级主题

**9.1。VBoxSDL，简化的VM显示器**

**9.1.1。介绍**

VBoxSDL是一个简单的图形用户界面（GUI），缺少我们主要的GUI提供的VirtualBox的良好的点击支持。VBoxSDL目前主要用于内部调试VirtualBox，因此没有正式支持。但是，您可能会发现，虚拟机不一定由使用虚拟机的同一个人控制的环境很有用。

**9.2。自动访客登录**

VirtualBox为Windows，Linux和Solaris提供Guest Addition模块，以便在客户端上启用自动登录。

当客户机操作系统在虚拟机中运行时，可能需要使用主登录系统的凭据执行协调和自动登录。（使用“凭据”，我们指的是由用户名，密码和域名组成的登录信息，每个值可能为空。）

**9.2.1。自动Windows访客登录**

由于Windows NT，Windows提供了一个模块化系统登录子系统（“Winlogon”），可以通过所谓的GINA模块（图形识别和验证）进行定制和扩展。对于Windows Vista和Windows 7，GINA模块被称为“凭据提供者”的新机制所替代。Windows的VirtualBox Guest Additions包含GINA和凭据提供程序模块，因此可以使任何Windows客户机执行自动登录。

要激活VirtualBox GINA或凭据提供程序模块，请使用命令行开关安装Guest Additions /with\_autologon。安装这些模块所需的所有以下手动步骤将由安装人员完成。

要手动安装VirtualBox GINA模块，请提取访客添加（请参见[第4.2.1.4节“手动文件提取”](https://www.virtualbox.org/manual/ch04.html#windows-guest-file-extraction)），并将文件复制VBoxGINA.dll到Windows SYSTEM32目录。然后，在注册表中，创建以下密钥：

HKEY\_LOCAL\_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ Winlogon \ GinaDLL

具有价值VBoxGINA.dll。

**注意**

VirtualBox GINA模块实现为标准Windows GINA模块（MSGINA.DLL）的封装。因此，最有可能无法正常使用第三方GINA模块。

要手动安装VirtualBox凭据提供程序模块，请提取访客添加（请参见[第4.2.1.4节“手动文件提取”](https://www.virtualbox.org/manual/ch04.html#windows-guest-file-extraction)），并将文件复制 VBoxCredProv.dll到Windows SYSTEM32目录。然后，在注册表中，创建以下密钥：

HKEY\_LOCAL\_MACHINE \ SOFTWARE \微软\的Windows \ CurrentVersion \ 认证\凭证提供商\ {275D3BCC-22BB-4948-A7F6-3A3054EBA92B} HKEY\_CLASSES\_ROOT \ CLSID \ {275D3BCC-22BB-4948-A7F6-3A3054EBA92B} HKEY\_CLASSES\_ROOT \ CLSID \ {275D3BCC-22BB-4948-A7F6-3A3054EBA92B} \ InprocServer32的

具有所有默认值（(Default)每个键中命名 的键）设置为 VBoxCredProv。之后，一个新的字符串命名

HKEY\_CLASSES\_ROOT \ CLSID \ {275D3BCC-22BB-4948-A7F6-3A3054EBA92B} \ InprocServer32的\的ThreadingModel

Apartment必须创建 一个值。

要设置凭据，请在*正在运行的* VM 上使用以下命令 ：

VBoxManage controlvm“Windows XP”setcredentials“John Doe”“secretpassword”“DOMTEST”

当VM运行时，可以使用VirtualBox Guest Additions设备驱动程序通过VirtualBox登录模块（GINA或凭据提供程序）查询凭据。当Windows处于“注销”模式时，登录模块将不断轮询凭据，如果存在，将尝试登录。检索凭证后，登录模块将擦除它们，以便上述命令必须重复以进行后续登录。

出于安全考虑，凭证不会以任何持久的方式存储，并在虚拟机重置时丢失。此外，凭证是“只写”，即没有办法从主机端检索凭据。可以通过设置空值从主机端复位凭证。

根据Windows客户机的特定变体，适用以下限制：

1. 对于**Windows XP guest**虚拟机**，**登录子系统需要配置为使用经典登录对话框，因为VirtualBox GINA模块不支持XP风格的欢迎对话框。
2. 对于**Windows Vista，Windows 7和Windows 8 guest虚拟机，**登录子系统不支持所谓的安全注意序列（CTRL+ALT+DEL）。因此，客人的组策略设置需要更改为不使用安全注意顺序。此外，给定的用户名仅与真实用户名进行比较，而不是用户友好名称。这意味着当您重命名用户时，您仍然必须提供原始用户名（内部，Windows从不重命名用户帐户）。
3. 默认情况下禁用内置Windows远程桌面服务（以前称为终端服务）的自动登录处理。要启用它，请创建注册表项

HKEY\_LOCAL\_MACHINE \ SOFTWARE \ Oracle \ VirtualBox Guest Additions \ AutoLogon

具有DWORD价值 1。

以下命令强制VirtualBox在客户端读取并重置VM后保留凭据：

VBoxManage setextradata“Windows XP”VBoxInternal / Devices / VMMDev / 0 / Config / KeepCredentials 1

请注意，这是潜在的安全风险，因为在客户端上运行的恶意应用程序可以使用适当的界面请求此信息。

**9.2.2。自动Linux / Unix访客登录**

从3.2版开始，VirtualBox提供了一个自定义PAM模块（可插拔认证模块），可用于在支持此框架的平台上执行自动访客登录。几乎所有现代Linux / Unix发行版都依赖于PAM。

对于使用LightDM作为显示管理器的Ubuntu（或Ubuntu派生）发行版的自动登录，请参见 [第9.2.2.1节“VirtualBox Greeter for Ubuntu / LightDM”](https://www.virtualbox.org/manual/ch09.html#autologon_unix_lightdm)。

该pam\_vbox.so模块本身 **并不**做传递给客户操作系统的凭证的实际验证; 相反，它依赖于其他模块，例如PAM堆栈中的pam\_unix.so或 pam\_unix2.so向下的模块，使用检索到的凭据进行实际验证pam\_vbox.so。因此 pam\_vbox.so必须在认证PAM服务列表之上。

**注意**

该pam\_vbox.so只支持auth原始。其他原语如account， session或 password不被支持。

pam\_vbox.so模块作为客户添加的一部分发送，但默认情况下不会在客户机操作系统上安装和/或激活该模块。为了安装它，它必须从/opt/VBoxGuestAdditions-<version>/lib/VBoxGuestAdditions/ 安全模块目录复制 到一般 /lib/security/的32位客户机Linux或/lib64/security/64 位客户机上。请参阅您的客户操作系统文档以获取正确的PAM模块目录。

例如，要使用pam\_vbox.so Ubuntu Linux客户机操作系统和GDM（GNOME桌面管理器）使用主机传递的凭据自动登录用户，必须将客户机操作系统配置如下：

1. 该pam\_vbox.so模块必须被复制到安全模块目录，在这种情况下才是 /lib/security。
2. 编辑GDM中找到的PAM配置文件 /etc/pam.d/gdm，将行添加 auth requisite pam\_vbox.so到顶部。此外，在大多数Linux发行版中都有一个文件 /etc/pam.d/common-auth。此文件包含在许多其他服务中（如上述GDM文件）。在那里你也必须添加行auth requisite pam\_vbox.so。
3. 如果使用pam\_unix.so或 pam\_unix2.so希望对镜像数据库进行身份验证， 则需要参数try\_first\_passfor pam\_unix.so或 use\_first\_passfor pam\_unix2.so才能将证书从VirtualBox模块传递到影子数据库认证模块。对于Ubuntu，这需要添加到/etc/pam.d/common-auth引用线的末尾 pam\_unix.so。该参数告诉PAM模块使用堆栈中已经存在的凭据，即VirtualBox PAM模块提供的凭据。

**警告**

配置不正确的PAM堆栈可以有效防止您登录到您的客系统！

为了使部署更容易，您可以debug在pam\_vbox.so语句之后 传递参数 。然后使用syslog记录调试日志输出。

**注意**

默认情况下，pam\_vbox不会等待从主机到达的凭据，换句话说：当显示登录提示时（例如由GDM / KDM或文本控制台）和pam\_vbox尚未拥有凭据，它不会等到他们到达。相反，PAM堆栈中的下一个模块（取决于PAM配置）将有机会进行身份验证。

从VirtualBox 4.1.4开始，pam\_vbox支持各种guest虚拟机属性参数 /VirtualBox/GuestAdd/PAM/。这些参数允许pam\_vbox等待主机提供的凭据，并可选地在等待这些信息时显示消息。可以设置以下宾客属性：

1. CredsWait：如果pam\_vbox应该开始等待，直到凭证从主机到达，则设置为“1”。在此之前，没有其他身份验证方法（如手动登录）将可用。如果此属性为空或被删除，则不会执行等待凭据，并且pam\_vbox将像之前一样执行（参见上面的段落）。该属性必须为guest（RDONLYGUEST）设置为只读。
2. CredsWaitAbort：设置为任何值时等待凭据中止。可以从主机和客人设置。
3. CredsWaitTimeout：超时（以秒为单位）让pam\_vbox等待凭据到达。当此超时期间没有凭据到达时，pam\_vbox的身份验证将被设置为失败，并且将询问链中的下一个PAM模块。如果未指定此属性，则设置为“0”或无效值，将使用无限超时。该属性必须为guest（RDONLYGUEST）设置为只读。

要进一步自定义pam\_vbox，有以下客户属性：

1. CredsMsgWaiting：pam\_vbox正在等待主机的凭据时显示自定义消息。该属性必须为guest（RDONLYGUEST）设置为只读。
2. CredsMsgWaitTimeout：在等待pam\_vbox的凭据超时时显示自定义消息，例如没有在时间内到达。该属性必须为guest（RDONLYGUEST）设置为只读。

**注意**

如果一个pam\_vbox guest属性没有设置正确的flags（RDONLYGUEST），那么这个属性将被忽略，而且 - 根据该属性 - 将设置默认值。这可能导致pam\_vbox不等待凭据。请参阅相应的syslog文件以获取更多信息并使用该 debug选项。

**9.2.2.1。VirtualBox Greeter for Ubuntu / LightDM**

从4.2.12版开始，VirtualBox配有一个名为vbox-greeter的自己的greeter模块，可以与LightDM 1.0.1或更高版本一起使用。LightDM是Ubuntu 10.11以来的默认显示管理器，因此也可以用于自动访客登录。

vbox-greer不需要上面描述的pam\_vbox模块才能运行 - 它具有自己的LightDM提供的认证机制。但是，为了提供最大的灵活性，两个模块可以在同一个客户端一起使用。

对于pam\_vbox模块，vbox-greeter是作为客户添加的一部分提供的，但默认情况下不会在客户机操作系统上安装和/或激活。对于安装客户端安装时自动安装vbox-greep，请--with-autologon在启动VBoxLinuxAdditions时使用该 开关.run文件：

＃./VBoxLinuxAdditions.run - --with-autologon

对于手动或延期安装 ，通常vbox-greeter.desktop 必须将文件复制 /opt/VBoxGuestAdditions-<version>/shared/VBoxGuestAdditions/ 到xgreeters目录中 /usr/share/xgreeters/。请参阅您的客户操作系统文档以获取正确的LightDM greeter目录。

vbox-greeter模块本身已经由VirtualBox Guest Additions安装程序安装，并且位于 /usr/sbin/。要启用vbox-greeter作为标准的greeter模块，/etc/lightdm/lightdm.conf需要对该文件 进行编辑：

[SeatDefaults] 招待员会话= VBOX-招待员

**注意**

LightDM服务器需要完全重新启动，才能使用vbox-greeter作为默认的选择器。作为root用户， service lightdm --full-restart在Ubuntu上执行，或者重新启动guest虚拟机。

**注意**

vbox-greeter独立于用户选择的图形会话（如Gnome，KDE，Unity等）。但是，它需要FLTK 1.3来表示自己的用户界面。

有许多客户属性可用于进一步定制登录体验。要自动登录用户，相同的guest虚拟机属性适用于pam\_vbox，请参见 [第9.2.2节“自动Linux / Unix访客登录”](https://www.virtualbox.org/manual/ch09.html#autologon_unix)。

除了上述访客属性之外，vbox-greeter还允许进一步定制其用户界面。这些特殊的宾客都属于 /VirtualBox/GuestAdd/Greeter/：

1. HideRestart：如果vbox-greeter应隐藏按钮重新启动guest虚拟机，请设置为“1”。该属性必须为guest（RDONLYGUEST）设置为只读。
2. HideShutdown：如果vbox-greeter应隐藏按钮关闭guest虚拟机，请设置为“1”。该属性必须为guest（RDONLYGUEST）设置为只读。
3. BannerPath：将.PNG文件用作顶部横幅的路径。图像尺寸必须为460 x 90像素，任何位深度。该属性必须为guest（RDONLYGUEST）设置为只读。
4. UseTheming：设置为“1”打开以下主题选项。该属性必须为guest（RDONLYGUEST）设置为只读。
5. Theme/BackgroundColor：十六进制RRGGBB颜色为背景。该属性必须为guest（RDONLYGUEST）设置为只读。
6. Theme/LogonDialog/HeaderColor：十六进制RRGGBB头文件的前景色。该属性必须为guest（RDONLYGUEST）设置为只读。
7. Theme/LogonDialog/BackgroundColor：用于登录对话框背景的十六进制RRGGBB颜色。该属性必须为guest（RDONLYGUEST）设置为只读。
8. Theme/LogonDialog/ButtonColor：用于登录对话框按钮的十六进制RRGGBB背景颜色。该属性必须为guest（RDONLYGUEST）设置为只读。

**注意**

对于上述宾客属性的相同限制适用于pam\_vbox部分中指定的。

**9.3。Windows客户机的高级配置**

**9.3.1。自动Windows系统准备**

从Windows NT 4.0开始，Microsoft提供了一个“系统准备”工具（简称为Sysprep），以准备Windows系统进行部署或重新分配。而Windows 2000和XP随安装介质上的Sysprep一起提供，该工具也可以在Microsoft网站上下载。在Windows Vista和7的标准安装中，Sysprep已经包含在内。Sysprep主要由一个可执行程序组成，sysprep.exe用户可以调用这个可执行文件 ，使Windows安装进入准备模式。

从VirtualBox 3.2.2开始，Guest Additions提供了一种以主机系统控制的自动方式在客户机操作系统上启动系统准备的方法。要实现这一点，请参阅 [第4.8节“访客控制”](https://www.virtualbox.org/manual/ch04.html#guestadd-guestcontrol)，使用特殊标识符sysprep作为要执行的程序的功能，以及凭据的用户名 sysprep和密码 sysprep。然后，Sysprep将启动所需的系统权限。

**注意**

指定“sysprep.exe”的位置是**不可能的** - 而是使用以下路径（基于操作系统）：

* C:\sysprep\sysprep.exe 适用于Windows NT 4.0,2.0和XP
* %WINDIR%\System32\Sysprep\sysprep.exe 适用于Windows Vista，2008 Server和7

Guest Additions将自动使用相应的路径来执行系统准备工具。

**9.4。适用于Linux和Solaris guest虚拟机的高级配置**

**9.4.1。在Linux上手动设置所选客服务**

VirtualBox Guest Additions包含几个不同的驱动程序。如果由于任何原因您不希望将其全部设置，您可以使用以下命令安装访客添加：

sh ./VBoxLinuxAdditions.run no\_setup

之后，您至少需要通过运行该命令来编译内核模块

rcvboxadd设置

作为root用户（您将需要通过 *lib64*替换*lib* 64位的guest *虚拟*机），而对于没有udev服务的老客户端，您需要将*vboxadd*服务添加 到默认的运行级别，以确保模块被加载。

要设置时间同步服务，请将服务vboxadd-service添加到默认运行级别。要设置Guest添加的X11和OpenGL部分，请运行命令

rcvboxadd-x11设置

（您不需要为此启用任何服务）。

要重新编译访客内核模块，请使用以下命令：

rcvboxadd设置

编译完成后，您应重新启动客人以确保实际使用新的模块。

**9.4.2。访客图形和鼠标驱动程序的深入设置**

本部分假设您熟悉使用xorg.conf配置X.Org服务器，并且可选地使用hal或udev和xorg.conf.d来更新机制。如果不是，您可以通过研究X.Org附带的文档来了解它们。

VirtualBox Guest Additions附带X.Org版本的驱动程序

* X11R6.8 / X11R6.9和XFree86版本4.3（vboxvideo\_drv\_68.o和vboxmouse\_drv\_68.o）
* X11R7.0（vboxvideo\_drv\_70.so和vboxmouse\_drv\_70.so）
* X11R7.1（vboxvideo\_drv\_71.so和vboxmouse\_drv\_71.so）
* X.Org服务器版本1.3及更高版本（vboxvideo\_drv\_13.so和vboxmouse\_drv\_13.so等）。

默认情况下，可以在目录中找到这些驱动程序

/opt/VBoxGuestAdditions-<version>/lib/VBoxGuestAdditions

并且X服务器的正确版本符号链接到X.Org驱动程序目录中。

为了使图形集成正常工作，X服务器必须加载vboxvideo驱动程序（许多最新的X服务器版本会自动查找它们是否在VirtualBox中运行），并且为了获得最佳用户体验，必须加载访客内核驱动程序，并且客户添加工具VBoxClient必须作为X会话中的客户端运行。为了使鼠标集成正常工作，必须加载来宾内核驱动程序，另外，在X.Org X11R6.8到X11R7.1和XFree86 4.3版本的X服务器中，必须加载正确的vboxmouse驱动程序并将其与/ dev /鼠标或/ dev / psaux; 在X.Org服务器1.3或更高版本中，必须加载PS / 2鼠标的驱动程序，并且必须将正确的vboxmouse驱动程序与/ dev / vboxguest关联。

VirtualBox客户图形驱动程序可以使用虚拟分辨率适合分配给虚拟机的虚拟视频内存（减去客户驱动程序使用的少量）的任何图形配置，如[第3.5节“显示设置”中所述](https://www.virtualbox.org/manual/ch03.html#settings-display)。驱动程序将提供一系列标准模式，至少达到所有活动的客户监视器的默认访客分辨率。在X.Org Server 1.3及更高版本中，可以通过将任何来宾监视器的输出属性VBOX\_MODE设置为“<width> x <height>”来更改默认模式。当VBoxClient和内核驱动程序处于活动状态时，当主机请求更改模式时，这将自动完成。

使用1.3之前的X Server，您还可以将自己的模式添加到X服务器配置文件中。您只需将它们添加到“屏幕”部分的“显示”子部分中的“模式”列表中。例如，此处显示的部分添加了自定义的2048x800分辨率模式：

部分“屏幕” 标识符“默认屏幕” 设备“VirtualBox显卡” 监视器“通用监视器” DefaultDepth 24 子区“显示” 深度24 模式“2048x800”“800x600”“640x480” EndSubSection EndSection

**9.5。CPU热插拔**

使用虚拟机运行现代服务器操作系统，VirtualBox支持CPU热插拔。[[41](https://www.virtualbox.org/manual/ch09.html" \l "ftn.idm7553) ]而在物理计算机上，这意味着在机器运行时可以添加或删除CPU，VirtualBox支持在虚拟机运行时添加和删除虚拟CPU。

CPU热插拔仅适用于支持它的客户机操作系统。到目前为止，这仅适用于Linux和Windows Server 2008 x64数据中心版。Windows仅支持热添加，而Linux支持热添加和热删除，但要使用此功能与8个以上的CPU，需要64位Linux客户机。

此时，CPU热插拔需要使用VBoxManage命令行界面。首先，需要为虚拟机启用热插拔：

VBoxManage modifyvm“VM名称”--cpuhotplug

之后，该--cpus选项指定虚拟机具有的最大CPU数量：

VBoxManage modifyvm“VM名称”--cpus 8

当VM关闭时，您可以使用modifyvm --plugcpu和 --unplugcpu子命令添加和删​​除虚拟CPU，并将虚拟CPU 的编号作为参数，如下所示：

VBoxManage modifyvm“VM名称”--plugcpu 3 VBoxManage modifyvm“VM名称”--unplugcpu 3

请注意，CPU 0永远不能被删除。

当VM运行时，可以使用controlvm plugcpu和 unplugcpu命令添加和删​​除CPU ：

VBoxManage controlvm“VM名称”plugcpu 3 VBoxManage controlvm“VM名称”unplugcpu 3

有关详细信息[，](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)和[第8.13节“VBoxManage controlvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm)。

对于Linux客户机，适用于以下情况：为防止在CPU仍然使用时弹出，必须先从客户机中弹出。Linux Guest Additions包含一个可以接收热删除事件并弹出CPU的服务。此外，在将CPU添加到VM之后，它不会被Linux自动使用。如果已安装，Linux Guest Additions服务将会处理此问题。如果不能使用以下命令启动CPU：

echo 1> / sys / devices / system / cpu / cpu <id> / online

**9.6。PCI通道**

当在Linux主机上运行时，最近有足够的内核（至少版本2.6.31）实验主机PCI设备传输可用。[[42](https://www.virtualbox.org/manual/ch09.html" \l "ftn.idm7577) ]

**注意**

PCI传输模块作为VirtualBox扩展包发货，必须单独安装。有关详细信息[，](https://www.virtualbox.org/manual/ch01.html#intro-installing)请参见[第1.5节“安装VirtualBox和扩展包”](https://www.virtualbox.org/manual/ch01.html#intro-installing)。

实质上，该功能允许客户端直接使用主机上的物理PCI设备，即使主机没有此特定设备的驱动程序。支持普通PCI和一些PCI Express卡。如果AGP和某些PCI Express卡依靠GART（图形地址重映射表）单元编程来进行纹理管理，那么它们不支持某些PCI Express卡，因为它与页面重新映射干扰IOMMU相当不平凡的操作。在以后的版本中可能会解除此限制。

要充分发挥功能，VirtualBox中的PCI通过支持取决于尚不太广泛的IOMMU硬件单元。如果设备使用总线主控（即它自己对OS存储器执行DMA），则需要IOMMU，否则这样的DMA事务可能会写入错误的物理内存地址，因为设备DMA引擎使用设备特定协议执行内存事务。IOMMU用作翻译单元，使用来自客体物理地址的知识到主机物理地址转换规则来从设备映射物理存储器访问请求。

英特尔的IOMMU解决方案是以“Intel Directization I / O”（VT-d）销售的，而AMD公司则称AMD-Vi。所以请检查您的主板数据表是否具有适当的技术。即使您的硬件没有IOMMU，某些PCI卡可能会工作（例如串行PCI适配器），但是客户端将显示启动警告，如果客户驱动程序尝试启用卡总线主控，则VM执行将终止。

默认情况下，BIOS或主机操作系统会禁用IOMMU是非常常见的。所以在尝试使用它之前，请确保

1. 您的主板有一个IOMMU单元。
2. 您的CPU支持IOMMU。
3. IOMMU在BIOS中启用。
4. VM必须使用VT-x / AMD-V运行，并启用嵌套分页。
5. 您的Linux内核是使用IOMMU支持编译的（包括DMA重映射，请参阅CONFIG\_DMAR 内核编译选项）。CONFIG\_PCI\_STUB也需要PCI存根驱动程序（）。
6. 您的Linux内核会识别并使用IOMMU单元（intel\_iommu=on可能需要启动选项）。在内核启动日志中搜索DMAR和PCI-DMA。

一旦确定主机内核支持IOMMU，下一步是选择PCI卡并将其附加到guest虚拟机。要找出可用PCI设备的列表，请使用 lspci命令。输出将如下所示：

01：00.0 VGA兼容控制器：ATI Technologies Inc Cedar PRO [Radeon HD 5450] 01：00.1音频设备：ATI Technologies Inc Manhattan HDMI Audio [Mobility Radeon HD 5000系列] 02：00.0以太网控制器：Realtek半导体有限公司RTL8111 / 8168B PCI Express千兆位 以太网控制器（rev 03） 03：00.0 SATA控制器：JMicron Technology Corp. JMB362 / JMB363串行ATA控制器（rev 03） 03：00.1 IDE接口：JMicron Technology Corp. JMB362 / JMB363串行ATA控制器（rev 03） 06：00.0 VGA兼容控制器：nVidia公司G86 [GeForce 8500 GT]（转a1）

第一列是PCI地址（格式 bus:device.function）。该地址可用于识别用于进一步操作的设备。例如，要将上述系统上的PCI网络控制器连接到客户机中的第二个PCI总线，作为设备5（功能0），请使用以下命令：

VBoxManage modifyvm“VM名称”--pciattach 02：00.0@01：05.0

要拆卸相同的设备，请使用

VBoxManage modifyvm“VM名称”--pcidetach 02：00.0

请注意，主机和客户端都可以自由为运行时附加的卡分配不同的PCI地址，因此这些地址只适用于附件（主机）时的卡的地址，以及BIOS PCI初始化（客户机）期间。

如果虚拟机已连接PCI设备，则会受到某些限制：

1. 目前只支持具有非共享中断的PCI卡（如主机使用MSI）。
2. 无法可靠地保存/恢复客户端状态（由于无法检索到PCI卡的内部状态）。
3. 传送（实时迁移）不起作用（原因相同）。
4. 没有懒惰的物理内存分配。主机将在启动时预先分配虚拟机所需的全部RAM（因为我们无法捕获到物理内存的物理硬件访问）。

**9.7。摄像头直通**

**9.7.1。在访客中使用主机网络摄像头**

VirtualBox 4.3包括一个实验功能，允许客人使用主机网络摄像头。这补充了通用USB传递支持，这是早期版本中使用主机网络摄像机的典型方式。理论上，网络摄像头直通支持可以处理非USB视频源，但这是完全未经测试的。

**注意**

网络摄像头直通模块作为Oracle VM VirtualBox扩展包的一部分提供，必须单独安装。有关详细信息[，](https://www.virtualbox.org/manual/ch01.html#intro-installing)请参见[第1.5节“安装VirtualBox和扩展包”](https://www.virtualbox.org/manual/ch01.html#intro-installing)。

主机网络摄像头可以使用VM菜单栏中的“设备”菜单连接到虚拟机。“网络摄像头”菜单包含主机上可用的视频输入设备的列表。点击网络摄像头名称附加或分离相应的主机设备。

VBoxManage命令行工具可用于启用网络摄像头直通。有关其他详细信息，请参阅下面的主机特定部分。以下命令可用：

* 获取主机网络摄像头（或其他视频输入设备）的列表：

VBoxManage列出网络摄像头

输出格式：

别名“用户友好名称” 主机路径或标识符

别名可以用作其他命令中的快捷方式。别名“0”表示主机上的默认视频输入设备，“.1”，“.2”等意思是第一，第二等视频输入设备。设备顺序是主机专用的。

* 将网络摄像头附加到正在运行的虚拟机：

VBoxManage controlvm“VM名称”webcam attach [host\_path | alias [settings]]

这将附加一个USB网络摄像头设备给客人。

该settings参数是一个字符串 Setting1=Value1;Setting2=Value2，允许配置仿真的摄像头设备。支持以下设置：

* + MaxFramerate将视频帧发送给访客的最高速率。较高的帧速率需要更多的CPU功率。因此，有时设置下限是有用的。默认值是无限制，并允许客人使用主机网络摄像头支持的所有帧速率。
  + MaxPayloadTransferSize仿真网络摄像头可以一次发送给客人的字节数。默认值为3060字节，由某些网络摄像头使用。如果客人能够使用较大的缓冲区，则较高的值可以略微降低CPU负载。但是，MaxPayloadTransferSize 某些客人可能不支持高价。
* 从正在运行的虚拟机中分离出一个网络摄像头：

VBoxManage controlvm“VM名称”webcam detach [host\_path | alias]

* 列出连接到正在运行的虚拟机的网络摄像头：

VBoxManage controlvm“VM名称”网络摄像头列表

输出包含路径或别名，在每个附加的网络摄像头的“webcam attach”命令中使用。

**9.7.2。Windows主机**

当网络摄像头设备与主机分离时，仿真的摄像头设备将自动从客户端分离。

**9.7.3。Mac OS X主机**

需要OS X版本10.9或更高版本。

当网络摄像头设备与主机分离时，仿真的摄像头设备保持附加到客人端，并且必须使用该VBoxManage controlvm "VM name" webcam detach ...命令手动分离 。

**9.7.4。Linux和Solaris主机**

当网络摄像头与主机分离时，只有网络摄像头是流式传输视频，仿真网络摄像头设备才会自动从客户端脱离。如果模拟的网络摄像头处于非活动状态，则应使用该VBoxManage controlvm "VM name" webcam detach ...命令手动分离 。

别名.0并被.1映射到/dev/video0，别名.2被映射到/dev/video1等等。

**9.8。高级显示配置**

**9.8.1。定制VESA分辨率**

除了标准的VESA分辨率，VirtualBox VESA BIOS允许您添加最多16个自定义视频模式，将被报告给客户机操作系统。当使用Windows客户机使用VirtualBox Guest Additions时，将使用自定义图形驱动程序代替后备VESA解决方案，因此此信息不适用。

可以使用额外的数据设施为每个VM配置其他视频模式。额外的数据密钥被调用 CustomVideoMode<x>，x 从1到16的数字。请注意，模式将从1读取，直到未定义以下数字或达到16。以下示例添加了与许多笔记本电脑的本地显示分辨率相对应的视频模式：

VBoxManage setextradata“VM name”“CustomVideoMode1”“1400x1050x16”

自定义视频模式的VESA模式ID从 0x160。为了使用上面定义的定制视频模式，以下命令行已经提供给Linux：

vga = 0x200 | 0x160 vga = 864

对于具有VirtualBox Guest Additions的客户操作系统，可以使用视频模式提示功能设置自定义视频模式。

**9.8.2。使用图形前端配置客人的最大分辨率**

当安装了Guest Additions的guest虚拟机系统使用图形化前端（正常的VirtualBox应用程序）启动时，不允许使用大于主机屏幕尺寸的屏幕分辨率，除非用户通过拖动窗口手动调整大小，切换到完全屏幕或无缝模式或使用VBoxManage发送视频模式提示。这种行为是大多数用户想要的，但如果您有不同的需求，可以通过从命令行发出以下命令之一来更改它：

VBoxManage setextradata全局GUI / MaxGuestResolution any

将删除客户端解决方案的所有限制。

VBoxManage setextradata全局GUI / MaxGuestResolution> width，height <

手动指定最大分辨率。

VBoxManage setextradata全局GUI / MaxGuestResolution auto

恢复默认设置。请注意，这些设置全局适用于所有客户系统，而不仅限于单台机器。

**9.9。高级存储配置**

**9.9.1。使用来宾的原始主机硬盘**

从版本1.4开始，作为使用虚拟磁盘映像的替代方法（如[第5章*虚拟存储中所述*](https://www.virtualbox.org/manual/ch05.html)），VirtualBox还可以将整个物理硬盘或其选定的分区作为虚拟磁盘呈现给虚拟机。

使用VirtualBox，这种类型的访问称为“原始硬盘访问”; 它允许客户机操作系统访问其虚拟硬盘，而无需通过主机操作系统文件系统。图像文件与原始磁盘的实际性能差异取决于主机文件系统的开销，无论是使用动态增长的映像以及主机操作系统缓存策略，都会有很大差异。缓存间接还会影响其他方面，如故障行为，即虚拟磁盘是否包含在主机操作系统崩溃之前写入的所有数据。有关详细信息，请参阅主机操作系统文档。

**警告**

原始硬盘访问仅适用于专家用户。错误使用或使用过时的配置可能会导致物理磁盘上的**数据丢失**。最重要的是，*不要*尝试使用guest虚拟机中当前运行的主机操作系统引导分区。这将导致严重的数据损坏。

原始硬盘访问（对于整个磁盘和单个分区）都是作为VMDK图像格式支持的一部分实现的。因此，您将需要创建一个特殊的VMDK映像文件，该文件定义数据的存储位置。创建这样一个特殊的VMDK映像后，可以像常规的虚拟磁盘镜像那样使用它。例如，您可以使用VirtualBox Manager（[第5.3节“虚拟介质管理器”](https://www.virtualbox.org/manual/ch05.html#vdis)）或VBoxManage将映像分配给虚拟机。

**9.9.1.1。访问整个物理硬盘**

虽然此变体是最简单的设置，但您必须注意，这将使客户机操作系统能够直接和完全访问*整个物理磁盘*。如果您的 *主机*操作系统也从此磁盘启动，请特别注意不要从客户端访问分区。在积极的一面，可以以任意方式重新分配物理磁盘，而无需重新创建允许访问原始磁盘的映像文件。

要创建一个表示整个物理硬盘的映像（不包含任何实际数据，因为这将全部存储在物理磁盘上），请在Linux主机上使用该命令

VBoxManage internalcommands createrawvmdk -filename /path/to/file.vmdk -rawdisk / dev / sda

这将创建图像 /path/to/file.vmdk（必须是绝对的），并且所有数据将被读取和写入/dev/sda。

在Windows主机上，而不是上述设备规范，请使用eg \\.\PhysicalDrive0。在Mac OS X主机上，而不是使用上述设备规范/dev/disk1。请注意，在OS X上，如果没有从其中装载卷，则只能访问整个磁盘。

创建映像需要给定设备的读/写访问。在使用虚拟机中的图像时，还需要读/写访问。在某些主机平台（例如Windows Vista和更高版本）上，原始磁盘访问可能受到限制，主机操作系统在某些情况下不允许。

就像使用常规磁盘映像一样，这不会将新创建的映像自动附加到虚拟机。这可以用例如

VBoxManage storageattach WindowsXP --storagectl“IDE控制器” --port 0 --device 0 --type hdd --medium /path/to/file.vmdk

完成后，所选的虚拟机将从指定的物理磁盘引导。

**9.9.1.2。访问单个物理硬盘分区**

这种“原始分区支持”与上述“完整硬盘”访问非常相似。但是，在这种情况下，任何分区信息都将存储在VMDK映像中，因此您可以在虚拟硬盘中安装不同的引导加载程序，而不会影响主机的分区信息。虽然访客将能够*看到*物理磁盘上存在的所有分区，但访问将从那些不允许访问的分区中读取，分区将只会产生零，并且对它们的所有写入都将被忽略。

要为原始分区支持创建一个特殊图像（其中包含少量数据，如前所述），请在Linux主机上使用命令

VBoxManage internalcommands createrawvmdk -filename /path/to/file.vmdk -rawdisk / dev / sda -partitions 1,5

如您所见，该命令与“完整硬盘”访问权限相同，但附加-partitions参数除外 。此示例将创建映像/path/to/file.vmdk（再次必须是绝对映像），并且访问者/dev/sda 可以访问其分区1和5 。

VirtualBox使用与Linux主机相同的分区编号。因此，上述示例中给出的数字将分别引用扩展分区中的第一个主分区和第一个逻辑驱动器。

在Windows主机上，而不是上述设备规范，请使用eg \\.\PhysicalDrive0。在Mac OS X主机上，而不是使用上述设备规范/dev/disk1。请注意，在OS X上，您只能使用未安装的分区（首先弹出相应的卷）。Linux，Windows和Mac OS X主机上的分区号是相同的。

分区列表的数字可以从输出中获取

VBoxManage internalcommands listpartitions -rawdisk / dev / sda

输出列出了分区类型和大小，为用户提供足够的信息来识别客人所需的分区。

允许访问单个分区的映像特定于特定的主机磁盘设置。您不能将这些图像传输到另一个主机; 此外，每当主机分区更改时，都*必须重新创建*映像 。

创建映像需要给定设备的读/写访问。在使用虚拟机中的图像时，还需要读/写访问。如果这是不可行的，则有一个特殊的变体，用于原始分区访问（目前仅在Linux主机上可用），避免必须给当前用户访问整个磁盘。要设置这样的图像，请使用

VBoxManage internalcommands createrawvmdk -filename /path/to/file.vmdk -rawdisk / dev / sda -partitions 1,5相关

当从虚拟机使用时，映像将不会引用到整个磁盘，而只能指向单个分区（在示例 /dev/sda1和/dev/sda5）中。因此，仅对受影响的分区而不是整个磁盘需要读/写访问。在创建期间，只需访问整个磁盘即可获取分区信息。

在某些配置中，可能需要更改创建的映像的MBR代码，例如替换另一个引导加载程序在主机上使用的Linux引导加载程序。这允许客户端直接启动到Windows，而主机从“相同”磁盘启动Linux。为此，-mbr提供了 参数。它指定从中获取MBR代码的文件名。分区表根本不被修改，因此可以使用来自具有完全不同分区的系统的MBR文件。这方面的一个例子

VBoxManage internalcommands createrawvmdk -filename /path/to/file.vmdk -rawdisk / dev / sda -partitions 1,5 -mbr winxp.mbr

修改的MBR将存储在映像内，而不是存储在主机磁盘上。

创建的映像可以像往常一样在虚拟机配置中附加到存储控制器。

**9.9.2。配置硬盘供应商产品数据（VPD）**

VirtualBox报告其虚拟硬盘的供应商产品数据，其中包括硬盘序列号，固件版本和型号。可以使用以下命令更改这些：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / SerialNumber”“serial” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / FirmwareRevision”“固件” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / ModelNumber”“model”

序列号是20字节的字母数字字符串，固件版本为8字节字母数字字符串，型号为40字节字母数字字符串。代替“Port0”（指第一个端口），指定所需的SATA硬盘端口。

上述命令适用于具有AHCI（SATA）控制器的虚拟机。具有IDE控制器的虚拟机的命令是：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / piix3ide / 0 / Config / PrimaryMaster / SerialNumber”“serial” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / piix3ide / 0 / Config / PrimaryMaster / FirmwareRevision”“固件” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / piix3ide / 0 / Config / PrimaryMaster / ModelNumber”“model”

对于硬盘，也可以将驱动器标记为具有非旋转介质：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / NonRotational”“1”

CD / DVD驱动器需要额外的三个参数来报告供应商的产品数据：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / ATAPIVendorId”“vendor” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / ATAPIProductId”“product” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / Config / Port0 / ATAPIRevision”“revision”

供应商ID是一个8字节的字母数字字符串，产品ID为16字节字母数字字符串，修订版为4字节字母数字字符串。代替“Port0”（指第一个端口），指定所需的SATA硬盘端口。

**9.9.3。通过内部网络访问iSCSI目标**

作为实验功能，VirtualBox允许访问在配置为使用内部网络模式的虚拟机中运行的iSCSI目标。请参见[第5.10节“iSCSI服务器”](https://www.virtualbox.org/manual/ch05.html#storage-iscsi) ; [第6.6节“内部联网”](https://www.virtualbox.org/manual/ch06.html#network_internal) ; 和[第8.18节“VBoxManage storageattach”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storageattach)获取更多信息。

必须在访问iSCSI目标的虚拟机中配置访问内部网络的IP协议栈。必须选择其他虚拟机不使用的免费静态IP和MAC地址。在下面的示例中，根据需要调整虚拟机的名称，MAC地址，IP配置和内部网络名称（“MyIntNet”）。必须首先发出以下8个命令：

VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / Trusted 1 VBoxManage setextradata“VM name”VBoxInternal / Devices / IntNetIP / 0 / Config / MAC 08：00：27：01：02：0f VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / Config / IP 10.0.9.1 VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / Config /网络掩码255.255.255.0 VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / LUN＃0 /驱动程序IntNet VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / LUN＃0 / Config / Network MyIntNet VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / LUN＃0 / Config / TrunkType 2 VBoxManage setextradata“VM名称”VBoxInternal / Devices / IntNetIP / 0 / LUN＃0 / Config / IsService 1

最后，iSCSI磁盘必须附有 --intnet选项，让iSCSI启动器使用内部网络：

VBoxManage storageattach ... --medium iscsi --server 10.0.9.30 --target iqn.2008-12.com.sun：sampletarget --intnet

与“常规”iSCSI设置相比，目标的IP地址 *必须*指定为数字IP地址，因为没有用于内部网络的DNS解析器。

具有iSCSI目标的虚拟机应在启动VM之前启动。如果使用iSCSI磁盘的虚拟机在iSCSI目标未上电的情况下启动，则最多可能需要200秒才能检测到此情况。虚拟机将无法启动。

**9.10。使用串行端口的传统命令**

从版本1.4开始，VirtualBox提供了对虚拟串行端口的支持，当时，使用一系列VBoxManage setextradata 语句进行设置是相当复杂的。从版本1.5开始，不再需要和*不建议使用*串口设置方式*。*要设置虚拟串行端口，请使用[第3.9节“串行端口”中](https://www.virtualbox.org/manual/ch03.html#serialports)描述的方法。

**注意**

为了向后兼容，setextradata从旧版本的手册中保留以下描述的旧 语句 *优先*于配置串行端口的新方式。因此，如果配置串行端口的新方式不起作用，请确保相关的VM没有旧的配置数据，如下面仍然有效。

使用以下6个命令配置串口的旧序列：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / Config / IRQ”4 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / Config / IOBase”0x3f8 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / LUN＃0 / Driver”Char VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / LUN＃0 / AttachedDriver / Driver”NamedPipe VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / LUN＃0 / AttachedDriver / Config / Location”“\\。\ pipe \ vboxCOM1” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / serial / 0 / LUN＃0 / AttachedDriver / Config / IsServer”1

这将在客户机中设置COM1（IRQ 4，I / O地址0x3f8）的默认设置的串行端口，并且该 Location设置假定此配置在Windows主机上使用，因为使用Windows命名管道语法。请记住，在Windows主机上，命名管道必须始终以\\.\pipe\。在Linux上，应用相同的配置设置，除了Location可以更自由地选择路径名 。本地域套接字可以放置在任何地方，只要运行VirtualBox的用户有权在目录中创建新的文件。上面的最后一个命令定义了VirtualBox作为服务器，即它创建了命名管道本身，而不是连接到已经存在的管道本身。

**9.11。微调VirtualBox NAT引擎**

**9.11.1。配置NAT网络接口的地址**

在NAT模式下，客户端网络接口10.0.x.0/24默认分配给IPv4范围， x对应于NAT接口+2的实例。那么x当只有一个NAT实例被激活时就是2。在这种情况下，客人被分配到地址10.0.2.15，网关设置为10.0.2.2并且可以找到名称服务器10.0.2.3。

如果由于任何原因需要更改NAT网络，可以通过以下命令来实现：

VBoxManage modifyvm“VM名称”--natnet1“192.168 / 16”

此命令会从预留的网络地址 192.168.0.0，以 192.168.254.254供“虚拟机名称”的第一NAT网络实例。访客IP将被分配到 192.168.0.15并且可以找到默认网关192.168.0.2。

**9.11.2。配置NAT网络接口的引导服务器（下一个服务器）**

对于以NAT模式进行网络引导，默认情况下VirtualBox使用IP地址为10.0.2.4的内置TFTP服务器。对于典型的远程启动方案，此默认行为应该正常工作。但是，可以使用以下命令更改引导服务器IP和引导映像的位置：

VBoxManage modifyvm“VM名称”--nattftpserver1 10.0.2.2 VBoxManage modifyvm“VM名称”--nattftpfile1 /srv/tftp/boot/MyPXEBoot.pxe

**9.11.3。调整NAT的TCP / IP缓冲区**

VirtualBox NAT堆栈性能通常由其与主机的TCP / IP协议栈和几个缓冲区（SO\_RCVBUF和 SO\_SNDBUF）的大小的交互来决定。对于某些设置，用户可能希望调整缓冲区大小以获得更好的性能。这可以通过使用以下命令实现（值为千字节，范围为8到1024）：

VBoxManage modifyvm“VM名称”--natsettings1 16000,128,128,0,0

此示例说明了调整NAT设置。第一个参数是MTU，然后是套接字发送缓冲区的大小，套接字接收缓冲区的大小，TCP发送窗口的初始大小，以及最后TCP接收窗口的初始大小。请注意，指定零表示回退到默认值。

每个缓冲区的默认大小为64KB，默认的MTU为1500。

**9.11.4。将NAT套接字绑定到特定接口**

默认情况下，VirtualBox的NAT引擎将通过主机的TCP / IP协议栈分配的默认接口路由TCP / IP数据包。（其技术原因是NAT引擎使用套接字进行通信。）如果由于某些原因想要更改此行为，可以通知NAT引擎绑定到特定的IP地址。使用以下命令：

VBoxManage modifyvm“VM名称”--natbindip1“10.45.0.2”

此后，所有出站流量将通过IP地址10.45.0.2的接口发送。请确保此接口在此分配之前已启动并运行。

**9.11.5。在NAT模式下启用DNS代理**

默认情况下，NAT引擎为主机上配置的guest虚拟机提供相同的DNS服务器。在某些情况下，可能希望从客户端隐藏DNS服务器IP，例如，由于DHCP租约过期，此信息可能在主机上更改。在这种情况下，您可以使用以下命令告诉NAT引擎作为DNS代理：

VBoxManage modifyvm“VM名称”--natdnsproxy1 on

**9.11.6。在NAT模式下使用主机的解析器作为DNS代理**

为了解决网络名称，NAT引擎的DHCP服务器提供主机的注册DNS服务器列表。如果由于某种原因，您需要隐藏此DNS服务器列表并使用主机的解析器设置，从而强制VirtualBox NAT引擎拦截DNS请求并将其转发到主机的解析器，请使用以下命令：

VBoxManage modifyvm“VM名称”--natdnshostresolver1 on

请注意，此设置与DNS代理模式类似，但是代理模式只将DNS请求转发到相应的服务器，解析器模式将解释DNS请求，并使用主机的DNS API查询信息并将其返回给客户。

**9.11.6.1。用户定义的主机名解析**

在某些情况下，拦截名称解析机制可能有用，在特定DNS请求上提供用户定义的IP地址。截取机制允许用户不仅映射单个主机，而且还需要映射域，甚至更复杂的命名约定。

以下命令设置将名称映射到指定IP的规则：

VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / {pcnet，e1000} / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ <拦截规则的唯一规则名称> / HostIP“<IPv4> VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / {pcnet，e1000} / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ <唯一规则名称> / HostName“<主机名>>

以下命令设置将模式名称映射到指定IP的规则：

VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / {pcnet，e1000} / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ <唯一规则名称> / HostIP“<IPv4> VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / {pcnet，e1000} / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ <唯一规则名称> / HostNamePattern“<hostpattern>

主机模式可以包括"|", "?" and "\*"。

此示例演示如何指导主机解析器机制来解析所有域，并且可能会使用IP 127.0.0.1解析www.blocked-site.info站点的一些镜像：

VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / e1000 / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ all\_blocked\_site / HostIP“127.0.0.1 VBoxManage setextradata“VM名称”\ “VBoxInternal / Devices / e1000 / 0 / LUN＃0 / AttachedDriver / Config / HostResolverMappings / \ all\_blocked\_site / HostNamePattern“”\* .blocked-site。\* | \* .fb.org“

应该启用主机解析器机制以使用用户定义的映射规则，否则它们没有任何影响。

**9.11.7。配置NAT引擎的别名**

默认情况下，NAT核心使用别名，并在为连接生成别名时使用随机端口。这对于大多数协议（如SSH，FTP等）都很有效。虽然一些协议可能需要更透明的行为，或者可能依赖于发送数据包的实际端口号。可以通过以下命令通过VBoxManage前端更改NAT模式：

VBoxManage modifyvm“VM名称”--nataliasmode1代理

和

VBoxManage modifyvm“Linux Guest”--nataliasmode1手册

第一个示例禁用别名并将NAT切换到透明模式，第二个示例强制保留端口值。如果需要，可以组合这些模式。

**9.12。配置BIOS DMI信息**

VirtualBox提供给客户的DMI数据可以针对特定虚拟机进行更改。使用以下命令配置DMI BIOS信息。如果你的虚拟机配置为使用EFI固件需要更换pcbios由efi在钥匙。

DMI BIOS信息

（类型0）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSVendor”“BIOS Vendor” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSVersion”“BIOS版本” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSReleaseDate”“BIOS发布日期” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSReleaseMajor”1 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSReleaseMinor”2 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSFirmwareMajor”3 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBIOSFirmwareMinor”4

DMI系统信息

（1型）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemVendor”“系统供应商” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemProduct”“系统产品” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemVersion”“系统版本” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemSerial”“系统串行” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemSKU”“系统SKU” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemFamily”“系统族” VBoxManage setextradata“VM名称” “VBoxInternal /设备/ PCBIOS / 0 /配置/ DmiSystemUuid” “9852bf98-b83c-49分贝-a8de-182c42c7226b”

DMI板信息

（类型2）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardVendor”“Board Vendor” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardProduct”“板卡产品” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardVersion”“板版” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardSerial”“Board Serial” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardAssetTag”“Board Tag” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardLocInChass”“Board Location” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiBoardBoardType”10

DMI系统机箱或机箱

（3型）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiChassisVendor”“机箱供应商” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiChassisType”3 VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiChassisVersion”“机箱版本” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiChassisSerial”“机箱串行” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiChassisAssetTag”“机箱标签”

DMI处理器信息

（4型）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiProcManufacturer”“GenuineIntel” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiProcVersion”“Pentium（R）III”

DMI OEM字符串

（类型11）

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiOEMVBoxVer”“vboxVer\_1.2.3” VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiOEMVBoxRev”“vboxRev\_12345”

如果未设置DMI字符串，则使用默认值VirtualBox。设置一个空字符串使用 "<EMPTY>"。

请注意，在上面的列表中，所有引用的参数（DmiBIOSVendor，DmiBIOSVersion，但不是DmiBIOSReleaseMajor）都是字符串。如果这样的字符串是有效的数字，则该参数被视为数字，并且VM最有可能拒绝从VERR\_CFGM\_NOT\_STRING错误开始 。在这种情况下，"string:<value>"例如使用

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / pcbios / 0 / Config / DmiSystemSerial”“string：1234”

可能需要更改此信息才能将主机的DMI信息提供给访客，以防止Windows要求新的产品密钥。在Linux主机上，可以获得DMI BIOS信息

dmidecode -t0

并可以获得DMI系统信息

dmidecode -t1

**9.13。配置自定义ACPI表**

VirtualBox可以配置为向guest虚拟机呈现自定义ACPI表。使用以下命令配置：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / acpi / 0 / Config / CustomTable”“/path/to/table.bin”

配置自定义ACPI表可能会阻止Windows Vista和Windows 7要求新的产品密钥。在Linux主机上，可以读取其中一个主机表 /sys/firmware/acpi/tables/。

**9.14。微调定时器和时间同步**

**9.14.1。配置访客时间戳计数器（TSC）以反映客人执行**

默认情况下，VirtualBox将所有时间的可见时间保持在同一时间源的单一时间源上，这是单调的主机时间。这反映了许多客户操作系统的假设，这些系统期望所有时间来源反映“挂钟”时间。在特殊情况下，使访客中的TSC（时间戳计数器）反映实际执行客人的时间可能是有用的。

可以在每个虚拟机的基础上启用这种特殊的TSC处理模式，并且只有结合硬件虚拟化才能获得最佳结果。要启用此模式，请使用以下命令：

VBoxManage setextradata“VM名称”“VBoxInternal / TM / TSCTiedToExecution”1

要恢复为默认的TSC处理模式，请使用：

VBoxManage setextradata“VM name”“VBoxInternal / TM / TSCTiedToExecution”

请注意，如果您使用特殊的TSC处理模式，客户机操作系统对时间源的一致性非常严格，可能会收到关于时序不一致的警告或错误消息。根据使用TSC的方式，还可能会使某些客户机操作系统的时钟变得不可靠。

**9.14.2。加速或减慢客人时钟**

为了某些目的，加速或减慢（虚拟）客机时钟可能是有用的。这可以实现如下：

VBoxManage setextradata“VM名称”“VBoxInternal / TM / WarpDrivePercentage”200

上述示例将使客人时钟的速度加倍

VBoxManage setextradata“VM name”“VBoxInternal / TM / WarpDrivePercentage”50

将客人时钟的速度减半。请注意，更改虚拟时钟的速率可能会使客户混淆，甚至可能导致客户异常行为。例如，较高的时钟速率意味着虚拟设备的超时时间更短，结果是由于主机负载增加，虚拟设备的响应时间略有增加，可能导致客户机故障。还要注意，任何时间同步机制都会频繁地尝试将来宾时钟与参考时钟（如果VirtualBox Guest Additives处于活动状态时是主机时钟）重新同步。因此，如果如上所述改变来宾时钟的速率，则任何时间同步都应被禁用（见[第9.14.3节，](https://www.virtualbox.org/manual/ch09.html#changetimesync)

**9.14.3。调整Guest添加时间同步参数**

VirtualBox Guest Additives可确保客人的系统时间与主机时间同步。有几个可以调整的参数。可以使用以下命令为特定VM设置参数：

VBoxManage guestproperty设置“VM名称”“/ VirtualBox / GuestAdd / VBoxService / PARAMETER”VALUE

哪里PARAMETER是以下之一：

--timesync-interval

指定与主机同步时间的时间间隔。默认值为10000 ms（10秒）。

--timesync-min-adjust

以毫秒为单位测量的最小绝对漂移值进行调整。默认情况下，OS / 2为1000 ms，其他地方为100 ms。

--timesync-latency-factor

乘以时间查询延迟的因素，以计算动态最小调整时间。默认值为8次，这意味着详细：测量确定主机时间所需的时间（客户机必须联系可能需要一些时间的VM主机服务），将该值乘以8，并且只有在主机与来宾之间的时差大于此值。不要做任何时间的调整。

--timesync-max-latency

最大主机定时器查询延迟接受。默认值为250 ms。

--timesync-set-threshold

绝对漂移阈值，以毫秒为单位给出，从而开始设置时间，而不是尝试平滑调整它。默认为20分钟。

--timesync-set-start

设置启动时间同步服务的时间。

--timesync-set-on-restore 0|1

在将1作为参数（默认值）设置为从保存状态恢复VM后的时间。通过0禁用。在后一种情况下，时间将顺利调整，这可能需要很长时间。

所有这些参数也可以指定为VBoxService的命令行参数。

**9.14.4。禁用客户添加时间同步**

一旦安装并启动，VirtualBox Guest Additions将尝试将访客时间与主机时间同步。这可以通过禁止访客服务读取主机时钟来防止：

VBoxManage setextradata“VM名称”“VBoxInternal / Devices / VMMDev / 0 / Config / GetHostTimeDisabled”1

**9.15。在Solaris 11主机上安装备用桥接网络驱动程序**

从VirtualBox 4.1开始，VirtualBox发布了一个新的网络过滤器驱动程序，它使用Solaris 11的Crossbow功能。默认情况下，为已支持的Solaris 11主机（构建159及更高版本）安装了此新驱动程序。

要强制安装旧的基于STREAMS的网络过滤器驱动程序，请在安装VirtualBox软件包之前以root身份执行以下命令：

触摸/ etc / vboxinst\_vboxflt

要强制安装基于Crossbow的网络过滤器驱动程序，请在安装VirtualBox软件包之前以root身份执行以下命令：

触摸/ etc / vboxinst\_vboxbow

要检查VirtualBox当前正在使用哪个驱动程序，请执行：

modinfo | grep vbox

如果输出包含“vboxbow”，则表示VirtualBox正在使用Crossbow网络过滤器驱动程序，名称“vboxflt”表示旧版STREAMS网络过滤器的使用。

**9.16。用于Solaris 11主机上的VLAN的VirtualBox VNIC模板**

VirtualBox支持用于通过VLAN配置虚拟机的VNIC（虚拟网络接口）模板。[[43](https://www.virtualbox.org/manual/ch09.html" \l "ftn.idm7980) ] VirtualBox VNIC模板是VNIC，其名称以“vboxvnic\_template”开头（区分大小写）。

在Solaris 11主机[[44](https://www.virtualbox.org/manual/ch09.html" \l "ftn.idm7983) ]上，可以使用VNIC模板来指定通过网络链路桥接时要使用的VLAN ID。

以下是如何使用VNIC模板通过VLAN配置虚拟机的示例。通过以root身份执行，创建VirtualBox VNIC模板：

dladm create-vnic -t -l nge0 -v 23 vboxvnic\_template0

这将通过VLAN ID为“nge0”的接口创建一个临时VNIC模板。要创建跨主机重新启动持久化的VNIC模板，请跳过-t上述命令中的参数。您可以使用以下方式查看链接的当前状态：

$ dladm show-link 连接类MTU状态桥 nge0 phys 1500 up - - nge1 phys 1500 down - vboxvnic\_template0 vnic 1500 up - nge0 $ dladm show-vnic 链接超速MACADDRTYPE VID vboxvnic\_template0 nge0 1000 2：8：20：25：12：75随机的23

VNIC模板创建完成后，任何需要通过“nge0”接口在VLAN 23上的虚拟机都可以配置为使用该VNIC模板桥接。

VNIC模板使VLAN中的虚拟机更简单，高效。VLAN详细信息不作为每个虚拟机配置的一部分存储，而是在启动虚拟机时从VNIC模板继承。VNIC模板本身随时可以修改dladm。

可以使用附加属性（如带宽限制，CPU扇出等）创建VNIC模板。有关如何完成此操作，请参阅Solaris网络文档。这些附加属性（如果有）也适用于使用VNIC模板桥接的VM。

**9.17。在Solaris主机上配置多个仅限主机的网络接口**

默认情况下，VirtualBox为您提供一个仅主机的网络接口。在Solaris主机上添加更多主机专用网络接口需要手动配置。以下是如何添加另一个仅主机的网络接口。

首先停止所有正在运行的虚拟机。然后，通过以root身份执行以下命令来解除现有的“vboxnet0”接口：

ifconfig vboxnet0 unplumb

如果你有几个vboxnet接口，那么你将需要解除所有这些接口。一旦所有的vboxnet接口都被取消，通过以root身份执行以下命令来删除驱动程序：

rem\_drv vboxnet

编辑文件/platform/i86pc/kernel/drv/vboxnet.conf 并为要添加的新界面添加一行，如下所示：

name =“vboxnet”parent =“pseudo”instance = 1; name =“vboxnet”parent =“pseudo”instance = 2;

根据需要添加这些行，每行具有唯一的实例编号。

接下来，通过以root身份执行以下命令来重新加载vboxnet驱动程序：

add\_drv vboxnet

在Solaris 11.1和更新的主机上，您可能需要重命名默认的虚拟化接口名称。要检查已分配的名称，请执行：

dladm show-phys 链接媒体状态速度双工器件 net0 Ethernet up 100 full e1000g0 net2 Ethernet up 1000 full vboxnet1 net1 Ethernet up 1000 full vboxnet0

在上面的例子中，我们可以将“net2”重命名为“vboxnet1”，然后再继续进行接口。这可以通过以root身份执行：

dladm rename-link net2 vboxnet1

现在使用所有接口 ifconfig vboxnetX plumb（在这种情况下，'X'将为1）。一旦接口被连接，它可以像任何其他网络接口一样配置。ifconfig有关详细信息，请参阅 文档。

要使重新启动时新添加的界面的设置保持不变，您将需要编辑文件 /etc/inet/netmasks，如果您正在使用NWAM /etc/nwam/llp并添加相应的条目来为每个接口设置网络掩码和静态IP。默认情况下，VirtualBox安装程序仅更新其创建的“vboxnet0”接口的这些配置文件。

**9.18。在Solaris主机上配置VirtualBox CoreDumper**

当出现问题时，VirtualBox能够生成自己的核心文件进行大量调试。目前这仅适用于Solaris主机。

可以使用以下命令启用VirtualBox CoreDumper：

VBoxManage setextradata“VM名称”VBoxInternal2 / CoreDumpEnabled 1

您可以使用此命令指定要用于核心转储的目录：

VBoxManage setextradata“VM名称”VBoxInternal2 / CoreDumpDir <path-to-directory>

确保您指定的目录在具有足够可用空间的卷上，并且VirtualBox进程具有足够的权限将文件写入此目录。如果您跳过此命令并且不指定任何核心转储目录，则将使用VirtualBox可执行文件的当前目录（在使用root权限保护时编写内核时，这很可能会失败）。建议您明确设置一个核心转储目录。

您必须指定何时触发VirtualBox CoreDumper。这是使用以下命令完成的：

VBoxManage setextradata“VM名称”VBoxInternal2 / CoreDumpReplaceSystemDump 1 VBoxManage setextradata“VM名称”VBoxInternal2 / CoreDumpLive 1

如果您启用了VirtualBox CoreDumper，则必须提供上述两个命令中的至少一个。

设置CoreDumpReplaceSystemDump 设置VM以覆盖主机的核心转储机制，并且在任何崩溃的情况下，只有VirtualBox CoreDumper将生成核心文件。

CoreDumpLive每当VM进程收到SIGUSR2信号时，设置设置虚拟机以产生核心 。生成核心文件后，VM将不会被终止，并将继续运行。因此，您可以使用以下方式获取VM进程的核心：

kill -s SIGUSR2 <VM-process-id>

core.vb.<ProcessName>.<ProcessID>例如，VirtualBox CoreDumper生成的核心文件的形式 core.vb.VBoxHeadless.11321。

**9.19。VirtualBox和Solaris内核区域**

基于x86的系统上的Solaris内核区域利用了诸如VirtualBox之类的硬件辅助虚拟化功能。但是，对于内核区域和VirtualBox来共享这个硬件资源，他们需要合作。

默认情况下，由于性能原因，VirtualBox在主机上全局获取了硬件辅助虚拟化资源（VT-x / AMD-V），并将其使用，直到需要断电的最后一个VirtualBox VM。这可以防止其他软件在VirtualBox控制之前使用VT-x / AMD-V。

当不执行访客代码时，可以指示VirtualBox放弃使用硬件辅助虚拟化功能，从而允许内核区域利用它们。为此，请关闭所有VirtualBox VM并执行以下命令：

VBoxManage setproperty hwvirtexclusive关闭

该命令只需要执行一次，因为设置作为全局VirtualBox设置的一部分存储，这些设置将在主机重新启动和VirtualBox升级之后继续保持不变。

**9.20。锁定VirtualBox GUI**

**9.20.1。自定义VM管理器**

有几个高级自定义设置用于锁定VirtualBox管理器，也就是删除用户不应该看到的某些功能。

VBoxManage setextradata全局GUI /自定义OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

noSelector

不允许启动VirtualBox管理器。尝试这样做将显示一个包含正确错误消息的窗口。

noMenuBar

VM窗口不会包含菜单栏。

noStatusBar

VM窗口不会包含状态栏。

禁用任何这些VM管理器自定义

VBoxManage setextradata全局GUI /自定义

**9.20.2。VM选择器自定义**

以下每台计算机虚拟机数据库设置可用于更改关于某些虚拟机的虚拟机选择器窗口的行为：

VBoxManage setextradata“VM name”设置为true

哪里SETTING可以：

GUI/HideDetails

不显示某个VM的VM配置。如果选择此VM，则详细信息窗口将保持为空。

GUI/PreventReconfiguration

不允许用户打开特定虚拟机的设置对话框。

GUI/PreventSnapshotOperations

在运行时或VM关闭电源时，防止来自GUI的虚拟机的快照操作。

GUI/HideFromManager

在VM选择器窗口中隐藏某个VM。

GUI/PreventApplicationUpdate

禁用自动更新检查并隐藏相应的菜单项。

请注意，这些设置不会阻止用户重新配置虚拟机VBoxManage modifyvm。

**9.20.3。配置VM选择器菜单条目**

您可以在VM选择器的全局设置页面中禁用（即黑名单）某些条目：

VBoxManage setextradata全局GUI / RestrictedGlobalSettingsPages OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

General

不显示*常规*设置窗格。

Input

不显示*输入*设置窗格。

Update

不显示*更新*设置窗格。

Language

不显示*语言*设置窗格。

Display

不显示“ *显示*设置”窗格。

Network

不显示*网络*设置窗格。

Extensions

不显示*扩展*设置窗格。

Proxy

不显示*代理*设置窗格。

这是一个全球性的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata全局GUI / RestrictedGlobalSettingsPages

**9.20.4。配置虚拟机窗口菜单项**

您可以在VM窗口中禁用（即黑名单）某些菜单操作：

VBoxManage setextradata“VM name”GUI / RestrictedRuntimeMenus OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

All

不要在虚拟机窗口中显示任何菜单。

Machine

不要在虚拟机窗口中显示*机器*菜单。

View

不要在VM窗口中显示“ *视图”*菜单。

Devices

不要在虚拟机窗口中显示*设备*菜单。

Help

不要在虚拟机窗口中显示*帮助*菜单。

Debug

不要在虚拟机窗口中显示*调试*菜单。调试菜单仅在使用特殊命令行参数或环境变量设置启动GUI时可见。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeMenus

您还可以禁用（即黑名单）某些菜单的某些菜单操作。使用以下命令禁用*应用程序*菜单的某些操作 （仅适用于Mac OS X主机）：

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeApplicationMenuActions OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

All

不要在此菜单中显示任何菜单项。

About

不要在此菜单中显示*关于*菜单项。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeMenus

使用以下命令禁用“ *机器”* 菜单的某些操作：

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeApplicationMenuActions OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

All

不要在此菜单中显示任何菜单项。

SettingsDialog

不要在此菜单中显示*设置*菜单项。

TakeSnapshot

不要在此菜单中显示*拍摄快照*菜单项。

TakeScreenshot

不要在此菜单中显示“ *拍摄”屏幕截图*菜单项。

InformationDialog

不要在此菜单中显示*会话信息*菜单项。

MouseIntegration

不要在此菜单中显示“ *禁用鼠标集成”*菜单项。

TypeCAD

不要在此菜单中显示*插入Ctrl + Alt + Del*菜单项。

TypeCABS

不要在此菜单中显示*插入Ctrl + Alt + Backspace*菜单项（仅适用于X11主机）。

Pause

不要在此菜单中显示*暂停*菜单项。

Reset

不要在此菜单中显示*重置*菜单项。

SaveState

不要在此菜单中显示*保存机器状态*菜单项。

Shutdown

不要在此菜单中显示*ACPI关机*菜单项。

PowerOff

不要在本菜单中显示*机器*菜单项的*电源*。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeApplicationMenuActions

使用以下命令禁用“ *视图”* 菜单的某些操作：

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeViewMenuActions OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

All

不要在此菜单中显示任何菜单项。

Fullscreen

不要在此菜单中显示*切换到全屏*菜单项。

Seamless

不要在此菜单中显示*切换到无缝模式*菜单项。

Scale

不要在此菜单中显示*切换到缩放模式*菜单项。

GuestAutoresize

在此菜单中不显示*自动调整客人显示*菜单项。

AdjustWindow

不要在此菜单中显示*调整窗口大小*菜单项。

Multiscreen

不要在此菜单中显示*Multiscreen*菜单项（仅在全屏/无缝模式下可见）。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeViewMenuActions

使用以下命令禁用“ *视图”* 菜单的某些操作：

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeDevicesMenuActions OPTION [，OPTION ...]

这里OPTION是以下关键字来禁用的行动之一*设备*菜单：

All

不要在此菜单中显示任何菜单项。

OpticalDevices

不要在此菜单中显示*CD / DVD设备*菜单项。

FloppyDevices

不要在此菜单中显示*FLoppy设备*菜单项。

USBDevices

不要在此菜单中显示*USB设备*菜单项。

SharedClipboard

不要在此菜单中显示*共享剪贴板*菜单项。

DragAndDrop

不要在此菜单中显示*拖放*菜单项。

NetworkSettings

不要在此菜单中显示*网络设置...*菜单项。

SharedFoldersSettings

不要在此菜单中显示*共享文件夹设置...*菜单项。

VRDEServer

不要在此菜单中显示*删除显示*菜单项。

InstallGuestTools

不要在此菜单中显示*插入访客添加CD imnage ...*菜单项。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeDevicesMenuActions

使用以下命令禁用“ *视图”* 菜单的某些操作：

VBoxManage setextradata“VM name”GUI / RestrictedRuntimeDebuggerMenuActions OPTION [，OPTION ...]

这里OPTION是以下关键字来禁用的行动之一*调试*菜单（通常完全禁用）：

All

不要在此菜单中显示任何菜单项。

Statistics

不显示此菜单中的*统计...*菜单项。

CommandLine

不要在此菜单中显示*命令行...*菜单项。

Logging

不要在此菜单中显示*日志记录...*菜单项。

LogDialog

不要在此菜单中显示“ *显示日志”*菜单项。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeDebuggerMenuActions

使用以下命令禁用“ *视图”* 菜单的某些操作：

VBoxManage setextradata“VM name”GUI / RestrictedRuntimeHelpMenuActions OPTION [，OPTION ...]

这里OPTION是以下关键字来禁用的行动之一*帮助*菜单（通常完全禁用）：

All

不要在此菜单中显示任何菜单项。

Contents

不要在此菜单中显示*内容...*菜单项。

WebSite

不要在此菜单中显示*VirtualBox网站...*菜单项。

ResetWarnings

不要在此菜单中显示“ *重置所有警告”*菜单项。

NetworkAccessManager

不要在此菜单中显示*Network Operations Manager*菜单项。

About

不要在此菜单中显示“ *关于”*菜单项（仅在非Mac OS X主机上）。

Contents

不要在此菜单中显示*内容...*菜单项。

Contents

不要在此菜单中显示*内容...*菜单项。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM名称”GUI / RestrictedRuntimeHelpMenuActions

**9.20.5。配置VM窗口状态栏条目**

您可以禁用（即黑名单）某些状态栏项目：

VBoxManage setextradata“VM name”GUI / RestrictedStatusBarIndicators OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

HardDisks

不要在虚拟机窗口状态栏中显示硬盘图标。默认情况下，如果VM配置包含一个或多个硬盘，则仅显示硬盘图标。

OpticalDisks

不要在虚拟机窗口状态栏中显示CD图标。默认情况下，仅当VM配置包含一个或多个CD驱动器时才会显示CD图标。

FloppyDisks

不要在虚拟机窗口状态栏中显示软盘图标。默认情况下，仅当VM配置包含一个或多个软盘驱动器时才会显示软盘图标。

Network

不要在虚拟机窗口状态栏中显示网络图标。默认情况下，如果VM配置包含一个或多个活动网络适配器，则仅显示网络图标。

USB

不要在状态栏中显示USB图标。

SharedFolders

不要在状态栏中显示共享文件夹图标。

VideoCapture

不要在状态栏中显示视频捕获图标。

Features

不要在状态栏中显示CPU功能图标。

Mouse

不要在状态栏中显示鼠标图标。

Keyboard

不要在状态栏中显示键盘图标。

这是每个虚拟机的设置。允许上述任何组合。如果指定了所有选项，则VM窗口的状态栏中不会显示任何图标。要恢复默认行为，请使用

VBoxManage setextradata“VM name”GUI / RestrictedStatusBarIndicators

**9.20.6。配置虚拟机视窗模式**

您可以禁用（即黑名单）某些VM视觉模式：

VBoxManage setextradata“VM name”GUI / RestrictedVisualStates OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

Fullscreen

不允许将虚拟机切换到全屏模式。

Seamless

不允许将VM切换到无缝模式。

Scale

不允许将VM切换到刻度模式。

这是每个虚拟机的设置。允许上述任何组合。要恢复默认行为，请使用

VBoxManage setextradata“VM name”GUI / RestrictedVisualStates

**9.20.7。主机密钥定制**

要禁用所有主机组合键，请打开首选项并将主机键更改为*无*。在Kiosk模式下使用VirtualBox时，这可能很有用。

要重新定义或禁用某些主机键操作，请使用以下命令：

VBoxManage setextradata全局GUI / Input / MachineShortcuts“FullscreenMode = F，....”

以下列表显示可能的主机密钥操作及其默认主机密钥快捷方式。将操作设置为*None*将禁用该主机键操作。

**表9.1。主机密钥定制**

| **行动** | **默认键** | **行动** |
| --- | --- | --- |
| TakeSnapshot | Ť | 拍一张快照 |
| TakeScreenshot | Ë | 拍摄截图 |
| MouseIntegration | 一世 | 切换鼠标集成 |
| TypeCAD | 德尔 | 注入Ctrl + Alt + Del |
| TypeCABS | 退格 | 注入Ctrl + Alt + Backspace |
| Pause | P | 暂停虚拟机 |
| Reset | [R | （硬）重设客人 |
| SaveState |  | 保存VM状态并终止虚拟机 |
| Shutdown | H | 按（虚拟）ACPI电源按钮 |
| PowerOff |  | 关闭虚拟机电源（不保存状态！） |
| Close | Q | 显示VM关闭对话框 |
| FullscreenMode | F | 将VM切换到全屏 |
| SeamlessMode | 大号 | 将VM切换到无缝模式 |
| ScaleMode | C | 将VM切换到刻度模式 |
| GuestAutoResize | G | 自动调整访客窗口的大小 |
| WindowAdjust | 一个 | 立即调整访客窗口的大小 |
| PopupMenu | 家 | 在全屏/ seaml中显示弹出菜单。模式 |
| SettingsDialog | 小号 | 打开VM设置对话框 |
| InformationDialog | ñ | 显示虚拟机信息窗口 |
| NetworkAdaptersDialog |  | 显示VM网络适配器对话框 |
| SharedFoldersDialog |  | 显示VM共享文件夹对话框 |
| InstallGuestAdditions | ð | 装载包含客户添加的ISO |

要禁用全屏模式以及无缝模式，请使用以下命令：

VBoxManage setextradata全局GUI / Input / MachineShortcuts“FullscreenMode = None，SeamlessMode = None”

**9.20.8。终止虚拟机时的操作**

终止虚拟机时，您可以禁止（即黑名单）某些操作。要禁止具体操作，请键入：

VBoxManage setextradata“VM name”GUI / RestrictedCloseActions OPTION [，OPTION ...]

OPTION以下关键字之一在哪里？

SaveState

终止虚拟机时，不要让用户保存虚拟机状态。

Shutdown

不允许用户通过向客户端发送ACPI关机事件来关闭虚拟机。

PowerOff

不要让用户关闭虚拟机。

PowerOffRestoringSnapshot

关闭虚拟机时，不要让用户返回上一张快照。

Detach

如果VM以独立模式启动，则不允许用户从VM进程中脱离。

这是每个虚拟机的设置。允许上述任何组合。如果指定了所有选项，则VM根本无法关闭。

**9.20.9。终止虚拟机时的默认操作**

您可以定义终止虚拟机的具体操作。与上一节中描述的设置相反，当用户终止虚拟机时，此设置只允许一个操作。不显示退出菜单。

VBoxManage setextradata“VM名称”GUI / DefaultCloseAction ACTION

ACTION以下关键字之一在哪里？

SaveState

在终止VM进程之前保存VM状态。

Shutdown

通过向客户端发送ACPI关闭电源事件来关闭虚拟机。

PowerOff

虚拟机已关闭电源。

PowerOffRestoringSnapshot

VM关闭电源，保存的状态返回到最后一个快照。

Detach

终止前端，但使VM进程正在运行。

这是每个虚拟机的设置。允许上述任何组合。如果指定了所有选项，则VM根本无法关闭。

**10年9月20日。处理上师冥想的行动**

如果有一个问题不能通过其他方法来解决，而不是终止进程，那么虚拟机就会进入上帝的冥想。默认是显示一个消息窗口，指示用户打开错误报告。

可以配置此行为：

VBoxManage setextradata“VM名称”GUI / GuruMeditationHandler MODE

MODE以下关键字之一在哪里？

Default

显示一个消息窗口。用户确认后，VM终止。

PowerOff

VM立即关闭，不显示任何消息窗口。VM日志文件将显示有关发生的情况的信息。

Ignore

VM处于卡住模式。执行停止，但没有显示消息窗口。VM必须手动关闭。

这是每个虚拟机的设置。

**11年9月20日。配置自动鼠标捕获**

默认情况下，如果用户单击访客窗口并且客人期望此时的相对鼠标坐标，则会捕获鼠标。如果指点设备被配置为PS / 2鼠标，并且guest虚拟机（尚未）启动VirtualBox Guest Additions（例如，guest虚拟机正在启动或根本没有安装Guest Additions），或者如果将指点设备配置为USB平板电脑，但客人没有加载USB驱动程序。一旦客户添加活动或启动USB访客驱动程序，鼠标捕获将自动释放。

有时不需要默认行为。因此可以配置：

VBoxManage setextradata“VM名称”GUI / MouseCapturePolicy MODE

MODE以下关键字之一在哪里？

Default

默认行为如上所述。

HostComboOnly

仅当主键被切换时，鼠标才被捕获。

Disabled

鼠标从不被捕获，也不能通过切换主机键

这是每个虚拟机的设置。

**12年9月20日。配置自动鼠标捕获**

默认情况下，如果用户单击访客窗口并且客人期望此时的相对鼠标坐标，则会捕获鼠标。如果指点设备被配置为PS / 2鼠标，并且guest虚拟机（尚未）启动VirtualBox Guest Additions（例如，guest虚拟机正在启动或根本没有安装Guest Additions），或者如果将指点设备配置为USB平板电脑，但客人没有加载USB驱动程序。一旦客户添加活动或启动USB访客驱动程序，鼠标捕获将自动释放。

有时不需要默认行为。因此可以配置：

VBoxManage setextradata“VM名称”GUI / MouseCapturePolicy MODE

MODE以下关键字之一在哪里？

Default

默认行为如上所述。

HostComboOnly

仅当主键被切换时，鼠标才被捕获。

Disabled

鼠标从不被捕获，也不能通过切换主机键

这是每个虚拟机的设置。

**13年9月20日。请求传统的全屏模式**

从版本4.3.16起，VirtualBox使用特殊的窗口管理器功能，在多显示器主机系统上将多屏幕机器切换为全屏。然而，并不是所有的窗口管理器正确地提供这些设施，所以VirtualBox可以被告知要使用旧的方法切换到全屏模式，而不是使用命令：

VBoxManage setextradata全局GUI /全屏/ LegacyMode为true

您可以使用以下命令返回到新方法：

VBoxManage setextradata全局GUI /全屏/ LegacyMode

这是一个全球性的设置。

**9.21。自动启动VirtualBox Web服务**

VirtualBox Web Service（vboxwebsrv）用于远程控制VirtualBox。它在VirtualBox软件开发工具包（SDK）中有详细记录; 请参见[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。随着使用这种接口的客户端不断增长，我们为我们支持的各种操作系统添加了启动脚本。以下部分介绍如何使用它们。由于标准安装，VirtualBox Web服务不会自动启动。

**9.21.1。Linux：启动webserviceinit**

在Linux上，通过向文件添加适当的参数，可以在主机启动期间自动启动Web服务 /etc/default/virtualbox。有一个强制参数，VBOXWEB\_USER必须设置为用户以后才能启动虚拟机。下表中的所有参数开始VBOXWEB\_ （VBOXWEB\_HOST， VBOXWEB\_PORT等）：

**表9.2。Web服务配置参数**

| **参数** | **描述** | **默认** |
| --- | --- | --- |
| USER | Web服务运行的用户 |  |
| HOST | 主机将Web服务绑定到 | 本地主机 |
| PORT | 将Web服务绑定到的端口 | 18083 |
| SSL\_KEYFILE | 服务器密钥和证书文件，PEM格式 |  |
| SSL\_PASSWORDFILE | 密码到服务器密钥的文件名 |  |
| SSL\_CACERT | CA证书文件，PEM格式 |  |
| SSL\_CAPATH | CA证书路径 |  |
| SSL\_DHFILE | DH文件名或DH密钥长度，以位为单位 |  |
| SSL\_RANDFILE | 包含随机数生成器的种子的文件 |  |
| TIMEOUT | 会话超时（以秒为单位）0禁用超时 | 300 |
| CHECK\_INTERVAL | 超时检查频率（秒） | 五 |
| THREADS | 并行运行的最大工作线程数 | 100 |
| KEEPALIVE | 套接字关闭前的最大请求数 | 100 |
| ROTATE | 日志文件数; 0禁用日志轮换 | 10 |
| LOGSIZE | 日志文件的最大大小（以字节为单位）来触发旋转 | 1MB |
| LOGINTERVAL | 触发日志旋转的最长时间间隔（以秒为单位） | 1天 |

设置参数SSL\_KEYFILE 可启用SSL / TLS支持。强烈建议使用加密，否则所有内容（包括密码）以明文形式传输。

**9.21.2。Solaris：通过SMF启动Web服务**

在Solaris主机上，VirtualBox Web服务守护程序集成到SMF框架中。您可以更改参数，但如果下面的默认值与您的需要相匹配，则不需要：

svccfg -s svc：/ application / virtualbox / webservice：default setprop config / host = localhost svccfg -s svc：/ application / virtualbox / webservice：default setprop config / port = 18083 svccfg -s svc：/ application / virtualbox / webservice：default setprop config / user = root

上一节中显示参数名称和默认值的表也适用于Solaris。参数名称必须更改为小写，并且config/ 必须添加前缀，例如config/user或 config/ssl\_keyfile。如果您进行了任何更改，请不要忘记运行以下命令以使更改立即生效：

svcadm refresh svc：/ application / virtualbox / webservice：default

如果您忘记了上述命令，则在启用服务时将使用以前的设置。检查当前属性设置：

svcprop -p config svc：/ application / virtualbox / webservice：default

当一切配置正确后，您可以使用以下命令启动VirtualBox Web服务：

svcadm enable svc：/ application / virtualbox / webservice：default

有关SMF的更多信息，请参考Solaris文档。

**9.21.3。Mac OS X：通过launchd启动webservice**

在Mac OS X上，launchd用于启动VirtualBox webservice。可以找到一个示例配置文件 $HOME/Library/LaunchAgents/org.virtualbox.vboxwebsrv.plist。它可以通过改变启用 Disabled来自键true来 false。要手动启动服务，请使用以下命令：

launchctl load〜/ Library / LaunchAgents / org.virtualbox.vboxwebsrv.plist

有关如何配置launchd服务的其他信息，请参阅[https://developer.apple.com/library/mac/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html](https://developer.apple.com/library/mac/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html" \t "_top)。

**9.22。VirtualBox看门狗**

从VirtualBox 4.2开始，内存VBoxBalloonCtrl扩展服务以前称为已更名为VBoxWatchdog，它现在包含了许多要在服务器环境中运行的主机服务。

这些服务是：

* 内存气球控制，可自动处理虚拟机配置的内存气球（请参见[第4.9.1节“内存气球”](https://www.virtualbox.org/manual/ch04.html#guestadd-balloon) ，介绍内存气球）。这对于在运行时期间VM可能需要更多或更少内存的服务器环境尤其有用。

该服务定期检查虚拟机的当前内存气球及其空闲的客户RAM，并通过相应地膨胀或放气来自动调整当前内存气球。此处理仅适用于安装了最近访客添加的运行虚拟机。

* 主机隔离检测提供了一种检测主机是否无法到达特定VirtualBox服务器实例的方法，并采取适当的措施，例如关闭，保存当前状态，甚至关闭某些虚拟机。

所有配置值可以通过命令行或全局extradata指定，而命令行值在设置时始终具有较高的优先级。还可以在每个虚拟机的基础上指定一些配置值。因此，整体查找顺序是：命令行，每个虚拟机基础的数据库（如果可用），全局数据库。

**9.22.1。记忆气球控制**

内存膨胀控制基于VM空闲内存和所需的最大气球大小来充气和缩小VM的内存气球。

要设置内存气球控制，需要设置VM可以达到的最大气泡尺寸。这可以通过命令行指定

--balloon-max <MB以MB为单位

，以每个虚拟机为基础的extradata值

VBoxManage setextradata <VM-Name> VBoxInternal2 / Watchdog / BalloonCtrl / BalloonSizeMax <Size in MB>

或使用全局extradata值

VBoxManage setextradata global VBoxInternal2 / Watchdog / BalloonCtrl / BalloonSizeMax <Size in MB>

**注意**

如果上述至少一个参数没有指定最大气球尺寸，则根本不会执行气球。

以MB为单位设置气球增量可以通过命令行来完成

--balloon-inc <大小（MB）>

或使用全局extradata值

VBoxManage setextradata全局VBoxInternal2 /看门狗/ BalloonCtrl / BalloonIncrementMB <大小在MB>

如果未指定，默认气球增量为256 MB。

与气球减少相同：通过命令行与

--balloon-dec <大小（MB）>

或使用全局extradata值

VBoxManage setextradata全局VBoxInternal2 /看门狗/ BalloonCtrl / BalloonDecrementMB <大小在MB>

如果未指定，默认气球减量为128 MB。

要定义MB的下限，气球可以是命令行

--balloon-lower-limit（以MB为单位）

可以使用或使用全局extradata值

VBoxManage setextradata全局VBoxInternal2 /看门狗/ BalloonCtrl / BalloonLowerLimitMB <大小（MB）>

是可用的 如果未指定，默认下限为128。

**9.22.2。主机隔离检测**

要检测主机是否被隔离，也就是主机不能再到达VirtualBox服务器实例，则主机需要在一段时间内将交替值设置为全局的数据值。如果在该时间段内未设置此值，则会发生超时，并对所处理的VM执行所谓的主机隔离响应。可以通过定义VM组并将VM分配给这些组来控制哪些VM被处理。默认情况下没有设置任何组，这意味着在30秒内没有收到主机响应时，将处理服务器上的所有虚拟机。

通过命令行设置通过主机隔离检测处理的组：

--apimon基团= <串[，stringN]>

或使用全局extradata值

VBoxManage setextradata global VBoxInternal2 / Watchdog / APIMonitor / Groups <string [，stringN]>

通过命令行设置主机隔离超时：

--apimon-ISLN超时= <毫秒>

或使用全局extradata值

VBoxManage setextradata全局VBoxInternal2 /看门狗/ APIMonitor / IsolationTimeoutMS <ms>

通过命令行设置实际的主机隔离响应：

--apimon-ISLN响应= <CMD>

或使用全局extradata值

VBoxManage setextradata全局VBoxInternal2 / Watchdog / APIMonitor / IsolationResponse <cmd>

以下响应命令可用：

* none，什么也没做。
* pause，它会暂停执行VM。
* poweroff，它通过按下虚拟电源按钮关闭虚拟机。VM将无法保存任何数据或否决关闭过程。
* save，它可以节省当前的机器状态，并在之后关闭VM。如果保存机器状态失败，VM将被暂停。
* shutdown，通过向ACPI VM的操作系统发送关闭事件，以温和的方式关闭虚拟机。操作系统然后有机会进行干净的关机。

**9.22.3。更多信息**

有关更多高级选项和参数（如详细日志记录），请检查内置的命令行帮助 --help。

**9.22.4。Linux：通过启动看门狗服务init**

在Linux上，通过向文件添加适当的参数，可以在主机启动期间自动启动看门狗服务 /etc/default/virtualbox。有一个强制参数，VBOXWATCHDOG\_USER必须设置为用户以后才能启动虚拟机。为了向后兼容，你也可以指定VBOXBALLOONCTRL\_USER的参数表中下面的所有开始VBOXWATCHDOG\_ （VBOXWATCHDOG\_BALLOON\_INTERVAL， VBOXWATCHDOG\_LOGSIZE等等，以及先前存在的参数，VBOXBALLOONCTRL\_INTERVAL仍可使用等参数）：

**表9.3。VirtualBox看门狗配置参数**

| **参数** | **描述** | **默认** |
| --- | --- | --- |
| USER | 看门狗服务运行的用户 |  |
| ROTATE | 日志文件数; 0禁用日志轮换 | 10 |
| LOGSIZE | 日志文件的最大大小（以字节为单位）来触发旋转 | 1MB |
| LOGINTERVAL | 触发日志旋转的最长时间间隔（以秒为单位） | 1天 |
| BALLOON\_INTERVAL | 检查气球尺寸的间隔（毫秒） | 30000 |
| BALLOON\_INCREMENT | 气球尺寸增量（MByte） | 256 |
| BALLOON\_DECREMENT | 气球尺寸减量（MByte） | 128 |
| BALLOON\_LOWERLIMIT | 气球尺寸下限（MByte） | 64 |
| BALLOON\_SAFETYMARGIN | 减少气球尺寸所需的可用内存（MByte） | 1024 |

**9.22.5。Solaris：通过SMF启动看门狗服务**

在Solaris主机上，VirtualBox看门狗服务守护程序集成到SMF框架中。您可以更改参数，但如果默认值与您的需求相匹配，则不需要：

svccfg -s svc：/ application / virtualbox / balloonctrl：default setprop config / balloon\_interval = 10000 svccfg -s svc：/ application / virtualbox / balloonctrl：default setprop config / balloon\_safetymargin = 134217728

上一节中显示参数名称和默认值的表也适用于Solaris。参数名称必须更改为小写，并且config/ 必须添加前缀，例如config/user或 config/balloon\_safetymargin。如果您进行了任何更改，请不要忘记运行以下命令以使更改立即生效：

svcadm refresh svc：/ application / virtualbox / balloonctrl：default

如果您忘记了上述命令，则在启用服务时将使用以前的设置。检查当前属性设置：

svcprop -p config svc：/ application / virtualbox / balloonctrl：default

当一切配置正确后，您可以使用以下命令启动VirtualBox看门狗服务：

svcadm enable svc：/ application / virtualbox / balloonctrl：default

有关SMF的更多信息，请参考Solaris文档。

**9.23。其他扩展包**

从VirtualBox 4.2.0开始，还有另一个扩展包， VNC它是开放源代码，取代了之前VNC远程访问协议的集成。这是实验代码，最初只能在VirtualBox源代码包中使用。这是用户贡献的大部分代码，Oracle不支持任何方式。

键盘处理受到严格限制，只有美式键盘布局才能正常工作。其他键盘布局至少会产生一些产生错误结果的键（通常非常令人惊讶的效果），而对于与美国键盘布局有显着差异的布局，它最有可能无法使用。

可以同时安装Oracle VM VirtualBox Extension Pack和VNC，但只能在一个VRDE模块中处于活动状态。以下命令切换到VNC中的VNC VRDE模块：

VBoxManage setproperty vrdeextpack VNC

配置远程访问的作品非常相似VRDP（参见 [7.1节，“远程显示（VRDP支持）”](https://www.virtualbox.org/manual/ch07.html#vrde)），有一些限制：VNC不支持指定的几个端口号，并在认证方式有所不同。VNC只能处理密码认证，没有选择使用密码哈希。这不会在VM配置中使用明文密码，而是可以使用以下命令进行设置：

VBoxManage modifyvm“VM名称”--vrdeproperty VNCPassword = secret

用户负责保密此密码，并且无论何种目的，将虚拟机配置传递给其他人员都应将其删除。一些VNC服务器声称在配置中具有“加密”密码。这不是真正的加密，它只是隐藏了密码，这与密码一样安全。

以下命令切换回VRDP（如果已安装）：

VBoxManage setproperty vrdeextpack“Oracle VM VirtualBox扩展包”

**9.24。在系统引导期间启动虚拟机**

从VirtualBox 4.2.0开始，可以在Linux，Solaris和Mac OS X的系统引导期间为所有用户自动启动虚拟机。

**9.24.1。Linux：启动自动启动服务init**

在Linux上，通过设置两个变量激活自动启动服务 /etc/default/virtualbox。第一个是VBOXAUTOSTART\_DB包含自动启动数据库目录的绝对路径。该目录应该具有对每个可以自动启动虚拟机的用户的写入权限。此外，目录应该设置粘性位。第二个变量是VBOXAUTOSTART\_CONFIG 哪些指向在引导期间使用的自动启动配置文件的服务，以确定是否允许单个用户自动启动VM并配置启动延迟。可以将配置文件放入/etc/vbox 并包含多个选项。default\_policy 一个控制自动启动服务是否允许或拒绝为不在异常列表中的用户启动VM。异常列表以exception\_list 逗号分隔的列表开头，并包含用户名。此外，可以为每个用户配置单独的启动延迟，以避免主机过载。示例配置如下：

＃默认策略是拒绝启动虚拟机，另一个选项是“允许”。 default\_policy = deny ＃Bob被允许启动虚拟机，但是启动它们 ＃将被延迟10秒钟 bob = { allow = true startup\_delay = 10 } ＃Alice不允许启动虚拟机，有助于排除某些用户 ＃如果默认策略设置为allow。 alice = { allow = false }

想要为单个机器启用自动启动的每个用户都必须设置自动启动数据库目录的路径

VBoxManage setproperty autostartdbpath <Autostart directory>

**9.24.2。Solaris：通过SMF启动自动启动服务**

在Solaris主机上，VirtualBox自动启动守护程序集成到SMF框架中。要启用它，您必须将服务指向与Linux相同格式的现有配置文件（请参见[第9.24.1节“Linux：启动自动启动服务init”](https://www.virtualbox.org/manual/ch09.html#autostart-linux)）：

svccfg -s svc：/ application / virtualbox / autostart：default setprop config / config = / etc / vbox / autostart.cfg

当一切配置正确后，您可以使用以下命令启动VirtualBox自动启动服务：

svcadm enable svc：/ application / virtualbox / autostart：default

有关SMF的更多信息，请参考Solaris文档。

**9.24.3。Mac OS X：通过launchd启动自动启动服务**

在Mac OS X上，launchd用于启动VirtualBox自动启动服务。可以找到一个示例配置文件 /Applications/VirtualBox.app/Contents/MacOS/org.virtualbox.vboxautostart.plist。要启用该服务将文件复制到/Library/LaunchDaemons和改变 Disabled从键 true到 false。此外，将第二个参数替换为与Linux相同格式的现有配置文件（参见[第9.24.1节“Linux：启动自动启动服务init”](https://www.virtualbox.org/manual/ch09.html#autostart-linux)）。要手动启动服务，请使用以下命令：

launchctl load /Library/LaunchDaemons/org.virtualbox.vboxautostart.plist

有关如何配置launchd服务的其他信息，请参阅[http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPSystemStartup/BPSystemStartup.html](http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPSystemStartup/BPSystemStartup.html" \t "_top)。

**9.25。VirtualBox专家存储管理**

如果VirtualBox的快照模型不够，可以启用特殊模式，这样可以在VM暂停时重新配置存储附件。用户必须确保磁盘数据与客户保持一致，因为与热插拔不同，客户不知道分离的或新连接的媒体。

可以在每个执行VM的情况下启用专家存储管理模式：

VBoxManage setextradata“VM名称”“VBoxInternal2 / SilentReconfigureWhilePaused”1

可以在虚拟机暂停之后使用以下方式重新配置存储附件：

VBoxManage storageattach ...

**9.26。处理主机电源管理事件**

一些主机电源管理事件由VirtualBox处理。实际行为取决于平台：

主机暂停

当主机即将暂停时，即主机将状态保存到某些非易失性存储并关闭电源时，会生成此事件。

此事件目前仅在Windows主机和Mac OS X主机上处理。生成此事件时，VirtualBox将暂停所有正在运行的虚拟机。

主持人简历

当主机从挂起状态唤醒时，会生成此事件。

此事件目前仅在Windows主机和Mac OS X主机上处理。当生成此事件时，VirtualBox将恢复所有在之前暂停的虚拟机。

电量不足

电池电量达到临界水平（通常小于5％充电）。

此事件目前仅在Windows主机和Mac OS X主机上处理。生成此事件时，VirtualBox将保存状态并终止所有虚拟机，以准备潜在的主机掉电。

可以配置该行为。通过执行以下命令，不保存虚拟机：

VBoxManage setextradata全局“VBoxInternal2 / SavestateOnBatteryLow”0

这是一个全局设置以及每个虚拟机的设置。每个VM值的优先级高于全局值。以下命令将保存所有VM的状态，但不会保存VM“foo”的状态：

VBoxManage setextradata全局“VBoxInternal2 / SavestateOnBatteryLow”1 VBoxManage setextradata“foo”“VBoxInternal2 / SavestateOnBatteryLow”0

实际上不需要第一行，因为默认情况下执行savestate操作。

**9.27。实验支持通过SSE4.1 / SSE4.2说明**

为了向客户提供SSE 4.1 / SSE 4.2支持，主机CPU必须实现这些指令集。从VirtualBox 4.3.8开始，可以使用以下命令为某些guest虚拟机启用这些指令：

VBoxManage setextradata“VM名称”VBoxInternal / CPUM / SSE4.1 1 VBoxManage setextradata“VM名称”VBoxInternal / CPUM / SSE4.2 1

这些是每个VM设置，默认情况下它们被关闭。

**9.28。支持键盘指示灯同步**

当选择机器窗口时，此功能使主机键盘灯与虚拟机的虚拟键盘相匹配。它目前在Mac OS X和Windows主机上实现，并且从版本4.2.24和4.3.8可用。可以使用以下命令启用该功能：

VBoxManage setextradata“VM名称”GUI / HidLedsSync“1”

为了禁用它，使用相同的命令，但将“1”更改为“0”，或使用VBoxManage命令删除额外的数据。这是一个每个虚拟机设置，默认情况下它被禁用。

**9.29。捕获所选设备的USB流量**

从VirtualBox 5.0开始，可以捕获单个USB设备或根集线器级别的USB流量，捕获连接到根集线器的所有USB设备的流量。VirtualBox以与Wireshark兼容的格式存储流量。要捕获特定USB设备的流量，必须使用以下命令使用VBoxManage连接到VM：

VBoxManage controlvm“VM名称”usbattach“device uuid | address”--capturefile“filename”

为了在虚拟机未运行时使用以下命令在根集线器上启用捕获：

VBoxManage setextradata“VM name”VBoxInternal / Devices / usb-ehci / 0 / LUN＃0 / Config / CaptureFilename“filename”

上述命令可以在连接到EHCI控制器的根集线器上进行捕获。以使其能够为OHCI或XHCI控制器更换usb-ehci 用usb-ohci或usb-xhci分别。

**9.30。配置心跳服务**

VirtualBox发送一个简单的心跳服务。一旦Guest Additives处于活动状态，客人会向主机发送频繁的心跳呼叫。如果客人停止发送心跳，而不正常终止服务，则VM进程会将此事件记录在VBox.log文件中。将来可能会配置专用操作，但现在在日志文件中只有一个警告。

有两个参数要配置。该*心跳间隔* 定义了两个心跳ping之间的时间。默认值为2秒，即VirtualBox Guest Additions的心跳线服务将每隔两秒发送一次心跳线。纳秒值可以这样配置：

VBoxManage controlvm“VM名称”VBoxInternal / Devices / VMMDev / 0 / Config / HeartbeatInterval 2000000000

该*心跳超时*定义了主机等待从最后一次心跳平开始它的客户定义为之前没有反应的时间。默认值为心跳间隔的2倍（4秒），可以配置如下（纳秒）：

VBoxManage controlvm“VM名称”VBoxInternal / Devices / VMMDev / 0 / Config / HeartbeatTimeout 4000000000

如果心跳超时到期，将会有一个日志消息，如 *VMMDev：HeartBeatCheckTimer：Guest似乎没有反应。5秒钟前收到最后的心跳。* 如果另一个心跳ping在此警告后到达，将会有一个日志消息，如*VMMDev：GuestHeartBeat：Guest还活着。*

**9.31。加密磁盘映像**

从VirtualBox 5.0开始，可以透明地为访客加密存储在硬盘映像中的数据。它不依赖于要使用的特定图像格式。加密数据的图像在VirtualBox和其他虚拟化软件之间不可移植。

VirtualBox在XTS模式下使用AES算法，并支持128或256位数据加密密钥（DEK）。DEK在媒体属性中加密存储，并在VM启动期间通过输入在加密图像时选择的密码进行解密。

由于DEK作为VM配置文件的一部分存储，所以保持安全是很重要的。丢失DEK意味着存储在磁盘映像中的数据不可恢复地丢失。对与VM相关的所有数据进行完整和最新的备份是用户的责任。

**9.31.1。限制**

使用此功能时，用户需要注意的一些限制：

* 此功能是需要安装的Oracle VM VirtualBox Extension Pack的一部分。否则磁盘加密不可用。
* 由于加密只对存储的用户数据有效，因此目前无法检查磁盘映像的元数据完整性。攻击者可能会通过删除或更改图像中的数据块或更改元数据项（如磁盘大小）来破坏数据。
* 导出包含加密磁盘映像的设备是不可能的，因为OVF规范不支持此功能。所有图像因此在导出过程中被解密。
* 当VM运行时，将DEK保存在内存中，以便能够解密读取和加密由访客写入的数据的数据。虽然这应该是显而易见的，但用户需要注意这一点，因为攻击者可能能够在受感染的主机上提取密钥并解密数据。
* 加密或解密图像时，密码将通过VirtualBox API以明文形式传递。需要牢记这一点，特别是在使用第三方API客户端时，使用可以通过网络传输密码的Web服务。在这种情况下，使用HTTPS是必需的。
* 只有没有快照或线性的快照链时，才能使用差分图像加密图像。这个限制可以在将来的VirtualBox版本中解决。

**9.31.2。加密磁盘映像**

加密磁盘映像可以使用GUI或VBoxManage完成。虽然GUI更易于使用，但它可以在每个VM的基础上进行加密，并附加到特定VM的所有磁盘映像。使用VBoxManage可以加密单个图像（包括所有差分图像）。要使用VBoxManage加密未加密的介质，请使用：

VBoxManage encryptmedium“uuid | filename”--newpassword“file | - ”--cipher“cipher id”--newpasswordid“id”

要将加密密码点VBoxManage提供给存储密码的文件或指定-，让VBoxManage在命令行中要求您输入密码。

cipher参数指定用于加密的密码，可以是 AES-XTS128-PLAIN64或AES-XTS256-PLAIN64。指定的密码标识符可以由用户自由选择，并在虚拟机启动期间提供多个密码时用于正确识别。

如果用户在加密多个图像时使用相同的密码，并且使用相同的密码标识符，则用户只需要在启动VM时提供一次密码。

**9.31.3。启动加密映像的虚拟机**

当使用GUI启动虚拟机时，会打开一个对话框，用户需要输入所有连接到虚拟机的加密映像的密码。如果使用另一个前端像VBoxHeadless，VM将在客户端尝试访问加密磁盘后立即暂停。用户需要使用以下命令通过VBoxManage提供密码：

VBoxManage controlvm“uuid | vmname”addencpassword“id”“password”[--removeonsuspend“yes | no”]

该id参数必须与加密图像时提供的密码标识符相同。password是加密图像时使用的密码。用户可以选择指定 --removeonsuspend "yes|no"在VM挂起时是否从VM内存中删除密码。在VM恢复之前，用户需要再次提供密码。当VM由主机挂起事件暂停并且用户不希望密码保留在内存中时，这是非常有用的。

**9.31.4。解密加密图像**

在某些情况下，可能需要解密先前加密的图像。这可以在GUI中完成一个完整的VM或使用VBoxManage与以下命令：

VBoxManage encryptmedium“uuid | filename”--oldpassword“file | - ”

唯一必需的参数是加密图像的密码。选项与加密图像相同。

**9.32。半虚拟化调试**

在本节中，我们将介绍使用半虚拟化提供程序支持的接口访客操作系统的调试。

**注意**

半虚拟化调试显着改变客户机操作系统的行为，只能由专家用户进行调试和诊断。

这些调试选项被指定为用逗号分隔的键 - 值对的字符串。空字符串禁用半虚拟化调试。

**9.32.1。Hyper-V调试选项**

下面列出的所有选项都是可选的，因此在指定对应的键值对时将使用指定的默认值。

* 键： **enabled**

价值：0或1

默认： 0

指定1启用Hyper-V调试接口。如果未指定此键值对或值不是 1，则无论是否存在其他键值对，都会禁用Hyper-V调试接口。

* 键： **address**

值：IPv4地址

默认值：127.0.0.1

指定远程调试器连接的IPv4地址。

* 键： **port**

值：UDP端口号

默认值：50000

指定远程调试器连接的UDP端口号。

* 键： **vendor**

价值：通过CPUID向访客报告Hyper-V供应商签名

默认值：启用调试时：Microsoft Hv否则：VBoxVBoxVBox

通过CPUID指定暴露给客人的Hyper-V供应商签名。对于调试Microsoft Windows客户机，需要虚拟机管理程序报告Microsoft供应商。

* 键： **hypercallinterface**

价值：0或1

默认： 0

指定在客人请求时，是否应建议使用超级呼叫来启动主机和客户端之间的调试数据传输，而不是MSR。

* 键： **vsinterface**

价值：0或1

默认值：启用调试时1，否则0

指定是否向guest虚拟机显示“VS＃1”（虚拟化服务）接口。此界面是调试Microsoft Windows 10 32位客户端所必需的，但对于其他Windows版本是可选的。

**9.32.1.1。使用Hyper-V半虚拟化提供程序设置Windows guest虚拟机进行调试**

Windows支持通过串行电缆，USB，IEEE 1394 Firewire和以太网（仅Windows 8及更高版本）进行调试。USB和IEEE 1394不适用于虚拟机，以太网需要Windows 8或更高版本。虽然串行连接是普遍可用的，但速度很慢。

使用Windows Vista及更高版本支持的Hyper-V调试传输进行调试提供了显着的优势。由于直接的主机到客户转移，它提供出色的性能，易于设置，并且需要从管理程序的最少支持。它可以与与VM相同的主机上运行的调试器或与通过网络连接的不同机器上的调试器和VM一起使用。

**9.32.1.1.1。先决条件**

* 配置为运行Windows Vista或更新的Windows guest虚拟机的Hyper-V半虚拟化的VM。您可以从以下VBoxManage命令的输出中检查虚拟机的有效半虚拟化提供程序：

VBoxManage showvminfo“VM名称”

* 要在虚拟机中调试Windows版本所需的Microsoft WinDbg调试器的最新版本。
* 虽然Windows 8和更新的Windows客户端配有Hyper-V调试支持，但Windows 7和Vista不支持。要在Windows 7或Vista的guest虚拟机中使用Hyper-V调试，请kdvm.dll从Windows 8.0安装中复制文件 [[45](https://www.virtualbox.org/manual/ch09.html" \l "ftn.idm9409) ]。此文件通常位于C:\Windows\System32。将其复制到Windows 7 / Vista访客中的相同位置。确保复制与客户机操作系统匹配的32位或64位版本的DLL。

**9.32.1.1.2。虚拟机和访客配置**

1. 关闭虚拟机。
2. 通过执行以下VBoxManage命令启用调试选项：

VBoxManage modifyvm“VM名称”--paravirtdebug“enabled = 1”

上述命令假设您的调试器将在UDP端口50000上连接到主机。但是，如果需要在远程计算机上运行调试器，则可以在此处指定远程地址和端口，例如：

VBoxManage modifyvm“VM名称”--paravirtdebug“enabled = 1，address = 192.168.32.1，port = 55000”

有关完整的选项[，](https://www.virtualbox.org/manual/ch09.html#gimdebughyperv)请参见[第9.32.1节“Hyper-V调试选项”](https://www.virtualbox.org/manual/ch09.html#gimdebughyperv)。

1. 启动虚拟机
2. 在访客中，启动提升的命令提示符并执行以下命令：
   * 对于Windows 8或更新的Windows guest：

bcdedit / dbgsettings net hostip：5.5.5.5 port：50000 key：1.2.3.4

* + 对于Windows 7或Vista客人：

bcdedit / set loadoptions host\_ip = 5.5.5.5，host\_port = 50000，encryption\_key = 1.2.3.4

bcdedit / set dbgtransport kdvm.dll

bcdedit使用Hyper-V调试传输时，将忽略该命令中的IP地址和端口。可以输入任何有效的IP和大于49151且低于65536的端口号。

bcdedit命令中的加密密钥是相关的，必须是有效的。上述示例中使用的密钥“1.2.3.4”是有效的，如果不是安全性，可以使用它。如果您不指定任何加密密钥，bcdedit将为您生成一个密钥， 您将需要复制此密钥以便稍后在远程端输入Microsoft WinDbg。此加密密钥用于加密在Windows和调试器之间交换的调试数据。

* + 执行一个或多个以下命令以启用对Windows客户机的相应阶段或组件的调试：

bcdedit / set调试

bcdedit / set bootdebug打开

bcdedit / set {bootmgr} bootdebug打开

请注意，这些bootdebug选项仅在使用Hyper-V调试传输时在Windows 8或更高版本上有效。有关bcdedit 选项的详细说明，请参阅Microsoft Windows文档。

1. 在主机或远程主机上启动Microsoft WinDbg。

从“文件”菜单中选择“内核调试”。在“NET”选项卡下，指定您在paravirtdebug选项中使用的UDP端口号。如果没有指定，请保留为50000.确保UDP端口未被防火墙或其他安全软件阻止。

在“密钥”字段中，1.2.3.4从bcdeditWindows客户机中的命令输入或加密密钥。

现在按“确定”开始侦听连接。Microsoft WinDbg通常在此阶段显示“等待重新连接”消息。

或者，从命令行启动WinDbg直接启动调试会话：

windbg.exe -k net：port = 50000，key = 1.2.3.4

有关完整的命令行语法，请参阅WinDbg文档。

1. 重新启动Windows客户端，然后应该使用Microsoft WinDbg作为调试工具进行连接。

**9.33。PC扬声器通道**

作为实验功能（主要由于仅限于Linux主机和未知的Linux发行版），VirtualBox支持通过PC扬声器传送到主机。PC扬声器（有时称为系统扬声器）是一种产生可听见的反馈（如哔音）而不需要常规音频/声卡支持的方式。

VirtualBox中的PC扬声器直通功能仅处理蜂鸣声。VM使用的高级PC扬声器（如PCM音频）将无法正常工作，导致未定义的主机行为。

不幸的是，在Linux上产生嘟嘟声是一个非常复杂的话题。VirtualBox提供了一系列选项，以尽可能地在尽可能多的Linux发行版和系统配置上使此工作确定性和可靠地工作：

**表9.4。PC扬声器配置选项**

| **码** | **设备** | **笔记** |
| --- | --- | --- |
| 1 | /dev/input/ by-path/platform- pcspkr-event-spkr | 直接主机PC扬声器使用。 |
| 2 | /dev/tty | 使用VM进程的终端关联。VM需要在虚拟控制台上启动。 |
| 3 | /dev/tty0 要么 /dev/vc/0 | 只能由root 具有能力的用户或用户使用cap\_sys\_tty\_config |
| 9 | 用户指定的控制台或evdev设备路径 | 像1-3，只需要一个自定义的设备路径。 |
| 70 | /dev/tty | 标准蜂鸣声。失去频率和长度。见代码2。 |
| 79 | 用户指定的终端设备路径 | 像70，只需要一个自定义的设备路径。 |
| 100 | 以上所有 | 尝试所有上述代码。 |

要启用PC扬声器直通，请使用以下命令：

VBoxManage setextradata“VM name”“VBoxInternal / Devices / i8254 / 0 / Config / PassthroughSpeaker”N

替换N代表您想要使用的案例的代码。更改此设置将在下次启动VM时生效。在所有主机操作系统上启用PC扬声器通路是安全的。它只会对Linux产生影响。

VM日志文件，VBox.log将包含前缀的行，PIT: speaker: 显示PC扬声器直通设置活动。它提供了它选择的设备或为什么它失败的提示。

为VM启用PC扬声器通路通常是简单的部分。真正的困难是确保VirtualBox可以访问必要的设备，因为在典型的Linux安装中，大多数只能由用户访问root。您应该遵循首选的方式来持久更改此内容，例如参考您的发行版的文档。由于有无数的Linux发行版本，我们只能提供一般的提示，通常有一种方法可以让X11会话用户访问其他设备，或者您需要使用udev配置文件找到一个工作解决方案。如果一切都失败，您可以尝试使用在主机系统启动中足够运行的脚本设置权限。

有时，内核应用附加规则来限制访问（例如，VM进程必须具有与配置为用于嘟嘟声的设备相同的控制终端，对于诸如VirtualBox的GUI应用程序来说通常很难实现）。上表包含一些提示，但通常参考Linux文档。

如果即使设置了设备权限，并且VBox.log确认它使用evdev或控制台进行PC扬声器控制，即使您有麻烦，请检查您的系统是否有PC扬声器。一些系统没有一个。Linux会将PC扬声器输出重新路由到声卡，可能会导致其他并发症。如果将扬声器连接到声卡，请检查是否可以听到哔声。今天几乎所有的系统都有一个。最后，检查音频混合器控制是否有一个名为“哔声”的通道（可能会被隐藏在混音器设置中），并且没有静音。

**9.34。通过USB / IP访问通过网络公开的USB设备**

从5.1.0开始，VirtualBox支持通过使用USB over IP协议的网络传输的USB设备，而无需配置内核和usbip工具提供的客户端。此外，该功能与在任何受支持的主机上运行的VirtualBox配合使用，而不仅仅是Linux（与官方客户端相同）。

要启用对通过USB / IP设备的支持，必须使用以下命令添加导出设备的设备服务器：

VBoxManage usbdevsource添加“唯一名称” - 后端“USBIP” - 地址“设备服务器[：端口]”

可以通过GUI或VBoxManage访问设备服务器上导出的USB设备，如本地连接的任何USB设备。这可以多次访问不同的设备服务器。

要删除设备服务器，可以使用以下命令：

VBoxManage usbdevsource删除“唯一名称”

**9.34.1。在Linux系统上设置USB / IP支持**

本节简要介绍如何将基于Linux的系统设置为充当USB设备服务器。在服务器上的系统要求 usbip-core.ko和 usbip-host.ko内核驱动程序可用，并且安装了USB / IP工具包。必要工具的特定安装方法取决于使用哪种分布。例如，对于基于Debian的系统，应使用以下命令来安装所需的工具：

apt-get install usbip-utils

要检查是否已安装必要的工具，请使用以下命令：

$ usbip list -l

这应该产生类似于下面的例子所示的输出：

- busid 4-2（0bda：0301） Realtek半导体公司：多卡读卡器（0bda：0301） - 总线5-1（046d：c52b） 罗技公司：统一接收器（046d：c52b）

如果安装了所有内容，则需要root使用以下命令启动USB / IP服务器 ：

usbipd -D

请参阅安装发行版的文档，以确定系统引导时如何启动服务。

默认情况下，不导出服务器上的设备 - 必须手动为每台设备完成设备。要导出设备，请使用：

usbip bind -b“总线标识符”

例如，要从上方导出多卡阅读器，请使用：

usbip bind -b 4-2

**9.34.2。安全考虑**

服务器和客户端之间的通信是未加密的，并且没有授权访问导出的设备。攻击者可能会窃取敏感数据或获取对设备的控制权。为了减轻这种风险，设备应该通过本地网络进行暴露，只有受信任的客户端才能访问该网络。要通过公共网络远程访问设备，应使用VPN解决方案来提供所需的安全保护级别。

[[41](https://www.virtualbox.org/manual/ch09.html" \l "idm7553) ] VirtualBox 3.2引入了CPU热插拔支持。

[[42](https://www.virtualbox.org/manual/ch09.html" \l "idm7577) ]对于PCI直通实验支持用的VirtualBox 4.1引入。

[[43](https://www.virtualbox.org/manual/ch09.html" \l "idm7980) ] VirtualBox 4.1引入了支持Crossbow的桥接网络，需要Solaris 11 build 159或更高版本。

[[44](https://www.virtualbox.org/manual/ch09.html" \l "idm7983) ]当使用基于Crossbow的桥接网络时。

[[45](https://www.virtualbox.org/manual/ch09.html" \l "idm9409) ]只有Windows 8.0出货kdvm.dll。Windows 8.1和更新的Windows版本没有。

# 第十章 技术细节

本章内容不需要成功使用VirtualBox。以下是为更熟悉计算机架构和技术的读者提供的附加信息，并希望了解有关VirtualBox如何工作的更多信息。

**10.1。VirtualBox存储其文件的位置**

在VirtualBox中，虚拟机及其设置在XML格式的虚拟机设置文件中进行了描述。此外，大多数虚拟机具有一个或多个虚拟硬盘，通常由磁盘映像（例如，以VDI格式表示）。存储所有这些文件的位置取决于VirtualBox创建该机器的版本。

**10.1.1。由VirtualBox 4.0或更高版本创建的机器**

从版本4.0开始，默认情况下，每个虚拟机在主机上都有一个目录，其中存储该机器的所有文件 - XML设置文件（具有 .vbox文件扩展名）及其磁盘映像。

默认情况下，这个“机器文件夹”被放置在一个名为“VirtualBox VMs”的公用文件夹中，VirtualBox在当前系统用户的主目录中创建。此主目录的位置取决于主机操作系统的约定：

* 在Windows上，这是由SHGetFolderPathWindows系统库Shell32.dll 的功能返回的位置 ，要求用户配置文件。只有在没有这个功能或意外地返回错误的非常旧的Windows版本上，才会有一个基于环境变量的回退：首先 %USERPROFILE%被检查，如果它不存在，那么尝试 %HOMEDRIVE%%HOMEPATH%进行。典型值是 C:\Users\username。
* 在Linux，Mac OS X和Solaris，这通常是从环境变量中获取$HOME，除用户 root为它是一个从帐户数据库中获取（如在组合使用VirtualBox的使用工具造成用户频繁麻烦一种变通方法 sudo，其通过默认情况下不会重置环境变量$HOME）。在Linux和Solaris是典型的价值 /home/username和Mac OS X上 /Users/username。

为简单起见，我们将简化 $HOME如下。使用该约定，所有虚拟机的公用文件夹是 $HOME/VirtualBox VMs。

例如，当您创建名为“示例VM”的虚拟机时，您会发现VirtualBox创建

1. 该文件夹$HOME/VirtualBox VMs/Example VM/，在该文件夹中，
2. 设置文件Example VM.vbox和
3. 虚拟磁盘映像Example VM.vdi。

如果使用[第1.7节“创建第一个虚拟机”中](https://www.virtualbox.org/manual/ch01.html#gui-createvm)所述的“创建新的虚拟机”向导，则这是默认布局。一旦开始使用虚拟机，将显示其他文件：您将在调用的子文件夹中找到日志文件 Logs，一旦您拍摄快照，它们将显示在Snapshots子文件夹中。对于每个虚拟机，您可以在VM设置中更改其快照文件夹的位置。

您可以通过从VirtualBox主窗口的“文件”菜单中选择“首选项”来更改默认的机器文件夹。然后，在弹出的窗口中，单击“常规”选项卡。或者，使用VBoxManage setproperty machinefolder; 请参见[第8.30节“VBoxManage setproperty”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-setproperty)。

**10.1.2。4.0之前由VirtualBox版本创建的机器**

如果您已经从早期版本的VirtualBox升级到VirtualBox 4.0，则可能在早期的文件系统布局中有设置文件和磁盘。

在版本4.0之前，VirtualBox将机器设置文件从虚拟磁盘映像中分离出来。机器设置文件有一个 .xml文件扩展名，位于全局VirtualBox配置目录下的名为“机器”的文件夹中（参见下一节）。所以，例如，在Linux上，这是隐藏的$HOME/.VirtualBox/Machines 目录。默认的硬盘文件夹被称为“硬盘”，也驻留在该.VirtualBox文件夹中。用户可以在全局偏好中更改这两个位置。（VirtualBox 4.0已经放弃了“默认硬盘文件夹”的概念，因为默认情况下磁盘映像现在位于每台机器的文件夹中。）

旧的布局有几个严重的缺点。

1. 将虚拟机从一个主机移动到另一个主机是非常困难的，因为涉及的文件不在同一个文件夹中。此外，所有机器的虚拟介质都在中央VirtualBox设置文件（$HOME/.VirtualBox/VirtualBox.xml）中注册了全局注册表。

要将机器移动到另一个主机，因此移动XML设置文件和磁盘映像（位于不同位置）是不够的，但是来自全局媒体注册表XML的硬盘条目也必须被精心复制，如果机器具有快照并且因此差异化图像，这是不可能的。

1. 在隐藏的.VirtualBox 目录（至少在Linux和Solaris主机上）存储可以增长非常大的虚拟磁盘映像使得许多用户想知道磁盘空间在哪里。

而使用VirtualBox 4.0或更高版本创建的新虚拟机将符合新的布局，为了最大程度的兼容性，旧的虚拟机 *不会*转换为新的布局。否则如果用户从4.0降级到较旧版本的VirtualBox，则机器设置将不可逆转地损坏。

**10.1.3。全局配置数据**

除虚拟机的文件外，VirtualBox还会维护全局配置数据。在VirtualBox 4.3的Linux和Solaris上，这是在隐藏的目录中$HOME/.config/VirtualBox，尽管$HOME/.VirtualBox如果存在与早期版本的兼容性，它将被使用; （在Windows和Linux和Solaris在VirtualBox中4.2和更早的版本），这是中$HOME/.VirtualBox; 在Mac上它驻留在 $HOME/Library/VirtualBox。

如果需要，VirtualBox会自动创建此配置目录。或者，您可以VBOX\_USER\_HOME 通过使用标准XDG\_CONFIG\_HOME变量设置环境变量或另外在Linux或Solaris上提供备用配置目录 。（由于全局 VirtualBox.xml设置文件指向所有其他配置文件，这允许完全在几个VirtualBox配置之间切换。）

最重要的是，在此目录中，VirtualBox存储其全局设置文件，另一个XML文件 VirtualBox.xml。这包括全局配置选项和已注册虚拟机的列表，并指向其XML设置文件。（这个文件的位置和它的目录都没有改变与VirtualBox 4.0。）

在VirtualBox 4.0之前，所有虚拟介质（磁盘映像文件）也都包含在此设置文件的全局注册表中。为了兼容性，如果您升级VirtualBox，并且有4.0版之前创建的机器的介质，此媒体注册表仍然存在。如果没有这样的机器，那么就没有全球媒体注册表; 与VirtualBox 4.0，每台机器的XML文件都有自己的媒体注册表。

在VirtualBox 4.0之前，默认的“Machines”文件夹和默认的“HardDisks”文件夹位于VirtualBox配置目录下（例如 $HOME/.VirtualBox/Machines在Linux上）。如果您从4.0之前的VirtualBox版本进行升级，则这些目录中的文件不会自动移动，以避免向后兼容。

**10.1.4。4.0配置更改摘要**

下表简要概述了旧版本和4.0或更高版本之间的配置更改：

**表10.1。版本4.0或更高版本中的配置更改**

| **设置** | **4.0之前** | **4.0或以上** |
| --- | --- | --- |
| 默认机器文件夹 | $HOME/.VirtualBox/Machines | $HOME/VirtualBox VMs |
| 默认磁盘映像位置 | $HOME/.VirtualBox/HardDisks | 在每台机器的文件夹中 |
| 机器设置文件扩展名 | .xml | .vbox |
| 媒体注册 | 全局VirtualBox.xml 文件 | 每个机器设置文件 |
| 媒体注册 | 需要明确的开/关 | 自动附加 |

**10.1.5。VirtualBox XML文件**

VirtualBox为机器设置文件和全局配置文件使用XML VirtualBox.xml。

所有VirtualBox XML文件都经过版本控制。当创建新的设置文件（例如，由于创建了新的虚拟机）时，VirtualBox会自动使用当前VirtualBox版本的设置格式。如果降级到较早版本的VirtualBox，这些文件可能无法读取。但是，当VirtualBox遇到早期版本的设置文件（例如升级VirtualBox后）时，它会尝试尽可能多地保留设置格式。如果当前设置无法以旧格式表示，则将仅默认升级设置格式，例如，因为启用了早期版本的VirtualBox中不存在的功能。[[46](https://www.virtualbox.org/manual/ch10.html" \l "ftn.idm9684) ]在这种情况下，VirtualBox备份虚拟机配置目录中的旧设置文件。如果您需要回到早期版本的VirtualBox，那么您将需要手动复制这些备份文件。

我们故意不记录VirtualBox XML文件的规格，因为我们必须保留将来修改它们的权利。因此，我们强烈建议您不要手动编辑这些文件。VirtualBox可以通过VBoxManage命令行工具（参见[第8章*VBoxManage*](https://www.virtualbox.org/manual/ch08.html)）及其API（参见[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)）完全访问其配置数据。

**10.2。VirtualBox可执行文件和组件**

VirtualBox被设计为模块化和灵活的。当VirtualBox图形用户界面（GUI）打开并启动VM时，至少有三个进程正在运行：

1. VBoxSVC，VirtualBox服务进程始终在后台运行。这个过程是由第一VirtualBox的客户端程序（图形用户界面，自动启动VBoxManage， VBoxHeadless中，Web服务或其他），并退出最后一个客户端退出后很短的时间。该服务负责簿记，维护所有虚拟机的状态，并提供VirtualBox组件之间的通信。此通信通过COM / XPCOM实现。

**注意**

当我们在这里引用“客户端”时，我们是指特定VBoxSVC服务器进程的本地客户端，而不是网络中的客户端。VirtualBox采用自己的客户端/服务器设计来允许其进程合作，但所有这些进程都在主机操作系统上的同一用户帐户下运行，这对用户来说完全透明。

1. GUI进程，VirtualBox基于跨平台Qt库的客户端应用程序。当没有--startvm 选项启动时，此应用程序充当VirtualBox管理器，显示虚拟机及其设置。然后它将设置和状态更改传达给，VBoxSVC并且还反映通过其他方式影响的更改，例如 VBoxManage。
2. 如果VirtualBox客户端应用程序以--startvm参数启动 ，则会加载包含实际管理程序的VMM库，然后运行虚拟机，并为guest虚拟机提供输入和输出。

任何VirtualBox前端（客户端）将与服务进程通信，并且可以控制和反映当前状态。例如，VM选择器或VM窗口或VBoxManage可用于暂停运行的VM，其他组件将始终反映已更改的状态。

VirtualBox GUI应用程序只是几个可用的前端（客户端）之一。VirtualBox附带的完整列表是：

1. VirtualBox，Qt前端实现管理员和运行虚拟机;
2. VBoxManage，少用户友好但更强大的替代方案，在[第8章，*VBoxManage中*](https://www.virtualbox.org/manual/ch08.html)描述。
3. VBoxSDL，基于SDL库的简单图形前端; 请参见[第9.1节“VBoxSDL，简化VM显示器”](https://www.virtualbox.org/manual/ch09.html#vboxsdl)。
4. VBoxHeadless，VM前端不直接提供任何视频输出和键盘/鼠标输入，但允许通过VirtualBox Remote Desktop Extension进行重定向; 请参见[第7.1.2节“VBoxHeadless，远程桌面服务器”](https://www.virtualbox.org/manual/ch07.html#vboxheadless)。
5. vboxwebsrv，VirtualBox Web服务进程可以远程控制VirtualBox主机。这在VirtualBox软件开发工具包（SDK）参考中有详细描述; 有关详细信息[，](https://www.virtualbox.org/manual/ch11.html)请参阅[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。
6. VirtualBox Python shell，一种Python替代VBoxManage。这也在SDK参考中描述。

在内部，VirtualBox由许多或多或少的独立组件组成。分析VirtualBox内部错误消息或日志文件时，可能会遇到这些问题。这些包括：

* IPRT是一种便携式运行库，可以抽象文件访问，线程，字符串操作等。每当VirtualBox访问主机操作功能时，它都会通过此库进行跨平台的可移植性。
* VMM（虚拟机监视器），虚拟机管理程序的核心。
* EM（执行管理器）控制访客代码的执行。
* REM（重新编译执行监视器）提供CPU指令的软件仿真。
* TRPM（Trap Manager）拦截和处理客户陷阱​​和异常。
* HM（硬件加速管理器）为VT-x和AMD-V提供支持。
* GIM（访客接口管理器）为客户提供各种准虚拟化接口的支持。
* PDM（可插拔设备管理器），VMM和仿真设备之间的抽象接口，可将设备实现与VMM内部分离，并可轻松添加新的仿真设备。通过PDM，第三方开发人员可以向VirtualBox添加新的虚拟设备，而无需更改VirtualBox本身。
* PGM（页面管理器），一个控制访客分页的组件。
* PATM（Patch Manager），修补访客代码，以改善和加快软件虚拟化。
* TM（时间管理器），处理客人的时间和时间的各个方面。
* CFGM（配置管理器）提供了一个树状结构，可以保存虚拟机和所有仿真设备的配置设置。
* SSM（保存的状态管理器），保存并加载VM状态。
* VUSB（虚拟USB），一个将模拟USB控制器与主机和USB设备上的控制器分离的USB层; 这也启用远程USB。
* DBGF（Debug Facility），一个内置的VM调试器。
* VirtualBox模拟了许多设备来提供各种客人所需的硬件环境。大多数这些是在许多PC兼容机器中发现的标准设备，并被客户操作系统广泛支持。对于特定的网络和存储设备，仿真设备可以访问底层硬件有几种选择。这些设备由PDM管理。
* 客人添加各种客户操作系统。这是从虚拟机中安装的代码; 请参阅[第4章，*客户添加*](https://www.virtualbox.org/manual/ch04.html)。
* “Main”组件是特殊的：它将所有上述位连接在一起，是VirtualBox唯一提供的公共API。上面列出的所有客户端进程仅使用此API，并且不会直接访问虚拟机监控程序组件。因此，使用VirtualBox Main API的第三方应用程序可以依赖于它经常被很好地测试，并且VirtualBox的所有功能都被完全暴露的事实。这是在上述VirtualBox SDK中描述的这个API（再次参见[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)）。

**10.3。硬件与软件虚拟化**

VirtualBox允许虚拟机中的软件直接在主机的处理器上运行，但采用一系列复杂技术来拦截会干扰主机的操作。每当客人尝试执行可能对您的计算机及其数据有害的内容时，VirtualBox将进入并执行操作。特别是，对于客户认为访问的大量硬件，VirtualBox会根据您如何配置虚拟机来模拟某个“虚拟”环境。例如，当访客尝试访问硬盘时，VirtualBox将这些请求重定向到您配置为虚拟机虚拟硬盘的任何内容 - 通常是主机上的映像文件。

不幸的是，x86平台从未被设计为虚拟化。如上所述，检测VirtualBox需要控制正在执行的访客代码的情况是困难的。实现这一目标有两种方法：

* 自2006年以来，英特尔和AMD处理器都支持所谓的**“硬件虚拟化”**。这意味着这些处理器可以帮助VirtualBox拦截客户机操作系统可能正在尝试的潜在危险操作，并且还可以使虚拟硬件呈现给虚拟机更为容易。

英特尔和AMD处理器之间的这些硬件功能不同。英特尔将其技术命名为**VT-x** ; AMD称**AMD-V**。英特尔和AMD对虚拟化的支持在很大程度上是完全不同的，但原则上并没有很大的不同。

**注意**

在许多系统上，在VirtualBox可以使用它们之前，首先需要在BIOS中启用硬件虚拟化功能。

* 与其他虚拟化软件相反，对于许多使用场景，VirtualBox *不需要*硬件虚拟化功能。通过复杂的技术，VirtualBox可以将许多客户机操作系统完全虚拟化为**软件**。这意味着即使在不支持硬件虚拟化的旧处理器上也可以运行虚拟机。

即使VirtualBox并不总是需要硬件虚拟化，在以下情况下*需要*启用它：

* 某些罕见的客户端操作系统（如OS / 2）使用了我们的软件虚拟化不支持的非常深奥的处理器指令。对于配置为包含此类操作系统的虚拟机，硬件虚拟化将自动启用。
* VirtualBox的64位客户机支持（添加2.0版）和多处理（SMP，3.0版本）都要求启用硬件虚拟化。（这不是一个限制，因为今天的绝大多数64位和多核CPU都支持硬件虚拟化;这个规则的例外是例如年龄较大的Intel Celeron和AMD Opteron CPU）

**警告**

不要与VirtualBox一起运行其他管理程序（开源或商业虚拟化产品）！通常可以并行*安装*几个管理程序，不要同时*运行*多个虚拟机与竞争管理程序。VirtualBox无法跟踪目前正在尝试在同一台主机上执行的另一个虚拟机管理程序，特别是若有几个产品尝试使用诸如VT-x之类的硬件虚拟化功能，则可能会使整个主机崩溃。此外，在VirtualBox中，您可以在运行多个虚拟机时混合使用软件和硬件虚拟化。在某些情况下，混合VT-x和软件虚拟化虚拟机时，性能损失将不可避免。如果最大性能和低开销是必不可少的，我们建议不要混合虚拟化模式。*但这并不*适用于AMD-V。

**10.4。半虚拟化提供商**

VirtualBox允许暴露一个半虚拟化接口，以便在虚拟机内准确高效地执行软件。这些接口要求客户机操作系统识别它们的存在并利用它们，以利用与VirtualBox管理程序通信的好处。

大多数现代主流的客户操作系统，包括Windows和Linux，都支持一个或多个半虚拟化接口。因此，通常不需要在访客中安装其他软件来利用此功能。

将客户操作系统的半虚拟化提供程序公开不依赖于主机平台的选择。例如， *Hyper-V*半虚拟化提供程序可用于虚拟机在任何主机平台上运行（由VirtualBox支持），而不仅仅是Windows。

VirtualBox提供以下接口：

* **Minimal**：宣布存在虚拟化环境。另外，向客户机操作系统报告TSC和APIC频率。此供应商必须运行任何Mac OS X guest虚拟机。
* **KVM**：提供Linux KVM虚拟机管理程序界面，由2.6.25版本开始由Linux内核识别。VirtualBox的实现目前支持半虚拟化时钟和SMP自旋锁。推荐给Linux客户机提供此提供程序。
* **Hyper-V**：提供Microsoft Hyper-V虚拟机管理程序界面，由Windows 7和更新的操作系统识别。VirtualBox的实现目前支持半虚拟化时钟，APIC频率报告，来宾调试，客机崩溃报告和轻松的定时器检查。推荐给Windows客户机提供此提供程序。

**10.5。关于软件虚拟化的细节**

在没有硬件虚拟化支持的情况下，在x86 CPU上实现虚拟化是一项非常复杂的任务，因为CPU架构不是设计为虚拟化的。这些问题通常可以解决，但是以牺牲性能为代价。因此，虚拟化性能和准确性之间存在着不断的冲突。

x86指令集最初是在20世纪70年代设计的，并在80年代增加了保护模式，并采用286 CPU架构，再次采用Intel 386及其32位架构，发生了重大变化。而386对虚拟化操作（V86模式，Windows 3.x和OS / 2 2.x的“DOS Box”使用）的虚拟化支持有限，但没有提供虚拟化整个架构的支持。

理论上说，软件虚拟化并不是太复杂。除了由硬件提供的四个特权级别（“振铃”）（通常只使用两个）：内核模式为ring 0，用户模式为ring3），需要区分“主机上下文”和“guest语境”。

在“主机上下文”中，一切都好像没有管理程序是活动的。如果您的主机上的另一个应用程序已经安排了CPU时间，这可能是活动模式; 在这种情况下，主机环3模式和主机环0模式。管理程序不涉及。

但是，在“访客上下文”中，虚拟机处于活动状态。只要访客代码在环3中运行，这不是一个问题，因为管理程序可以正确设置页表，并在处理器上本地运行该代码。问题主要在于如何拦截访客内核所做的工作。

这些问题有几种可能的解决方案。一种方法是完全软件仿真，通常涉及重新编译。也就是说，客户端要运行的所有代码被分析，转换成不允许客户端修改或看到CPU的真实状态，然后才执行的形式。这个过程在性能方面显然是非常复杂和昂贵的。（VirtualBox包含一个基于QEMU的重编译器，可以用于纯软件仿真，但是重新编译器仅在特殊情况下被激活，如下所述）。

另一种可能的解决方案是半虚拟化，其中只允许特殊修改的客户机运行。这样，大部分硬件访问是抽象的，通常访问硬件或特权CPU状态的任何功能将被传递到管理程序。半虚拟化可以在标准x86 CPU上实现良好的功能和性能，但只有在客户操作系统可以被修改的情况下才能工作，这显然并不总是如此。

VirtualBox选择不同的方法。当启动一个虚拟机时，通过其支持0的内核驱动，VirtualBox已经设置了主机系统，以便它能够本地运行大部分的访客代码，但是它已经插入了图片的“底部”。然后，它可以在需要时进行控制 - 如果执行特权指令，客户陷阱（特别是因为访问了I / O寄存器，需要虚拟化设备）或外部中断。VirtualBox然后可以处理这个，并且将请求路由到虚拟设备，或者可能将处理这样的事情委托给guest虚拟机或主机OS。因此，在访客上下文中，VirtualBox可以是三种状态之一：

* Guest ring 3代码尽可能地全速运行。故障数量通常会很低（除非客户端允许来自环3的端口I / O，我们不能做，因为我们不希望访客能够访问实际端口）。这也被称为“原始模式”，因为客户端ring-3代码未修改。
* 对于环0中的客户代码，VirtualBox采用了一个讨厌的技巧：它实际上重新配置了guest虚拟机，以使其ring-0代码在环1中运行（通常不会在x86操作系统中使用）。结果，当诸如访客设备驱动程序的访客ring-0代码（实际上在环1中运行）尝试写入I / O寄存器或执行特权指令时，“真实”环0中的VirtualBox管理程序可以接管。
* 管理程序（VMM）可以是活动的。每当发生故障时，VirtualBox会查看有问题的指令，并将其转移到虚拟设备或主机操作系统或客户机操作系统，或者在重新编译器中运行。

特别地，当访客代码禁用中断时，使用重编译器，并且VirtualBox无法确定何时被切换回来（在这些情况下，VirtualBox实际上使用自己的反汇编程序分析访客代码）。另外，某些特权指令（如LIDT）需要特别处理。最后，任何实模式或保护模式代码（例如BIOS代码，DOS访客或任何操作系统启动）都将在重新编译器中运行。

不幸的是，这只能在一定程度上奏效。其中以下情况需要特别处理：

1. 环1中的运行环0代码导致大量额外的指令故障，因为环1不允许执行任何特权指令（其中客户的ring-0包含大量）。对于每个这些故障，VMM必须进入并仿真代码以实现所需的行为。尽管如此，仿真上千个这些故障非常昂贵，严重损害了虚拟化客户的性能。
2. 在x86架构中执行环1有一些从未修复的缺陷。*应该*陷入环1的某些指令 不是。这会影响例如LGDT / SGDT，LIDT / SIDT或POPF / PUSHF指令对。而“加载”操作是有特权的，因此可以被捕获，“存储”指令总是成功。如果客人被允许执行这些，它将看到CPU的真实状态，而不是虚拟化状态。CPUID指令也有同样的问题。
3. 管理程序通常需要保留客人地址空间的一部分（线性地址空间和选择器）以供自己使用。这对于客户操作系统并不完全透明，可能会导致冲突。
4. 由客户机OS中运行的应用程序执行的SYSENTER指令（用于系统调用）始终转换为ring0。但是，这是管理程序运行的位置，而不是客户机操作系统。在这种情况下，即使不希望，管理程序也必须捕获并仿真指令。
5. CPU段寄存器包含一个不可软件访问的“隐藏”描述符缓存。虚拟机管理程序无法读取，保存或恢复此状态，但客户机操作系统可能会使用它。
6. 一些资源必须（并且可以）被管理程序所困扰，但是访问是如此的频繁，这会产生显着的性能开销。一个例子是32位模式下的TPR（任务优先级）寄存器。访问此注册表必须被管理程序所困扰，但某些客户机操作系统（特别是Windows和Solaris）会经常写入该寄存器，从而对虚拟化性能产生不利影响。

为了解决这些性能和安全问题，VirtualBox包含一个代码扫描和分析管理器（CSAM），它拆分访客代码，而Patch Manager（PATM）可以在运行时替换它。

在执行环0代码之前，CSAM递归扫描以发现有问题的指令。然后*，* PATM执行*原位*修补，即它用跳转到管理程序存储器来替换指令，其中集成代码生成器已经放置了更合适的实现。实际上，这是一个非常复杂的任务，因为有很多奇怪的情况被发现和处理正确。因此，由于目前的复杂性，人们可以认为PATM是一种先进 *的原位*重编译器。

此外，每当发生故障时，VirtualBox会分析违规代码，以确定是否可以对其进行修补，以防止将来出现更多故障。这种方法在实践中运行良好，并大大提高了软件虚拟化性能。

**10.6。关于硬件虚拟化的细节**

使用Intel VT-x，有两种不同的CPU操作模式：VMX根模式和非根模式。

* 在根模式下，CPU的运行方式与旧版处理器相似，不需要VT-x支持。有四个特权级别（“ring”），并且支持相同的指令集，并附加了多个虚拟化特定指令。Root模式是没有虚拟化使用的主机操作系统，当虚拟化处于活动状态时，它也被管理程序使用。
* 在非根模式下，CPU操作明显不同。仍然有四个特权环和相同的指令集，但是称为VMCS（虚拟机控制结构）的新结构现在控制CPU操作并确定某些指令的行为。非root模式是客户机系统运行的地方。

从根模式切换到非根模式称为“VM条目”，切换回为“VM退出”。VMCS包括在VM进入和退出时保存/恢复的guest虚拟机和主机状态区域。最重要的是，VMCS控制哪些客户机操作会导致VM退出。

VMCS对客人可以做什么和不能做什么提供相当细粒度的控制。例如，虚拟机管理程序可以允许客户机在阴影控制寄存器中写入某些位，但不允许其他位。这样可以让客人在不中断管理程序的情况下允许客户写入控制位，同时防止他们改变管理程序需要保留完全控制的控制位。VMCS还提供对中断传送和异常的控制。

每当指令或事件导致VM退出时，VMCS包含关于退出原因的信息，通常伴随详细信息。例如，如果对CR0寄存器的写入导致退出，则会记录违规指令，以及对控制寄存器的写入访问导致退出以及关于源和目标寄存器的信息的事实。因此，管理程序可以有效地处理该情况，而不需要上述诸如CSAM和PATM的高级技术。

VT-x固有地避免了软件虚拟化面临的几个问题。客人拥有自己完全独立的地址空间，与管理程序不共享，从而消除潜在的冲突。此外，客户操作系统内核代码在VMX非根模式下以特权环0运行，通过以较低特权级别运行环0代码来消除问题。例如，SYSENTER指令可以转换到环0而不会引起问题。当然，即使在VMX非根模式下的环0中，访客代码的任何I / O访问仍然导致VM退出，从而允许进行设备仿真。

VT-x和AMD-V的最大区别在于AMD-V提供了更完整的虚拟化环境。VT-x要求VMX非根代码在启用分页时运行，这样可以排除实模式代码和非分页保护模式软件的硬件虚拟化。这通常只包括固件和OS装载程序，但仍然使VT-x虚拟机管理程序实现复杂化。AMD-V没有这个限制。

当然硬件虚拟化并不完美。与软件虚拟化相比，VM退出的开销相对较高。这导致仿真需要大量陷阱的设备出现问题。VGA设备的一个例子是16色模式，不仅每个I / O端口访问，而且每次访问帧缓冲存储器都必须被捕获。

**10.7。嵌套分页和VPID**

除了“普通”硬件虚拟化外，您的处理器还可以支持其他复杂技术：[[47](https://www.virtualbox.org/manual/ch10.html" \l "ftn.idm9879) ]

* 称为**“嵌套分页”的**新功能在硬件中实现了一些内存管理，可以大大加速硬件虚拟化，因为这些任务不再需要由虚拟化软件执行。

通过嵌套分页，当将线性转换为物理地址时，硬件提供了另一个间接级别。页表功能如前所述，但是线性地址现在首先转换为“客体物理”地址，而不是直接转换为物理地址。传统的寻呼机制现在存在一组新的寻呼寄存器，从客户物理地址转换为用于访问存储器的主机物理地址。

嵌套分页消除了VM退出和页表访问引起的开销。实质上，使用嵌套页表，客户端可以处理分页，而无需管理程序的干预。嵌套分页显着提高了虚拟化性能。

在AMD处理器上，嵌套分页从巴塞罗那（K10）架构开始就可以使用 - 它们现在称之为“快速虚拟化索引”（RVI）。英特尔增加了对嵌入式分页的支持，他们称之为“扩展页面表”（EPT），以及Core i7（Nehalem）处理器。

如果启用了嵌套分页，VirtualBox管理程序也可以使用**大页面**来减少TLB的使用和开销。这可以提高高达5％的性能提升。要为VM启用此功能，您需要使用 VBoxManage modifyvm --largepages 命令; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

* 在Intel CPU上，另一个称为**“虚拟处理器标识符”（VPID）的**硬件功能可以通过减少处理器翻译后备缓冲区（TLB）的昂贵冲洗的需要，大大加快上下文切换。

要为VM启用这些功能，您需要使用 VBoxManage modifyvm --vtxvpid和 --largepages命令; 请参见[第8.8节“VBoxManage modifyvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm)。

[[46](https://www.virtualbox.org/manual/ch10.html" \l "idm9684) ]例如，在VirtualBox 3.1之前，只能在虚拟机中启用或禁用单个DVD驱动器。如果启用，则始终可以看到IDE控制器的辅助主控。使用VirtualBox 3.1，DVD驱动器可以连接到任意控制器的任意插槽，因此它们可以是IDE控制器或SATA插槽的辅助从站。如果您具有早期版本的机器设置文件，并将VirtualBox升级到3.1，然后将DVD驱动器从默认位置移动，则不能以旧的设置格式表示; XML机器文件将以新格式写入，并保留旧格式的备份文件。

[[47](https://www.virtualbox.org/manual/ch10.html" \l "idm9879) ] VirtualBox 2.0增加了对AMD嵌套分页的支持; 支持英特尔EPT和VPID的版本已经添加了2.1版本。

# 第十一章VirtualBox编程接口

VirtualBox全面支持第三方开发人员。VirtualBox的所谓“主API”揭示了虚拟化引擎的整个功能集。任何想以编程方式控制VirtualBox的人都可以完全记录并提供。

主API通过COM（Windows主机）或XPCOM（在其他主机上）提供给C ++客户端。SOAP，Java和Python也存在桥接器。

所有编程信息（文档，参考信息，标题和其他界面文件以及示例）都已分配到单独的**软件开发工具包（SDK），** 可从[http://www.virtualbox.org](http://www.virtualbox.org/" \t "_top)下载。特别地，SDK附带了PDF格式的“编程指南和参考”，其中包含以前在用户手册本章中的信息。

# 第十二章 问题排查

本章提供常见问题的答案。为了提高您使用VirtualBox的用户体验，建议阅读本部分，了解有关常见故障的更多信息，并获取有关如何使用产品的建议。

**12.1。程序和工具**

**12.1.1。分类和分离问题**

通常情况下，虚拟客户机的行为就像一个物理系统。物理机器遇到的任何问题，虚拟机也会遇到。例如，如果由于外部问题而导致互联网连接丢失，虚拟机将受到物理影响的影响。

如果遇到真正的VirtualBox问题，它有助于首先对问题进行分类和隔离。以下是在报告问题之前应该回答的一些问题：

1. 特定客户操作系统的问题是否具体？特定版本的客户操作系统？特别是与Linux客户相关的问题，问题可能是具体到一定的分发和版本的Linux。
2. 某个主机操作系统的问题是否具体？问题通常不是主机操作系统（因为大多数VirtualBox代码库在所有支持的平台之间共享），但特别是在网络和USB支持领域，主机平台之间存在显着差异。一些GUI相关问题也是主机特定的。
3. 某些主机硬件的问题是否具体？这类问题通常与主机CPU有关。由于VT-x和AMD-V之间的显着差异，问题可能是一种或另一种技术特有的。因为不同的CPU支持不同的功能，所以CPU模型的确可能会有所不同（即使是软件虚拟化），这可能会影响访客CPU操作的某些方面。
4. 特定的虚拟化模式是否具体问题？一些问题可能仅在软件虚拟化模式下发生，其他问题可能会针对硬件虚拟化。
5. 问题特定于客人SMP吗？也就是说，它是否与访客中虚拟CPU（VCPU）的数量有关？使用多个CPU通常会严重影响访客操作系统的内部操作。
6. 问题特定于客人添加吗？在某些情况下，这是一个给定的（例如，共享文件夹问题），在其他情况下可能不太明显（例如，显示问题）。而如果问题是Guest Additions具体的，它是否还具体到某个版本的Additions？
7. 问题具体到某一环境吗？一些问题与VM外部的特定环境有关; 这通常涉及网络设置。外部服务器（如DHCP或PXE）的某些配置可能会暴露出与其他类似服务器不会发生的问题。
8. 问题是否回归？知道一个问题是回归通常使得找到解决方案更容易。在这种情况下，至关重要的是要知道哪个版本受到影响，哪个版本不受影响。

**12.1.2。收集调试信息**

对于问题确定，收集可通过VirtualBox支持分析的调试信息往往很重要。本节包含有关可以获取什么样的信息的信息。

每次VirtualBox启动一个虚拟机时，都会创建一个所谓的**“发行日志文件”**，其中包含有关VM配置和运行时事件的大量信息。日志文件被调用VBox.log 并驻留在VM日志文件文件夹中。通常这将是一个这样的目录：

$ HOME / VirtualBox VMs / {machinename} / Logs

启动虚拟机时，上次运行的配置文件将被重命名为.1，最多 .3。有时出现问题时，看看日志是有用的。此外，当请求对VirtualBox的支持时，提供相应的日志文件是强制性的。

为方便起见，对于每个虚拟机，VirtualBox主窗口可以在窗口中显示这些日志。要访问它，请从左侧的列表中选择一个虚拟机，然后从“机器”窗口中选择“显示日志...”。

发布日志文件（VBox.log）包含丰富的诊断信息，例如主机操作系统类型和版本，VirtualBox版本和构建（32位或64位），客户配置（CFGM）的完整转储，详细有关主机CPU类型和支持的功能的信息，是否启用硬件虚拟化，有关VT-x / AMD-V设置的信息，状态转换（创建，运行，暂停，停止等），来宾BIOS消息，访客添加消息，设备特定的日志条目，在执行结束时，最终的访客状态和浓缩的统计信息。

在崩溃的情况下，收集**崩溃转储**非常重要。主机和客户机崩溃都是如此。有关在Linux，Solaris和OS X系统上启用核心转储的信息，请参阅VirtualBox网站上的核心转储文章。[[48](https://www.virtualbox.org/manual/ch12.html" \l "ftn.idm9948) ]

您还可以使用VBoxManage debugvm创建完整虚拟机的转储; 请参见[第8.40节“VBoxManage debugvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-debugvm)。

对于网络相关的问题，通常有助于捕获网络流量的踪迹。如果流量通过主机上的适配器进行路由，则可以使用Wireshark或类似工具来捕获那里的流量。但是，这通常还包括与VM无关的大量流量。

VirtualBox提供了仅在特定虚拟机的网络适配器上捕获网络流量的功能。有关启用此捕获的信息，请参阅VirtualBox网站上的网络跟踪文章[[49](https://www.virtualbox.org/manual/ch12.html" \l "ftn.idm9956) ]。由VirtualBox创建的跟踪文件.pcap 格式化，可以使用Wireshark轻松分析。

**12.1.3。内置的VM调试器**

VirtualBox包括一个内置的VM调试器，高级用户可能会发现有用。该调试器允许检查并在一定程度上控制VM状态。

**警告**

使用VM调试器风险自负。它不支持它，并且以下文档仅适用于对x86 / AMD64机器指令集具有非常高的熟练程度的高级用户以及PC架构的详细知识。熟悉有关客户操作系统的内部程序也可能非常有帮助。

虚拟机调试器可以在VirtualBox的所有常规生产版本中使用，但默认情况下它将被禁用，因为普通用户几乎不会使用它。有两种访问调试器的方法：

* 虚拟机旁边显示一个调试器控制台窗口
* 通过telnet端口5000 的协议

调试器可以通过三种方式启用：

* 启动虚拟机直接使用VirtualBox --startvm，有一个额外的 --dbg， --debug或者 --debug-command-line争论。有关详细信息，请参阅VirtualBox使用帮助。
* 在启动VirtualBox进程之前设置 VBOX\_GUI\_DBG\_ENABLED或 VBOX\_GUI\_DBG\_AUTO\_SHOW 环境变量true。即使第一个VirtualBox进程是VM选择器窗口，设置这些变量（仅检查它们的存在）也是有效的。随后从选择器启动的VM将启用调试器。
* 在启动VM之前设置GUI/Dbg/Enabled 额外的数据项true。这可以在全局或每个VM的基础上设置。

一个新的“调试”菜单项将被添加到VirtualBox应用程序。此菜单允许用户打开调试器控制台。

VM调试器命令语法在DOS，OS / 2和Windows上使用的Microsoft和IBM调试器上进行了松散建模。熟悉symdeb，CodeView或OS / 2内核调试器的用户将会发现VirtualBox VM调试器很熟悉。

最重要的命令是 help。这将打印所有调试器命令的简要使用帮助。VM调试器支持的命令集经常更改，并且该 help命令始终是最新的。

常用命令的简要说明如下：

* stop - 停止VM执行并启用单步执行
* g - 继续执行VM
* t - 单步教学
* rg/rh/r - 打印guest / hypervisor / current寄存器
* kg/kh/k - 打印guest / hypervisor /当前调用堆栈
* da/db/dw/dd/dq - 以ASCII /字节/字/双字/双字打印内存内容
* u - 拆卸记忆
* dg - 打印客人的GDT
* di - 打印客人的IDT
* dl - 打印客人的LDT
* dt - 打印客人的TSS
* dp\* - 打印客人的页面表结构
* bp/br - 设置一个正常/重新编译器断点
* bl - 列出断点
* bc - 清除断点
* writecore- 将VM核心文件写入磁盘，请参见[第12.1.4节“VM核心格式”](https://www.virtualbox.org/manual/ch12.html#ts_guest-core-format)

请参阅内置的help其他可用命令。

VM调试器支持符号调试，尽管客户代码的符号通常不可用。对于Solaris detectguest虚拟机， 命令将自动确定客户机操作系统版本，并在guest虚拟机的内存中找到内核符号。符号调试可用。对于Linux客户机，这些 detect命令还可以确定客户机操作系统版本，但客户端的内存中没有符号。内核符号/proc/kallsyms在Linux guest虚拟机的文件中可用 。该文件必须复制到主机，例如使用 scp。该 loadmap调试命令可以用来提供给虚拟机调试的符号信息。请注意，该 kallsyms文件包含当前加载的模块的符号; 如果客人

对于所有客人，一个简单的方式来验证正确的符号是加载的k命令。客人通常空闲，从符号信息应清楚客户机操作系统的空闲循环正在执行。

另一组调试器命令是一组 info命令。运行 info help提供完整的使用信息。信息命令提供与VMM的各种仿真设备和方面相关的特殊数据。没有使用info命令的一般准则，使用正确的命令完全取决于正在调查的问题。一些信息命令是：

* cfgm - 打印配置树的分支
* cpuid - 显示客人CPUID的叶子
* ioport - 打印注册的I / O端口范围
* mmio - 打印注册的MMIO范围
* mode - 打印当前的寻呼模式
* pit - 打印i8254 PIT状态
* pic - 打印i8259A PIC状态
* ohci/ehci/xhci - 打印OHCI / EHCI / xHCI USB控制器状态的子集
* pcnet0 - 打印PCnet状态
* vgatext - 打印格式化为标准文本模式的VGA帧缓冲区的内容
* timers - 打印所有VM计时器

info命令的输出通常需要对仿真设备和/或VirtualBox VMM内部部件的深入了解。但是，如果正确使用，所提供的信息是非常宝贵的。

**12.1.4。VM核心格式**

VirtualBox使用64位ELF格式的VM核心文件创建VBoxManage debugvm; 请参见 [第8.40节“VBoxManage debugvm”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-debugvm)。VM核心文件包含VM的内存和CPU转储，可用于调试客户机操作系统。64位ELF对象格式规范可以在这里获得：[http://downloads.openwatcom.org/ftp/devel/docs/elf-64-gen.pdf](http://downloads.openwatcom.org/ftp/devel/docs/elf-64-gen.pdf" \t "_top)。

VM核心格式的总体布局如下：

[ELF 64头] [程序头，类型PT\_NOTE] →偏移到COREDESCRIPTOR [程序头，类型PT\_LOAD] - 每个连续物理内存范围一个 → 范围的内存偏移量 →文件偏移量 [注意头，类型NT\_VBOXCORE] [COREDESCRIPTOR] → Magic →VM核心文件版本 →VBox版本 →vCPU数量等 [Note Header，type NT\_VBOXCPU] - 每个vCPU一个 [vCPU 1注意头] [DBGFCORECPU - vCPU 1 dump] [附加注释+数据] - 当前未使用 [内存转储]

内存描述符包含相对于guest虚拟机的物理地址，而不包含虚拟地址。核心文件中不包括MMIO区域等内存区域。

可以在VirtualBox的来源如下头文件中找到相关的数据结构和定义： include/VBox/dbgfcorefmt.h， include/iprt/x86.h和 src/VBox/Runtime/include/internal/ldrELFCommon.h。

可以使用elfdump和GNU readelf或其他类似实用程序检查VM核心文件 。

**12.2。一般**

**12.2.1。访客在慢主机文件系统上显示基于文件的映像的IDE / SATA错误**

偶尔，一些主机文件系统提供非常差的写入性能，结果导致客户超时执行IDE / SATA命令。这是正常行为，通常不会导致真正的问题，因为客人应重复已超时的命令。但是，如果对映像文件的写入时间超过约15秒，某些客户端（例如某些Linux版本）将会遇到严重问题。但是，如果主机缓存包含大量需要写入的数据，则需要一分钟以上才能完成单个写入。

此问题的症状是，在大型写入或复制操作期间，客户端无法再访问其文件，通常会导致客户端立即挂起。

为了解决这个问题（真正的修复是使用不会出现这种不可接受的写入性能的更快的文件系统），可以在写入一定量的数据之后刷新图像文件。该间隔通常是无限的，但可以为每个VM的磁盘单独配置。

对于IDE磁盘，请使用以下命令：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / piix3ide / 0 / LUN＃[x] / Config / FlushInterval”[b]

对于SATA磁盘，请使用以下命令：

VBoxManage setextradata“VM名称” “VBoxInternal / Devices / ahci / 0 / LUN＃[x] / Config / FlushInterval”[b]

选择IDE的磁盘的值[x]为第一个通道上的主设备为0，第一个通道的从设备为1，第二个通道为主设备为2，第二个通道为主设备为3渠道。对于SATA，使用0到29之间的值。只有磁盘支持此配置选项; 不能为CD / DVD驱动器设置。

间隔[b]的单位是自上次刷新以来写入的字节数。必须选择其值，以免偶尔发生长写延迟。由于正确的刷新间隔取决于主机和主机文件系统的性能，因此找到使问题消失的最佳值需要进行一些实验。1000000到10000000之间的值（1到10兆字节）是一个很好的起点。减少间隔会降低问题的可能性和访客的写入性能。将该值设置为不必要的低，将无需提供任何好处来降低性能。间隔1将导致每次写入操作的刷新，并且在任何情况下都应该解决问题，但是具有严重的写入性能损失。

为[b]提供值为0被视为无限冲突间隔，有效地禁用此解决方法。通过指定[b]的值来删除额外的数据密钥具有相同的效果。

**12.2.2。响应来宾IDE / SATA刷新请求**

如果需要，当客户端发出IDE FLUSH CACHE命令时，可以刷新虚拟磁盘映像。通常这些请求被忽略以提高性能。以下参数仅适用于磁盘驱动器。它们不能设置为DVD驱动器。

要启用IDE磁盘刷新，请执行以下命令：

VBoxManage setextradata“VM名称”“VBoxInternal / Devices / piix3ide / 0 / LUN＃[x] / Config / IgnoreFlush”0

第一个通道上的主设备选择磁盘的值[x]为0，第一个通道的从设备为1，第二个通道为主设备为2，第二个通道为主设备为3。

要启用SATA磁盘刷新，请执行以下命令：

VBoxManage setextradata“VM名称”“VBoxInternal / Devices / ahci / 0 / LUN＃[x] / Config / IgnoreFlush”0

选择磁盘的值[x]可以是0到29之间的值。

请注意，这不影响根据[12.2.1中](https://www.virtualbox.org/manual/ch12.html#ts_config-periodic-flush)描述的配置执行的刷新。通过将值设置为1或通过删除键，可以恢复忽略flush命令的默认值。

**12.2.3。性能随频率提升变化**

许多较新的多核处理器支持某种形式的频率提升，这意味着如果仅使用一个内核，则可以比额定CPU频率运行更快（可能比50％更快甚至更高）。这导致测量的性能随着瞬时整个系统负载的变化而有所不同。确切的行为在很大程度上取决于具体的处理器型号。

因此，利用频率提升的系统上的基准测试可能会产生不稳定和不可重复的结果，特别是如果基准运行较短（大约数秒）。为了获得稳定的结果，基准测试必须在更长的时间内运行，并且除了被测试的VM之外，还要保持恒定的系统负载。

**12.2.4。频率缩放对CPU使用的影响**

在某些硬件平台和操作系统上，CPU频率缩放可能导致CPU使用率报告高度误导。这种情况发生在主机CPU负载很大但不重的情况下，如最大值的15-30％。

大多数操作系统根据花费的时间确定CPU使用情况，例如在一秒钟内测量系统或进程活动的纳秒数。然而，为了节省能源，当系统未完全加载时，现代系统可以显着地降低CPU速度。当然，当CPU以（例如）最大速度的一半运行时，与全速运行相比，相同数量的指令将执行大约两倍的时间。

根据具体的硬件和主机操作系统，这种效果可能会极大地扭曲操作系统报告的CPU使用情况; 报告的CPU使用率可能比CPU以全速运行的速度高出几倍。在主机操作系统和客户操作系统中都可以观察到效果。

**12.2.5。不准确的Windows CPU使用率报告**

Windows（任务管理器，资源监视器）随附的CPU使用率报告工具不需要花费时间来处理硬件中断。如果中断负载很重（每秒数千个中断），CPU使用率可能会大大低估。

此问题会影响Windows作为主机和来宾操作系统。Sysinternals工具（例如Process Explorer）不会遇到此问题。

**12.2.6。主机电源管理造成的性能差**

在某些硬件平台和操作系统上，虚拟化性能受到主机CPU电源管理的负面影响。症状可能是宾客中不稳定的音频或不规则的客人时钟行为。

某些问题可能由固件和/或主机操作系统错误引起。因此，建议更新固件和应用操作系统修复程序。

为了获得最佳的虚拟化性能，如果有这样的设置（不是所有的系统都支持C1E电源状态），系统BIOS中的C1E电源状态将被禁用。在Intel系统上，Intel C State应禁用该设置。禁用其他电源管理设置也可能会提高性能。但是，必须始终考虑性能和功耗之间的平衡。

**12.2.7。GUI：2D视频加速选项显示为灰色**

要在VirtualBox中使用2D视频加速，主机的显卡应该支持某些OpenGL扩展。在启动时，VirtualBox将检查这些扩展，如果测试失败，则此选项将默认为灰色。

要了解为什么它失败，您可以手动执行以下命令：

VBoxTestOGL --log“log\_file\_name”--test 2D

它将逐个列出所需的OpenGL扩展，并显示哪个失败的测试。这通常意味着您在主机上运行过时或配置不当的OpenGL驱动程序。这也意味着你的视频芯片缺乏必要的功能。

**12.3。Windows客人**

**12.3.1。Windows 7中没有USB 3.0支持**

如果Windows 7或Windows Server 2008 R2 guest虚拟机配置为支持USB 3.0（xHCI），则客户机操作系统根本不会有任何USB支持。这是因为Windows 7早于USB 3.0，因此不随附任何xHCI驱动程序; Microsoft也不会通过Windows Update提供任何供应商提供的xHCI驱动程序。

要解决这个问题，有必要在客户端下载并安装Intel xHCI驱动程序。英特尔为Intel 7系列/ C216芯片组的USB 3.0可扩展主机控制器（xHCI）驱动程序提供驱动程序。

请注意，驱动程序仅支持Windows 7和Windows Server 2008 R2。驱动程序包包括对32位和64位操作系统变体的支持。

**12.3.2。更改虚拟机配置后，Windows会蓝屏**

更改某些虚拟机设置可能导致Windows guest虚拟机在启动蓝屏时失败。如果在安装Windows之后更改VM设置，或者将已安装Windows的磁盘映像复制到新创建的具有与原始机器不同的设置的VM时，可能会发生这种情况。

这尤其适用于以下设置：

* 安装Windows后，不得更改ACPI和I / O APIC设置。根据这些硬件功能的存在，Windows安装程序会选择特殊的内核和设备驱动程序版本，如果删除这些硬件功能，将无法启动。（为没有它们而安装的Windows VM启用它们不会造成任何伤害，但是在这种情况下Windows不会使用这些功能。）
* 更改存储控制器硬件也会导致启动失败。如果您将磁盘映像从较早版本的VirtualBox复制到使用较新的VirtualBox版本创建的虚拟机，则这也可能适用于您。使用VirtualBox 2.2将IDE控制器硬件的默认子类型从PIIX3更改为PIIX4。确保这些设置是相同的。

**12.3.3。启用S​​MP的Windows 0x101 bluescreens（IPI超时）**

如果虚拟机配置为具有多个处理器（对称多处理器SMP），某些Windows客户端配置将以0x101错误消息进行崩溃，表示处理器间中断（IPI）超时。这些中断使处理器之间的内存管理同步。

据微软称，这是由于Windows的竞争条件。修补程序可用。[[50](https://www.virtualbox.org/manual/ch12.html" \l "ftn.idm10184) ]如果没有帮助，请将虚拟处理器的数量减少到1。

**12.3.4。Windows 2000安装失败**

安装Windows 2000客户端时，可能会遇到以下问题之一：

* 安装重新启动，通常在组件注册期间。
* 安装使用空的日志文件填满整个硬盘。
* 安装时抱怨安装失败 msgina.dll。

这些问题都是由Windows 2000硬盘驱动程序中的错误引起的。在发出硬盘请求后，Windows驱动程序代码中存在竞争条件，如果操作完成得太快导致损坏，即硬件中断IDE控制器即将到达。使用物理硬件，大多数系统都有一个保证的延迟，所以问题通常在那里被隐藏（然而应该有可能在物理硬件上重现它）。在虚拟环境中，可以立即执行操作（特别是在具有多个CPU的非常快的系统上），并且中断比物理系统更早发信号。解决方案是在发送这样的中断之前引入人为延迟。

VBoxManage setextradata“VM名称”“VBoxInternal / Devices / piix3ide / 0 / Config / IRQDelay”1

这将延迟设置为一毫秒。如果这不能帮助，请将其增加到1到5毫秒之间的值。请注意，这会降低磁盘性能。安装后，您应该可以删除该键（或将其设置为0）。

**12.3.5。如何从Windows访客录制蓝屏信息**

当Windows客户端遇到内核崩溃时，它们显示臭名昭着的蓝屏。根据Windows的配置方式，信息将保留在屏幕上，直到机器重新启动或将自动重新启动。在安装过程中，Windows通常配置为自动重新启动。通过自动重新启动，没有机会记录对于问题确定可能很重要的蓝屏信息。

VirtualBox提供一种在客户端执行重置时停止客户端的方法。为了启用此功能，请发出以下命令：

VBoxManage setextradata“VM名称”“VBoxInternal / PDM / HaltOnReset”1

**12.3.6。32位Windows Server 2003客户端的PCnet驱动程序故障**

某些版本的Windows 2000和2003服务器在32位系统上支持超过4 GB的RAM。如果32位客户操作系统使用分页扩展（将分配给虚拟机的大概3.5 GB的RAM），则Windows Server 2003附带的AMD PCnet网络驱动程序无法加载。

PCnet驱动程序的版本4.38.0.0会出现此问题。驱动程序版本4.51.0.0中的问题已解决，可单独下载。另一种解决方案可能是将模拟的NIC类型更改为Intel PRO / 1000 MT Desktop（82540EM），或将分配给VM的RAM减少到大约3.5 GB或更少。

**12.3.7。Windows Vista中没有网络客户端**

对于Windows Vista，Microsoft已经将VirtualBox用作在1.6.0之前提供的默认虚拟网卡的AMD PCNet卡支持。对于Windows Vista guest虚拟机，VirtualBox默认使用Intel E1000卡。

如果由于某些原因，您仍然想使用AMD卡，则需要从AMD网站下载PCNet驱动程序（仅适用于32位Windows）。您可以使用共享文件夹将其传输到虚拟机，请参阅（参见[第4.3节“共享文件夹”](https://www.virtualbox.org/manual/ch04.html#sharedfolders)）。

**12.3.8。Windows客户端可能会导致高CPU负载**

众所周知，Windows客户端的几个后台应用程序，特别是病毒扫描程序，即使客户机空闲，也会显着增加CPU负载。我们建议尽可能在虚拟客户端中停用病毒扫描程序。

**12.3.9。访问共享文件夹时长时间延迟**

由于在VirtualBox共享文件夹名称服务的解析期间的延迟，可能会降低从Windows guest虚拟机访问共享文件夹的性能。要修复这些延迟，请将以下条目添加到\windows\system32\drivers\etc\lmhostsWindows guest 虚拟机的文件 中：

255.255.255.255 VBOXSVR #PRE 255.255.255.255 VBOXSRV #PRE

进行此更改后，需要重新启动guest虚拟机。

**12.3.10。USB平板电脑坐标错误在Windows 98客人**

如果Windows 98虚拟机配置为使用仿真的USB平板电脑（绝对指点设备），则坐标转换可能不正确，指针被限制在客人屏幕的左上角。

Windows 98中的USB HID（人机接口设备）驱动程序非常旧，不像最近的操作系统（Windows 2000及更高版本，Mac OS X，Solaris）一样处理平板电脑。要解决此问题，请发出以下命令：

VBoxManage setextradata“VM name”“VBoxInternal / USB / HidMouse / 0 / Config / CoordShift”0

要恢复默认行为，请删除该键或将其值设置为1。

**12.3.11。还原快照后，Windows客户端将从Active Directory域中删除**

如果Windows客户机是Active Directory域的成员，并且使用VirtualBox的快照功能，则可能会在恢复较旧的快照后失去此状态。

原因是为了安全起见，Windows定期执行自动机器密码更改。您可以按照 Microsoft的[http://support.microsoft.com/kb/154501](http://support.microsoft.com/kb/154501" \t "_top)文章的说明禁用此功能。

**12.3.12。恢复d3d8.dll和d3d9.dll**

在4.1.8之前的Windows的VirtualBox Guest Additions在选择和安装实验Direct3D支持时没有正确备份原始的d3d8.dll和d3d9.dll系统文件。此过程将使用VirtualBox Guest Additions中的文件替换两个系统文件，以便可以正确处理Direct3D调用。虽然VirtualBox 4.1.8修复了此问题，但Windows Guest Additions安装程序无法修复这些文件。

如果启用了3D加速，并且安装了基本的Direct3D支持，也就是说，没有WDDM（在Windows Vista或更高版本）上，或者在较早的Windows系统（如Windows XP）上，这些文件的损坏也没有任何影响。通过Direct3D的基本支持，Direct3D 8.0和Direct3D 9.0应用程序将直接使用VirtualBox Direct3D文件，因此将按预期运行。

然而，对于WDDM Direct3D支持，为了运行Direct3D 8.0和Direct3D 9.0应用程序，需要最初发货的d3d8.dll和d3d9.dll文件。由于上述系统文件损坏，这些应用程序将不再工作。有关恢复原始d3d8.dll和d3d9.dll系统文件的分步指南，请参阅以下内容，以防在VirtualBox Guest Additions安装程序警告这些不正确的文件或运行Direct3D应用程序时遇到问题。

**注意**

从Windows 7开始，3D桌面（又名Aero）使用DirectX 10进行渲染，以致损坏的d3d8.dll和d3d9.dll系统文件对实际渲染没有影响。

这就是为什么这种检测到的文件损坏不被认为对所有受支持的Windows客户端上的基本Direct3D安装以及在Windows 7和更高版本的guest虚拟机上安装WDDM Direct3D是致命的。

从Windows XP安装光盘中提取d3d8和d3d9.dll：

1. 下载并安装最新版本的7-Zip文件管理器[http // www.7-zip.org](https://www.virtualbox.org/manual/http//www.7-zip.org" \t "_top)
2. 浏览到安装光盘，例如E：\ i386（或64位版本的amd64）
3. 找到文件d3d8.dl\_和d3d9.dl\_，双击它并提取d3d8.dll和d3d9.dll
4. 以安全模式重新启动Windows
5. 将提取的d3d8.dll和d3d9.dll复制到C：\ Windows \ system32和C：\ Windows \ system32 \ dllcache
6. 重启

从Windows XP Service Pack中提取d3d8和d3d9.dll

1. 1，3-6与安装光盘相同
2. 使用“打开内部”打开WindowsXP-KB936929-SP3-x86.exe作为存档并浏览i386目录。

从Vista / Windows7安装CD或Service Pack iso中提取d3d8和d3d9.dll

1. 下载并安装最新版本的7-Zip文件管理器[http // www.7-zip.org](https://www.virtualbox.org/manual/http//www.7-zip.org" \t "_top)
2. 浏览到安装CD，例如E：\ sources
3. 找到文件install.wim并双击它。7-Zip实用程序打开文件后，您将得到一些编号的文件夹。每个数字子文件夹代表不同版本的Windows（Starter，Home Basic等）
4. 进入数字文件夹后，浏览到Windows \ System32（或C：\ Windows \ SysWOW64为64位版本）目录找到d3d8.dll和d3d9.dll并提取
5. 将提取的d3d8.dll和d3d9.dll复制到C：\ Windows \ system32或C：\ Windows \ SysWOW64（来自system32的文件应该转到system32，从SysWOW64到SysWOW64）
6. 重启

**12.3.13。Windows 3.x限制为64 MB RAM**

Windows 3.x guest虚拟机通常限制为64 MB RAM，即使VM分配的内存更多。虽然Windows 3.1理论上能够使用高达512 MB的RAM，但它只能使用通过XMS界面可用的内存。MS-DOS和Microsoft Windows 3.x附带的HIMEM.SYS（Microsoft XMS管理器）的版本在标准PC上最多只能使用64 MB。

这是Microsoft在知识库文章KB 116256中记录的HIMEM.SYS限制。Microsoft知识库文章KB 84388中详细描述了Windows 3.1内存限制。

如果使用不同的XMS提供程序，Windows 3.x guest虚拟机可以使用超过64 MB的RAM。这可能是一个较新的HIMEM.SYS版本（如Windows 98附带的版本）或更强大的第三方内存管理器（如QEMM）。

**12.4。Linux和X11客人**

**12.4.1。Linux客户端可能会导致CPU负载高**

某些Linux客户机可能会导致CPU负载高，即使客机系统看起来是空闲的。这可能是由客户内核的高定时器频率引起的。一些Linux发行版，例如Fedora，发布配置为**1000Hz**定时器频率的Linux内核。我们建议重新编译来宾内核并选择100Hz的定时器频率。

与红帽企业Linux（RHEL）作为发货发布4.7和5.1以及相关的Linux发行版的内核（例如CentOS的和Oracle的Linux）的Linux内核支持的内核参数*分频器= N*。因此，这样的内核支持较低的定时器频率，无需重新编译。我们建议添加内核参数*divider = 10*来选择一个来宾内核定时器频率为100Hz。

**12.4.2。AMD巴塞罗那CPU**

由于Linux内核中的错误，大多数基于Linux的客户将因AMD Phenoms或Barcelona级Opterons而失败。启用I / O-APIC解决问题（请参见[第3.4节“系统设置”](https://www.virtualbox.org/manual/ch03.html#settings-system)）。

**12.4.3。Buggy Linux 2.6内核版本**

Linux内核中的以下错误阻止它们在VirtualBox中正确执行，导致VM引导崩溃：

* Linux内核版本2.6.18（和一些2.6.17版本）引入了可能导致VirtualBox引发崩溃的竞争条件。请使用内核版本2.6.19或更高版本。
* 使用硬件虚拟化并启用I / O APIC，2.6.24-rc6之前的内核可能会在引导时出现以下信息：

内核恐慌 - 不同步：IO-APIC +计时器不起作用！启动 apic = debug并发送报告。然后尝试使用'noapic'选项启动

如果看到此消息，请禁用硬件虚拟化或I / O APIC（请参见[第3.4节“系统设置”](https://www.virtualbox.org/manual/ch03.html#settings-system)），或将guest 虚拟机升级到较新的内核。[[51](https://www.virtualbox.org/manual/ch12.html" \l "ftn.idm10310) ]

**12.4.4。共享剪贴板，自动调整大小和无缝桌面在X11客人**

访客桌面服务在运行X11窗口系统（Solaris，Linux和其他）的客户端由访问服务器提供，该客户机服务 VBoxClient运行在启动桌面会话的用户的ID下，并使用以下命令行自动启动

VBoxClient - clipboard VBoxClient - 显示 VBoxClient --seamless

当您使用常见的桌面环境（Gnome，KDE等）时，X11用户会话启动。如果一个特定的桌面服务不能正常工作，那么值得检查应该提供它的进程是否正在运行。

这些VBoxClient进程在用户主目录中创建文件，该文件在.vboxclient-\*.pid运行时具有表单的名称， 以防止给定的服务被启动两次。由于配置错误，这些文件是由root创建的，并且在服务停止时不会被删除，这将阻止它们在将来的会话中启动。如果服务无法启动，您可能希望检查这些文件是否仍然存在。

**12.5。Solaris客户端**

**12.5.1。较旧的Solaris 10在64位模式下发生崩溃**

Solaris 10发行版包括Solaris 10 8/07（“S10U4”）错误地检测到自2007年以来生产的较新的Intel处理器。此问题导致64位Solaris内核在启动过程中几乎立即崩溃（通常会导致三重故障）在虚拟化和物理环境中。

建议的解决方案至少升级到Solaris 10 5/08（“S10U5”）。替代解决方案包括强制Solaris始终启动32位内核或为错误6574102应用补丁（而Solaris正在使用32位内核）。

**12.5.2。某些Solaris 10版本可能需要很长时间才能使用SMP启动**

当使用多个CPU时，Solaris 10发布5/08（“S10U5”），10/08（“S10U6”）和5/09（“S10U7”）可能需要很长时间才能启动，并可能会在系统控制台关于从磁盘读取的故障。这是Solaris 10中的一个错误，它会影响特定的物理和虚拟配置。这是由于当磁盘中断重新分配给尚未完全初始化的辅助CPU时尝试从引导磁盘读取微代码更新。磁盘读取将超时并失败，触发延迟（约45秒）和警告。

建议的解决方案至少升级到Solaris 10 10/09（“S10U8”），其中包括此问题的修复。替代解决方案包括将虚拟CPU的数量限制在一个或可能使用不同的存储控制器。

**12.5.3。Solaris 8 5/01及更早版本在启动时可能会崩溃**

在Solaris 8 4/01（“S8U4”）上发布的Solaris 2.6,7和8版本在Pentium 4和更高版本的Intel CPU上错误地设置了机器检查异常（MCE）MSR。在虚拟化和物理环境中，问题导致Solaris内核在启动过程中几乎立即崩溃（并且通常会导致三重故障）。Solaris 9及更高版本不受此问题的影响，也不受Solaris 2.5.1及更早版本的影响。

推荐的解决方案至少升级到Solaris 8 7/01（“S8U5”）。替代解决方案包括为错误4408508和4414557应用补丁（在未受影响的系统上）。

**12.6。FreeBSD的客人**

**12.6.1。FreeBSD 10.0可能会挂起xHCI**

如果对FreeBSD 10.0客户端启用了xHCI（USB 3.0）仿真，则客户操作系统将挂起。这是由于客户操作系统不正确地处理MSI（消息信号中断）不与xHCI设备一起使用的系统引起的。

早期的FreeBSD版本中不存在此问题，并在FreeBSD 10.1中修复。

**12.7。Windows主机**

**12.7.1。VBoxSVC出处理COM服务器问题**

VirtualBox利用Microsoft组件对象模型（COM）进行进程间和进程间通信。这允许VirtualBox在不同的虚拟机进程之间共享一个通用配置，并提供基于通用架构的多个用户界面选项。所有全局状态信息和配置由进程维护，该进程VBoxSVC.exe是一个进程外的COM服务器。每当启动VirtualBox进程时，它都要求访问COM服务器，Windows会自动启动进程。请注意，终端用户不应该启动它。

当最后一个进程与COM服务器断开连接时，它会在几秒钟后自动终止。VirtualBox配置（XML文件）由COM服务器维护和拥有，每当服务器运行时，文件都被锁定。

在某些情况下（例如虚拟机意外终止），COM服务器将不会注意到客户端已断开连接并保持活动状态较长时间（10分钟左右），从而保持配置文件锁定。在其他罕见情况下，COM服务器可能会遇到内部错误，其他进程无法初始化。在这些情况下，建议使用Windows任务管理器来杀死进程 VBoxSVC.exe。

**12.7.2。CD / DVD更改无法识别**

如果您已将物理CD / DVD驱动器分配给访客，并且客户没有注意到介质更改时，请确保Windows介质更改通知（MCN）功能未关闭。这在Windows注册表中由以下键表示：

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Cdrom\Autorun

某些应用程序可能会针对Microsoft的建议禁用此密钥。如果设置为0，请将其更改为1并重新启动系统。VirtualBox依靠Windows通知媒体变化。

**12.7.3。使用Microsoft RDP客户端时响应缓慢**

如果通过Microsoft RDP客户端（称为远程桌面连接）连接到虚拟机，则输入之间可能会有较大的延迟（将鼠标移动到菜单上是最明显的情况）并输出。这是因为此RDP客户端在将其发送到RDP服务器之前收集输入一段时间。

可以通过将Windows注册表项设置为比默认值100更小的值来减小间隔。该键最初不存在，并且必须是DWORD类型。其值的单位为毫秒。大约20个值适用于RDP客户端和服务器之间的低带宽连接。大约4个值可用于千兆以太网连接。通常，低于10的值实现了与本地输入设备和运行虚拟机的主机屏幕非常接近的性能。

取决于是否应为单个用户或系统更改设置

HKEY\_CURRENT\_USER \ Software \ Microsoft \ Terminal Server Client \ Min Send Interval

要么

HKEY\_LOCAL\_MACHINE \ Software \ Microsoft \ Terminal Server Client \ Min Send Interval

可以适当设置

**12.7.4。在单个系统上运行iSCSI启动器和目标**

尝试使用主机上运行的iSCSI启动器（例如Microsoft iSCSI Initiator）访问在虚拟机中运行的iSCSI目标时，Windows主机上可能会发生死锁。这是由Windows缓存管理器组件中的缺陷引起的，并导致主机系统响应迟缓几分钟，之后是系统托盘或单独的消息窗口中的“延迟写入失败”错误消息。客人在此期间被阻止，可能会显示错误信息或变得不稳定。

将环境变量设置 VBOX\_DISABLE\_HOST\_DISK\_CACHE为1将启用此问题的解决方法，直到Microsoft解决问题。例如，打开命令提示符窗口并启动VirtualBox，如下所示：

设置VBOX\_DISABLE\_HOST\_DISK\_CACHE = 1 VirtualBox

虽然这会降低客户机磁盘性能（尤其是写入），但它不会影响主机上运行的其他应用程序的性能。

**12.7.5。桥接网络适配器丢失**

如果VM设置的“Networking”部分中没有显示桥接适配器，则通常意味着桥接网络驱动程序在主机上未正确安装。这可能是由于以下原因：

* 主机达到最大允许过滤器数。在这种情况下，MSI日志会提到0x8004a029在NetFlt网络组件安装上返回的 错误代码：

VBoxNetCfgWinInstallComponent：安装失败，hr（0x8004a029）

您可以尝试在Windows注册表中增加以下密钥的最大过滤器计数：

HKEY\_LOCAL\_MACHINE \系统\ CurrentControlSet \控制\网络\ MaxNumFilters

允许的最大数量为14.重启后，尝试重新安装VirtualBox。

* INF缓存已损坏。在这种情况下，安装日志（%windir%\inf\setupapi.log在XP或%windir%\inf\setupapi.dev.log Vista或更高版本上）通常会提到无法找到适用于sun\_VBoxNetFlt或sun\_VBoxNetFltmp组件的驱动程序包 。解决方案是卸载VirtualBox，删除INF缓存（%windir%\inf\INFCACHE.1），重新启动并尝试重新安装VirtualBox

**12.7.6。仅主机网络适配器无法创建**

如果无法创建仅主机适配器（通过Manager或VBoxManage），则INF缓存可能已损坏。在这种情况下，安装日志（%windir%\inf\setupapi.log 在XP或%windir%\inf\setupapi.dev.log Vista或更高版本上）通常会提到无法为sun\_VBoxNetAdp组件找到合适的驱动程序包 。再次，与上述桥接网络问题一样，解决方案是卸载VirtualBox，删除INF缓存（%windir%\inf\INFCACHE.1），重新启动并尝试重新安装VirtualBox。

**12.8。Linux主机**

**12.8.1。Linux内核模块拒绝加载**

如果VirtualBox内核模块（vboxdrv）拒绝加载，即您收到“错误插入vboxdrv：无效参数”，请以（作为root用户）检查dmesg命令的输出，找出加载失败的原因。很可能内核不同意用于编译模块的gcc的版本。确保使用与用于构建内核相同的编译器。

**12.8.2。未找到Linux主机CD / DVD驱动器**

如果您已配置虚拟机使用主机的CD / DVD驱动器，但是这似乎并没有工作，确保当前用户有权限访问对应的Linux设备文件（/dev/hdc或 /dev/scd0或 /dev/cdrom或类似）。在大多数发行版中，用户必须添加到相应的组（通常称为cdrom或 cdrw）。

**12.8.3。未找到Linux主机CD / DVD驱动器（旧版本）**

在较旧的Linux发行版上，如果您的CD / DVD设备有不同的名称，VirtualBox可能无法找到它。在较旧的Linux主机上，VirtualBox执行以下步骤找到您的CD / DVD驱动器：

1. 如果VBOX\_CDROM定义了环境变量，VirtualBox会检查 （见下文）。如果是这样，VirtualBox将省略所有以下检查。
2. VirtualBox测试是否 /dev/cdrom可行。
3. 另外，VirtualBox通过检查来检查当前是否安装了任何CD / DVD驱动器 /etc/mtab。
4. 此外，VirtualBox检查/etc/fstab指向CD / DVD设备的任何条目 。

换句话说，您可以尝试将VBOX\_CDROM设置为包含CD / DVD设备列表，以冒号分隔，例如如下所示：

导出VBOX\_CDROM ='/ dev / cdrom0：/ dev / cdrom1'

在现代Linux发行版中，VirtualBox使用硬件抽象层（hal）来定位CD和DVD硬件。

**12.8.4。找不到Linux主机软盘**

以前的说明（对于CD和DVD驱动器）应用于软盘，除了在较早的发行版上VirtualBox针对/dev/fd\*设备进行测试，默认情况下可以覆盖此 VBOX\_FLOPPY环境变量。

**12.8.5。写入CD / DVD时出现奇怪的客户端IDE错误信息**

如果实验CD / DVD刻录机支持启用不正确的VirtualBox，主机或客户端配置，则可能会访问CD / DVD刻录机的任何尝试失败，并且只会导致客户机内核错误消息（对于Linux客户机）或应用程序错误消息（Windows客户机）。当虚拟机启动时，VirtualBox会执行常规的一致性检查（特别是如果启动虚拟机的用户CD / DVD刻录机的设备不可写入），则会出现错误消息，但是无法检测到所有错误配置。VirtualBox所需的主机和客户机操作系统配置不是特定的，但是在这里列出了与VirtualBox有关的一些常见问题。

必须特别注意使用正确的设备。配置的主机CD / DVD设备文件名（在大多数情况下 /dev/cdrom）必须指向允许写入CD / DVD单元的设备。对于连接到SCSI控制器或连接到Linux SCSI子系统（对于某些SATA控制器通用）的IDE控制器的CD / DVD刻录机，必须参考SCSI设备节点（例如/dev/scd0）。即使对于IDE CD / DVD刻录机，/dev/scd0如果ide-scsi加载了内核模块，则它必须参考相应的SCSI CD-ROM设备节点（例如） 。所有Linux 2.4内核和一些早期2.6内核的CD / DVD刻录机支持需要此模块。许多Linux发行版在系统中检测到CD / DVD刻录机时都会加载该模块，即使内核将支持没有模块的CD / DVD刻录机。/dev/hdcVirtualBox支持使用IDE设备文件（例如），前提是内核支持此功能，并且ide-scsi模块未加载。

类似的规则（除了在guest虚拟机内，CD / DVD刻录机始终是IDE设备）适用于guest虚拟机配置。由于此设置非常常见，因此客户端的默认配置很可能会按预期工作。

**12.8.6。VBoxSVC IPC问题**

在Linux上，VirtualBox利用自定义版本的Mozilla XPCOM（跨平台组件对象模型）进行进程间和进程间通信（IPC）。该过程 VBoxSVC充当不同VirtualBox进程之间的通信中心，并维护全局配置，即XML数据库。当开始VirtualBox的组成部分，过程VBoxSVC和 VBoxXPCOMIPCD自动启动。它们只能从运行的用户帐户访问。VBoxSVC拥有通常所在的VirtualBox配置数据库 ~/.config/VirtualBox或适用于您的操作系统的相应配置目录。当它正在运行时，配置文件被锁定。VBoxSVC各种VirtualBox组件之间的通信， 通过驻留在本地域的套接字进行 /tmp/.vbox-<username>-ipc。如果有通信问题（即VirtualBox应用程序无法通信VBoxSVC），请终止守护程序并删除本地域套接字目录。

**12.8.7。USB不工作**

如果USB在Linux主机上不起作用，请确保当前用户是该vboxusers组的成员 。请记住，组成员资格不会立即生效，而是在下次登录时生效。如果可用，该 newgrp命令可以避免需要注销/登录。

**12.8.8。PAX / grsec内核**

包括grsec补丁（见[http://www.grsecurity.net/](http://www.grsecurity.net/" \t "_top)）和派生的Linux内核必须禁用VBox二进制文件的PAX\_MPROTECT才能启动虚拟机。原因是VBox必须在匿名内存上创建可执行代码。

**12.8.9。Linux内核vmalloc池耗尽**

当在Linux系统上运行大量RAM（大约有20个VM，每个RAM为1 GB）时，其他虚拟机可能无法从内核错误开始，说明vmalloc池已耗尽，应该被扩展。错误消息还会告诉您vmalloc=256MB在内核参数列表中指定 。如果将此参数添加到GRUB或LILO配置中，则内核无法启动（出现诸如“无法安装根分区”等异常错误消息），则可能会遇到内核和初始RAM磁盘的内存冲突。这可以通过在GRUB配置中添加以下参数来解决：

上半场524288

**12.9。Solaris主机**

**12.9.1。无法启动VM，连续内存不足**

如果默认系统设置未更改，则ZFS文件系统将使用几乎所有可用的RAM作为缓存。这可能导致主机内存的严重碎片阻止VirtualBox VM的启动。我们建议通过添加一行来限制ZFS缓存

设置zfs：zfs\_arc\_max = xxxx

到/ etc / system，其中xxxxbytes是可用于ZFS缓存的内存量。

**12.9.2。VM在Solaris 10主机上中断内存不足错误**

32位Solaris 10主机（错误1225025）要求交换空间等于或大于主机的物理内存大小。例如，8 GB物理内存将至少需要8 GB的交换。这可以在Solaris 10安装期间通过选择“自定义安装”和更改默认分区进行配置。

**注意**

此限制仅适用于32位Solaris主机，64位主机不受影响！

对于现有的Solaris 10安装，需要安装附加的交换映像并将其用作交换。因此，如果您有1 GB的交换和8 GB的物理内存，则需要添加7 GB更多的交换。这可以做到如下：

对于ZFS（以root用户身份）：

zfs create -V 8gb / \_ <ZFS卷> \_ / swap swap -a / dev / zvol / dsk / \_ <ZFS卷> \_ / swap

要挂载，如果重新启动后，将以下行添加到/ etc / vfstab中：

/ dev / zvol / dsk / \_ <ZFS卷> \_ / swap - - swap - no -

或者，您可以使用以下方式扩展现有的交换：

zfs set volsize = 8G rpool / swap

并重新启动系统以使更改生效。

对于UFS（以root用户身份）：

mkfile 7g /path/to/swapfile.img swap -a /path/to/swapfile.img

要重新启动后挂载，请将以下行添加到/ etc / vfstab中：

/path/to/swap.img - - swap - 否 -

[[48](https://www.virtualbox.org/manual/ch12.html" \l "idm9948) ][http://www.virtualbox.org/wiki/Core\_dump](http://www.virtualbox.org/wiki/Core_dump" \t "_top)。

[[49](https://www.virtualbox.org/manual/ch12.html" \l "idm9956) ][http://www.virtualbox.org/wiki/Network\_tips](http://www.virtualbox.org/wiki/Network_tips" \t "_top)。

[[50](https://www.virtualbox.org/manual/ch12.html" \l "idm10184) ]请参阅[http://support.microsoft.com/kb/955076](http://support.microsoft.com/kb/955076" \t "_top)。

[[51](https://www.virtualbox.org/manual/ch12.html" \l "idm10310) ]见[http://www.mail-archive.com/git-commits-head@vger.kernel.org/msg30813.html](http://www.mail-archive.com/git-commits-head@vger.kernel.org/msg30813.html" \t "_top) 有关内核修复细节。

# 第十三章 安全指南

**13.1。一般安全原则**

以下原则是安全使用任何应用程序的基础。

保持软件更新

良好安全实践的原则之一是使所有软件版本和补丁保持最新。激活VirtualBox更新通知，以便在新的VirtualBox版本可用时收到通知。更新VirtualBox时，不要忘记更新Guest Additions。保持主机操作系统以及客户机操作系统的最新状态。

限制对关键服务的网络访问

使用正确的方法，例如防火墙，以保护您的计算机和客人免受外部访问。为虚拟机选择适当的组网模式有助于将主机网络与客户端分离，反之亦然。

遵循最低权限原则

最低权限的原则指出，用户应该获得执行其工作所需的最少权限。始终以常规用户身份执行VirtualBox。我们强烈劝阻任何人使用系统权限执行VirtualBox。

在创建配置文件时选择限制性权限，例如创建/ etc / default / virtualbox时，请参见 [第2.3.3.7节“自动安装选项”](https://www.virtualbox.org/manual/ch02.html#linux_install_opts)。模式0600将是首选。

监控系统活动

系统安全性建立在三个支柱上：良好的安全协议，正确的系统配置和系统监控。审计和审查审计记录涉及第三个要求。系统中的每个组件都具有一定程度的监控能力。遵循本文档中的审计意见，并定期监控审计记录。

保持最新的安全信息

Oracle不断改进其软件和文档。每年检查一次此修订版本。

**13.2。安全安装和配置**

**13.2.1。安装概述**

VirtualBox基础软件包只能从可靠的来源下载，例如官网 [http://www.virtualbox.org](http://www.virtualbox.org/" \t "_top)。应使用提供的SHA256校验和来验证软件包的完整性，这些校验和可以在官方网站上找到。

有关受支持主机的通用VirtualBox安装说明，请参见[第2章“ *安装详细信息”*](https://www.virtualbox.org/manual/ch02.html)。

在Windows主机上，安装程序允许禁用USB支持，支持桥接网络，支持仅主机联网和Python语言绑定，请参见[第2.1节“在Windows主机上安装”](https://www.virtualbox.org/manual/ch02.html#installation_windows)。默认情况下启用所有这些功能，但如果任何虚拟机不需要相应的功能，则禁用其中一些功能可能会适用。只有当外部Python应用程序使用VirtualBox API时，才需要Python语言绑定。特别是USB支持和两种网络模式的支持需要在主机上安装Windows内核驱动程序。

一般情况是安装完整的VirtualBox包。安装必须用系统权限完成。所有VirtualBox二进制文件应作为常规用户执行，而不应作为特权用户执行。

Oracle VM VirtualBox扩展包提供了其他功能，必须单独下载和安装，请参见 [第1.5节“安装VirtualBox和扩展包”](https://www.virtualbox.org/manual/ch01.html#intro-installing)。对于基本包，应验证扩展包的SHA256校验和。由于安装需要系统权限，VirtualBox将在安装扩展包时询问系统密码。

**13.2.2。后安装配置**

通常没有需要VirtualBox组件的后安装配置。但是，在Solaris和Linux主机上，必须为执行虚拟机的用户配置适当的权限，以及谁可以访问某些主机资源。例如，Linux用户必须是*vboxus*组的成员才能将USB设备传递给guest 虚拟机。如果从VM访问串行主机接口，则必须授予用户正确的权限才能访问该设备。同样适用于其他资源，如原始分区，DVD / CD驱动器和声音设备。

**13.3。安全功能**

本节概述了VirtualBox提供的特定安全机制。

**13.3.1。安全模型**

VirtualBox的虚拟机监视器（VMM）的一个属性是通过在受保护的环境（虚拟机）中执行它来封装guest虚拟机，虚拟机在主机操作系统上作为用户进程运行。访客不能直接与硬件或其他计算机通信，而只能通过VMM进行通信。VMM向guest虚拟机提供仿真的物理资源和设备，由客户机操作系统访问以执行所需的任务。VM设置控制提供给guest虚拟机的资源，例如guest虚拟机内存量或访客处理器数量（参见[第3.3节“常规设置”](https://www.virtualbox.org/manual/ch03.html#generalsettings)）和该guest虚拟机 启用的功能（例如远程控制，某些屏幕设置和其他）。

**13.3.2。虚拟机的安全配置**

虚拟机配置的几个方面受到安全考虑。

**13.3.2.1。联网**

虚拟机的默认联网模式是NAT，这意味着虚拟机像路由器后面的计算机一样运行，请参见 [第6.3节“网络地址转换（NAT）”](https://www.virtualbox.org/manual/ch06.html#network_nat)。guest虚拟机属于此VM的私有子网的一部分，并且来自外部的guest IP不可见。这种网络模式的工作原理没有任何额外的设置，足以满足许多目的。

如果使用桥接网络，VM就像主机在同一个网络内的计算机一样，请参见[第6.5节“桥接网络”](https://www.virtualbox.org/manual/ch06.html#network_bridged)。在这种情况下，客户机具有与主机相同的网络访问权限，并且可能需要防火墙来保护子网上的其他计算机免受潜在的恶意客户端的影响，并保护客户机免受其他计算机的直接访问。在某些情况下，值得考虑的是在NAT模式下使用特定端口的转发规则，而不是使用桥接网络。

一些设置根本不需要将VM连接到公共网络。内部网络（参见[第6.6节“内部网络”](https://www.virtualbox.org/manual/ch06.html#network_internal)）或仅主机网络（请参见[第6.7节“仅主机联网”](https://www.virtualbox.org/manual/ch06.html#network_hostonly)）通常足以将虚拟机彼此连接或仅将虚拟机与主机连接，但不与主机公共网络。

**13.3.2.2。VRDP远程桌面认证**

当使用Oracle提供的VirtualBox扩展包进行VRDP远程桌面支持时，可以选择使用各种方法配置RDP身份验证。“null”方法非常不安全，应该在公共网络中避免。有关详细信息[，](https://www.virtualbox.org/manual/ch07.html#vbox-auth)请参见[第7.1.5节“RDP身份验证”](https://www.virtualbox.org/manual/ch07.html#vbox-auth)。

**13.3.2.3。剪贴板**

共享剪贴板允许用户在主机和客户端之间共享数据。以“双向”模式启用剪贴板允许访客读取和写入主机剪贴板。“主机到访客”模式和“访客主机”模式限制访问一个方向。如果客人能够访问主机剪贴板，它还可以从剪贴板上共享的主机中潜在地访问敏感数据。

如果客人能够读取和/或写入主机剪贴板，则通过网络连接到访客的远程用户也将获得这种能力，这可能是不可取的。因此，新机器的共享剪贴板被禁用。

**13.3.2.4。共享文件夹**

如果任何主机文件夹与guest虚拟机共享，则通过网络连接到guest虚拟机的远程用户也可以访问这些文件，因为无法为远程用户选择性地禁用文件夹共享机制。

**13.3.2.5。3D图形加速**

通过Guest Additives启用3D图形会使主机面临额外的安全隐患; 请参见[第4.5.1节“硬件3D加速（OpenGL和Direct3D 8/9）”](https://www.virtualbox.org/manual/ch04.html#guestadd-3d)。

**13.3.2.6。CD / DVD passthrough**

启用CD / DVD传送允许客人在CD / DVD驱动器上执行高级操作，请参见[第5.9节“CD / DVD支持”](https://www.virtualbox.org/manual/ch05.html#storage-cds)。这可能会引起安全风险，因为客户可以覆盖CD / DVD介质上的数据。

**13.3.2.7。USB传输**

将USB设备传递给客人可以让客人完全访问这些设备，请参见[第3.10.1节“USB设置”](https://www.virtualbox.org/manual/ch03.html#settings-usb)。例如，除了读取和写入外部U盘的分区的内容之外，访客还将能够读取和写入该磁盘的分区表和硬件数据。

**13.3.3。配置和使用身份验证**

VirtualBox的以下组件可以使用密码进行身份验证：

* 当使用远程iSCSI存储并且存储服务器需要认证时，可以选择性地向该VBoxManage storageattach 命令提供启动器密码。只要不提供设置密码（命令行选项）

--settingspwfile

，这种秘密在机器配置中**未加密**地存储，因此在主机上可能是可读的。请参见[第5.10节“iSCSI服务器”](https://www.virtualbox.org/manual/ch05.html#storage-iscsi)和[第8.18节“VBoxManage storageattach”](https://www.virtualbox.org/manual/ch08.html#vboxmanage-storageattach)。

* 当使用VirtualBox Web服务远程控制VirtualBox主机时，与Web服务的连接以各种方式进行身份验证。这在VirtualBox软件开发工具包（SDK）参考中有详细描述; 请参见[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。

**13.3.4。潜在的不安全行动**

VirtualBox的以下功能可以提出安全问题：

* 通过Guest Additives启用3D图形会使主机面临额外的安全隐患; 请参见[第4.5.1节“硬件3D加速（OpenGL和Direct3D 8/9）”](https://www.virtualbox.org/manual/ch04.html#guestadd-3d)。
* 当传送机器时，机器的存储器内容从一个主机传输到另一个主机的数据流不被加密。因此，能够访问传输数据的网络的第三方可能会拦截该数据。可以使用SSH隧道来保护两台主机之间的连接。但是，当考虑通过不受信任的网络传送VM时，首先要解决的问题是两个虚拟机如何以合理的性能安全地访问同一个虚拟磁盘映像。
* 当页面融合（参见[第4.9.2节“页面融合”](https://www.virtualbox.org/manual/ch04.html#guestadd-pagefusion)）被启用时，侧通道可能打开，允许恶意客户端确定地址空间布局（即通常加载DLL）的另一个VM在同一主机上运行。这种信息泄漏自身是无害的，但是恶意客户可以使用它来通过无关的攻击向量优化对该VM的攻击。如果您不认为这是您的设置中的问题，建议仅启用页面融合。
* 当使用VirtualBox Web服务远程控制VirtualBox主机时，与Web服务（API调用通过SOAP XML传输）的连接未加密，但默认情况下使用纯HTTP。这是一个潜在的安全隐患！有关Web服务的详细信息，请参阅[第11章*VirtualBox编程接口*](https://www.virtualbox.org/manual/ch11.html)。

Web服务默认情况下不启动。请参阅 [第9.21节“自动启动VirtualBox Web服务”](https://www.virtualbox.org/manual/ch09.html#vboxwebsrv-daemon)，了解如何启动此服务以及如何启用SSL / TLS支持。它必须作为常规用户启动，并且只能控制该用户的VM。默认情况下，服务绑定到本地主机，防止任何远程连接。

* 通过UDP隧道网络附件发送的流量未加密。您可以在主机网络级别（使用IPsec）进行加密，也可以在访客网络（如SSH）中使用加密协议。安全属性类似于桥接以太网。
* 由于较旧的Windows版本的缺点，因此不推荐在Windows Vista以上的Service Pack 1上使用VirtualBox。

**13.3.5。加密**

VirtualBox的以下组件使用加密来保护敏感数据：

* 当使用Oracle提供的VirtualBox扩展包进行VRDP远程桌面支持时，RDP数据可以选择加密。有关详细信息[，](https://www.virtualbox.org/manual/ch07.html#vrde-crypt)请参见[第7.1.6节“RDP加密”](https://www.virtualbox.org/manual/ch07.html#vrde-crypt)。只有具有TLS协议的增强型RDP安全方法（RDP5.2）才能提供安全连接。标准RDP安全（RDP4和RDP5.1）容易受到中间人攻击。

# 第十四章 已知限制

由于本章不属于通用的一般使用上的问题，因此这些内容不在这里列出，具体的可以参考其官方手册：

<https://www.virtualbox.org/manual/ch14.html>

# 第十五章 更改日志

本章总结了VirtualBox各个版本之间的变化，因此也会随着新版本发布而变，本文档主要是了解VirtualBox并能够灵活使用它，因此本章内容不在这里列出，具体的参考其官方手册：

<https://www.virtualbox.org/manual/ch15.html>

词汇表

**一个**

ACPI

高级配置和电源接口，用于配置PC硬件和执行电源管理的BIOS和硬件扩展的行业规范。Windows 2000及更高版本以及Linux 2.4及更高版本支持ACPI。Windows只能在安装时启用或禁用ACPI支持。

AHCI

高级主机控制器接口，支持SATA设备（如硬盘）的接口。请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

AMD-V

现代AMD处理器内置的硬件虚拟化功能。请参见[第10.3节“硬件与软件虚拟化”](https://www.virtualbox.org/manual/ch10.html#hwvirt)。

API

应用程序接口。

APIC

高级可编程中断控制器，较新版本的原始PC PIC（可编程中断控制器）。大多数现代CPU都包含片上APIC（“本地APIC”）。许多系统还包含一个I / O APIC（输入输出APIC）作为提供16个以上IRQ的独立芯片。如果Windows 2000及更高版本在安装期间检测到I / O APIC，则使用不同的内核。因此，安装后不能删除I / O APIC。

ATA

高级技术附件，硬盘接口行业标准（与IDE同义）。请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

**乙**

BIOS

基本输入/输出系统，大多数个人电脑内置的固件，负责在计算机开机后启动硬件，然后启动操作系统。VirtualBox随附虚拟机启动时运行的虚拟BIOS。

**C**

COM

Microsoft组件对象模型，用于模块化软件的编程基础架构。COM允许应用程序提供可从各种其他编程语言和应用程序访问的应用程序编程接口。VirtualBox在内部和外部使用COM，为第三方开发人员提供了一个全面的API。

**ð**

DHCP

动态主机配置协议。这允许网络中的网络设备自动获取其IP地址（和其他网络细节），以避免必须配置具有固定IP地址的网络中的所有设备。VirtualBox具有内置的DHCP服务器，当网络配置为NAT时，可将IP地址传递给虚拟机; 请参见[第6章*虚拟网络*](https://www.virtualbox.org/manual/ch06.html)。

**Ë**

EFI

可扩展固件接口，内置于计算机中的固件，旨在替代老化的BIOS。最初由英特尔设计，大多数现代操作系统现在可以在具有EFI而不是内置BIOS的计算机上启动; 请参见[第3.13节“替代固件（EFI）”](https://www.virtualbox.org/manual/ch03.html#efi)。

EHCI

增强的主机控制器接口，实现USB 2.0标准的接口。

**G**

GUI

图形用户界面。通常用作“命令行界面”的反义词，在VirtualBox的上下文中，我们有时将主图形 VirtualBox程序称为“GUI”，以将其与VBoxManage 界面区分开来。

GUID

参见UUID。

**一世**

IDE

集成驱动电子，硬盘接口行业标准。请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

I / O APIC

见APIC。

iSCSI的

互联网SCSI 请参见[第5.10节“iSCSI服务器”](https://www.virtualbox.org/manual/ch05.html#storage-iscsi)。

**中号**

苹果电脑

媒体访问控制，以太网网卡的一部分。MAC地址是一个6字节的数字，用于标识网卡。它通常以十六进制符号形式写入，其中字节由冒号分隔，例如 00:17:3A:5E:CB:08。

MSI

消息信号中断，由ICH9等现代芯片组支持; 请参见[第3.4.1节“主板”选项卡](https://www.virtualbox.org/manual/ch03.html#settings-motherboard)。与传统的基于pin的中断相反，使用MSI，少量数据可能伴随着实际的中断消息。这减少了所需的硬件引脚数量，允许更多的中断和更好的性能。

**ñ**

NAT

网络地址转换 一种共享网络接口的技术，通过该接口，接口根据特定规则修改网络报文的源和/或目标IP地址。路由器和防火墙通常采用屏蔽Internet的内部网络，VirtualBox可以使用NAT轻松地与其虚拟机共享主机的物理网络硬件。请参见[第6.3节“网络地址转换（NAT）”](https://www.virtualbox.org/manual/ch06.html#network_nat)。

**Ø**

OVF

开放虚拟化格式，跨虚拟化产品交换虚拟设备的跨平台行业标准; 请参见 [第1.14节“导入和导出虚拟机”](https://www.virtualbox.org/manual/ch01.html#ovf)。

**P**

PAE

物理地址扩展。即使在32位环境中也能访问超过4 GB的RAM; 请参见[第3.3.2节“高级”选项卡](https://www.virtualbox.org/manual/ch03.html#settings-general-advanced)。

PIC

见APIC。

PXE

预引导执行环境，是从远程网络位置引导PC系统的行业标准。它包括用于IP配置的DHCP和用于文件传输的TFTP。使用UNDI，可以使用硬件独立的驱动程序堆栈从引导代码访问网卡。

**[R**

RDP

远程桌面协议，由Microsoft开发的协议，作为ITU T.128和T.124视频会议协议的扩展。使用RDP，可以使用网络连接从远程位置控制PC系统，数据在两个方向传输。通常从远程机器发送图形更新和音频，并从客户端发送键盘和鼠标输入事件。Oracle提供的VirtualBox扩展包提供了VRDP，这是与Microsoft的RDP实现基本兼容的相关标准的增强型实现。有关详细信息[，](https://www.virtualbox.org/manual/ch07.html#vrde)请参见[第7.1节“远程显示（VRDP支持）”](https://www.virtualbox.org/manual/ch07.html#vrde)。

**小号**

SAS

串行连接SCSI，是硬盘接口的行业标准。请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

SATA

串行ATA，硬盘接口行业标准。请参见 [第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

SCSI

小型计算机系统接口。用于设备之间的数据传输的行业标准，特别是用于存储。请参见[第5.1节“硬盘控制器：IDE，SATA（AHCI），SCSI，SAS，USB MSD，NVMe”](https://www.virtualbox.org/manual/ch05.html#harddiskcontrollers)。

SMP

对称多处理，意味着计算机的资源在多个处理器之间共享。这些可以是几个处理器芯片，或者在现代硬件中更常见的是一个处理器中的多个CPU内核。

SSD

固态驱动器，使用微芯片将数据存储在计算机系统中。与古典硬盘相比，它们没有机械组件，如旋转磁盘。

**Ť**

柏油

广泛使用的归档文件格式。最初，这代表“磁带ARCHive”，并且已被非常早期的Unix版本支持，用于备份磁带上的数据。文件格式今天仍然广泛使用，例如，使用OVF存档（具有 .ova文件扩展名）; 请参见[第1.14节“导入和导出虚拟机”](https://www.virtualbox.org/manual/ch01.html#ovf)。

**ü**

UUID

通常也称为GUID（全局唯一标识符）的通用唯一标识符是一系列数字和字母，可以动态计算并保证是唯一的。一般来说，它被用作识别实体的全局句柄。VirtualBox利用UUID来识别虚拟机，虚拟磁盘映像（VDI文件）和其他实体。

**V**

VM

虚拟机 - VirtualBox允许您在实际硬件之上运行的虚拟计算机。详见[第1.2节“一些术语”](https://www.virtualbox.org/manual/ch01.html#virtintro)。

VMM

虚拟机管理器 - 控制虚拟机执行的VirtualBox组件。有关[VirtualBox组件](https://www.virtualbox.org/manual/ch10.html#technical-components)的列表，请参见[第10.2节“VirtualBox可执行文件和组件”](https://www.virtualbox.org/manual/ch10.html#technical-components)。

VRDE

VirtualBox远程桌面扩展。此接口内置于VirtualBox中，以允许VirtualBox扩展程序包提供对虚拟机的远程访问。Oracle提供的VirtualBox扩展包提供VRDP支持; 有关详细信息[，](https://www.virtualbox.org/manual/ch07.html#vrde)请参见[第7.1节“远程显示（VRDP支持）”](https://www.virtualbox.org/manual/ch07.html#vrde)。

VRDP

见RDP。

VT-X

现代英特尔处理器内置的硬件虚拟化功能。请参见[第10.3节“硬件与软件虚拟化”](https://www.virtualbox.org/manual/ch10.html#hwvirt)。

**X**

的xHCI

扩展主机控制器接口，实现USB 3.0标准的接口。

XML

可扩展标记语言，用于各种文本信息的转移。XML仅指定文档中的数据如何组织，并且不规定如何语义地组织内容。

XPCOM

Mozilla Cross Platform组件对象模型，由Mozilla浏览器项目开发的编程基础架构，类似于Microsoft COM，允许应用程序提供模块化编程接口。VirtualBox在内部和外部使用Linux上的XPCOM，为第三方开发人员提供全面的API。