Oracle® VM VirtualBox

Administrator's Guide for Release 6.0

February 2020
F12469-04

Oracle Legal Notices

Table of Contents

Preface

The *Oracle VM VirtualBox Administrator's Guide* provides information on the advanced features of Oracle VM VirtualBox.

Audience

This document is intended for administrators with previous experience of using Oracle VM VirtualBox. It is assumed that readers are familiar with Web technologies and have a general understanding of Windows and UNIX platforms.

Related Documents

The documentation for this product is available at:

https://docs.oracle.com/en/virtualization/virtualbox/index.html

Conventions

The following text conventions are used in this document:

- **boldface**: Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
- *italic*: Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
- `monospace`: Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Chapter 1 Remote Virtual Machines

1.1. Remote Display (VRDP Support)

Oracle VM VirtualBox can display virtual machines remotely, meaning that a virtual machine can execute on one computer even though the machine will be displayed on a second computer, and the machine will be controlled from there as well, as if the virtual machine was running on that second computer.

For maximum flexibility, Oracle VM VirtualBox implements remote machine display through a generic extension interface called the VirtualBox Remote Desktop Extension (VRDE). The base open source Oracle VM VirtualBox package only provides this interface, while implementations can be supplied by third parties with Oracle VM VirtualBox extension packages, which must be installed separately from the base package. See Installing Oracle VM VirtualBox and Extension Packs.

Oracle provides support for the VirtualBox Remote Display Protocol (VRDP) in such an Oracle VM VirtualBox extension package.

VRDP is a backwards-compatible extension to Microsoft's Remote Desktop Protocol (RDP). As a result, you can use any standard RDP client to control the remote VM.

Even when the extension is installed, the VRDP server is disabled by default. It can easily be enabled on a per-VM basis either in the VirtualBox Manager in the **Display** settings, see Display Settings, or with the `VBoxManage` command, as follows:

```
$ VBoxManage modifyvm VM-name --vrde on
```

By default, the VRDP server uses TCP port `3389`. You will need to change the default port if you run more than one VRDP server, since the port can only be used by one server at a time. You might also need to change it on Windows hosts since the default port might already be used by the RDP server that is built into Windows itself. Ports 5000 through 5050 are typically not used and might be a good choice.

The port can be changed either in the **Display** settings of the graphical user interface or with the `--vrdeport` option of the `VBoxManage modifyvm` command. You can specify a comma-separated list of ports or ranges of ports. Use a dash between two port numbers to specify a range. The VRDP server will bind to *one* of the available ports from the specified list. For example, `VBoxManage modifyvm VM-name --vrdeport 5000,5010-5012` configures the server to bind to one of the ports 5000, 5010, 5011, or 5012. See VBoxManage modifyvm.

The actual port used by a running VM can be either queried with the `VBoxManage showvminfo` command or seen in the GUI on the **Runtime** tab of the **Session Information** dialog, which is accessible from the **Machine** menu of the VM window.

Oracle VM VirtualBox supports IPv6. If the host OS supports IPv6 the VRDP server will automatically listen for IPv6 connections in addition to IPv4.

1.1.1. Common Third-Party RDP Viewers

Since VRDP is backwards-compatible to RDP, you can use any standard RDP viewer to connect to such a remote virtual machine. For this to work, you must specify the IP address of your *host* system, not of the virtual machine, as the server address to connect to. You must also specify the port number that the VRDP server is using.

The following examples are for the most common RDP viewers:

- On Windows, you can use the Microsoft Terminal Services Connector, `mstsc.exe`, that is included with Windows. Press the Windows key + R, to display the **Run** dialog. Enter `mstsc` to start the program. You can also find the program in **Start**, **All Programs**, **Accessories**, **Remote Desktop Connection**. If you use the **Run** dialog, you can enter options directly. For example:

  ```
  mstsc 1.2.3.4:3389
  ```

  Replace `1.2.3.4` with the host IP address, and `3389` with a different port, if necessary.

  

  Note

  - IPv6 addresses must be enclosed in square brackets to specify a port. For example: `mstsc [fe80::1:2:3:4]:3389`
  - When connecting to localhost in order to test the connection, the addresses `localhost` and `127.0.0.1` might not work using `mstsc.exe`. Instead, the address `127.0.0.2[:3389]` has to be used.

- On other systems, you can use the standard open source `rdesktop` program. This ships with most Linux distributions, but Oracle VM VirtualBox also comes with a modified variant of `rdesktop` for remote USB support. See Section 1.1.4, “Remote USB”.

  With `rdesktop`, use a command line such as the following:

  ```
  $ rdesktop -a 16 -N 1.2.3.4:3389
  ```

  Replace `1.2.3.4` with the host IP address, and `3389` with a different port, if necessary. The `-a 16` option requests a color depth of 16 bits per pixel, which we recommend. For best performance, after installation of the guest operating system, you should set its display color depth to the same value. The `-N` option enables use of the NumPad keys.

- You can use the Remmina remote desktop client with VRDP. This application is included with some Linux distributions, such as Debian and Ubuntu.

- If you run the KDE desktop, you can use `krdc`, the KDE RDP viewer. A typical command line is as follows:

  ```
  $ krdc rdp://1.2.3.4:3389
  ```

  Replace `1.2.3.4` with the host IP address, and `3389` with a different port, if necessary. The `rdp://` prefix is required with `krdc` to switch it into RDP mode.

- With Sun Ray thin clients you can use `uttsc`, which is part of the Sun Ray Windows Connector package. See the Sun Ray documentation for details.

1.1.2. VBoxHeadless, the Remote Desktop Server

While any VM started from the VirtualBox Manager is capable of running virtual machines remotely, it is not convenient to have to run the full GUI if you never want to have VMs displayed locally in the first place. In particular, if you are running server hardware whose only purpose is to host VMs, and all your VMs are supposed to run remotely over VRDP, then it is pointless to have a graphical user interface on the server at all. This is especially true for Linux or Oracle Solaris hosts, as the VirtualBox Manager comes with dependencies on the Qt and SDL libraries. This is inconvenient if you would rather not have the X Window system on your server at all.

Oracle VM VirtualBox therefore comes with a front-end called `VBoxHeadless`, which produces no visible output on the host at all, but still can deliver VRDP data. This front-end has no dependencies on the X Window system on Linux and Oracle Solaris hosts.

Note

In legacy releases of Oracle VM VirtualBox, the headless server was called `VBoxVRDP`. For backwards compatibility, the Oracle VM VirtualBox installation still includes an executable with that name.

To start a virtual machine with `VBoxHeadless`, you have the following options:

- Use the `VBoxManage` command, as follows:

```
$ VBoxManage startvm VM-name --type headless
```

  The `--type` option causes Oracle VM VirtualBox to use `VBoxHeadless` as the front-end to the internal virtualization engine, instead of the Qt front-end.

- Use the `VBoxHeadless` command, as follows:

```
VBoxHeadless --startvm uuid|vmname
```

  This way of starting the VM helps troubleshooting problems reported by `VBoxManage startvm`, because you can sometimes see more detailed error messages, especially for early failures before the VM execution is started. In normal situations `VBoxManage startvm` is preferred, since it runs the VM directly as a background process which has to be done explicitly when directly starting with `VBoxHeadless`.

- Start `VBoxHeadless` from the VirtualBox Manager GUI, by pressing the Shift key when starting a virtual machine or by selecting **Headless Start** from the **Machine** menu.

When you use the `VBoxHeadless` command to start a VM, the VRDP server will be enabled according to the VM configuration. You can override the VM's setting using `--vrde` command line parameter. To enable the VRDP server, start the VM as follows:

```
VBoxHeadless --startvm uuid|vmname --vrde on
```

To disable the VRDP server:

```
VBoxHeadless --startvm uuid|vmname --vrde off
```

To have the VRDP server enabled depending on the VM configuration, as for other front-ends:

```
VBoxHeadless --startvm uuid|vmname --vrde config
```

This command is the same as the following:

```
VBoxHeadless --startvm uuid|vmname
```

If you start the VM with `VBoxManage startvm` then the configuration settings of the VM are always used.

1.1.3. Step by Step: Creating a Virtual Machine on a Headless Server

The following instructions describe how to create a virtual machine on a headless server over a network connection. This example creates a virtual machine, establishes an RDP connection and installs a guest operating system. All of these tasks are done without having to touch the headless server. You need the following prerequisites:

- Oracle VM VirtualBox on a server machine with a supported host operating system. The Oracle VM VirtualBox Extension Pack for the VRDP server must be installed, see Section 1.1, “Remote Display (VRDP Support)”. The procedures assume a Linux server is used.

- An ISO file accessible from the server, containing the installation data for the guest operating system to install. Windows XP is used in the example.
- A terminal connection to that host through which you can access a command line, such as `ssh`.
- An RDP viewer on the remote client. See Section 1.1.1, “Common Third-Party RDP Viewers” for examples.

Note that on the server machine, since we will only use the headless server, Qt and the X Window system are not required.

1. On the headless server, create a new virtual machine. For example:

   ```
   VBoxManage createvm --name "Windows XP" --ostype WindowsXP --register
   ```

   If you do not specify `--register`, you will have to manually use the `registervm` command later.

   You do not need to specify `--ostype`, but doing so selects some sensible default values for certain VM parameters. For example, the RAM size and the type of the virtual network device. To get a complete list of supported operating systems you can use the following command:

   ```
   VBoxManage list ostypes
   ```

2. Make sure the settings for the VM are appropriate for the guest operating system that we will install. For example:

   ```
   VBoxManage modifyvm "Windows XP" --memory 256 --acpi on --boot1 dvd --nic1 nat
   ```

3. Create a virtual hard disk for the VM. For example, to create a 10 GB virtual hard disk:

   ```
   VBoxManage createhd --filename "WinXP.vdi" --size 10000
   ```

4. Add an IDE Controller to the new VM. For example:

   ```
   VBoxManage storagectl "Windows XP" --name "IDE Controller"
     --add ide --controller PIIX4
   ```

5. Set the VDI file you created as the first virtual hard disk of the new VM. For example:

   ```
   VBoxManage storageattach "Windows XP" --storagectl "IDE Controller"
    --port 0 --device 0 --type hdd --medium "WinXP.vdi"
   ```

6. Attach the ISO file that contains the operating system installation that you want to install later to the virtual machine. This is done so that the VM can boot from it.

   ```
   VBoxManage storageattach "Windows XP" --storagectl "IDE Controller"
    --port 0 --device 1 --type dvddrive --medium /full/path/to/iso.iso
   ```

7. Enable the VirtualBox Remote Desktop Extension, the VRDP server, as follows:

   ```
   VBoxManage modifyvm "Windows XP" --vrde on
   ```

8. Start the virtual machine using the `VBoxHeadless` command:

   ```
   VBoxHeadless --startvm "Windows XP"
   ```

   If the configuration steps worked, you should see a copyright notice. If you are returned to the command line, then something did not work correctly.

9. On the client machine, start the RDP viewer and connect to the server. See Section 1.1.1, “Common Third-Party RDP Viewers” for details of how to use various common RDP viewers.

The installation routine of your guest operating system should be displayed in the RDP viewer.

1.1.4. Remote USB

As a special feature additional to the VRDP support, Oracle VM VirtualBox also supports remote USB devices over the wire. That is, an Oracle VM VirtualBox guest that runs on one computer can access the USB devices of the remote computer on which the VRDP data is being displayed the same way as USB devices that are connected to the actual host. This enables running of virtual machines on an Oracle VM VirtualBox host that acts as a server, where a client can connect from elsewhere that needs only a network adapter and a display capable of running an RDP viewer. When USB devices are plugged into the client, the remote Oracle VM VirtualBox server can access them.

For these remote USB devices, the same filter rules apply as for other USB devices. See USB Settings. All you have to do is specify Remote, or Any, when setting up these rules.

Accessing remote USB devices is only possible if the RDP client supports this extension. On Linux and Oracle Solaris hosts, the Oracle VM VirtualBox installation provides a suitable VRDP client called `rdesktop-vrdp`. Some versions of `uttsc`, a client tailored for the use with Sun Ray thin clients, also support accessing remote USB devices. RDP clients for other platforms will be provided in future Oracle VM VirtualBox versions.

To make a remote USB device available to a VM, `rdesktop-vrdp` should be started as follows:

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

See Section 5.7.7, “USB Not Working” for further details on how to properly set up the permissions for USB devices. Furthermore it is advisable to disable automatic loading of any host driver on the remote host which might work on USB devices to ensure that the devices are accessible by the RDP client. If the setup was properly done on the remote host, plug and unplug events are visible in the VBox.log file of the VM.

1.1.5. RDP Authentication

For each virtual machine that is remotely accessible using RDP, you can individually determine if and how client connections are authenticated. For this, use the `VBoxManage modifyvm` command with the `--vrdeauthtype` option. See VBoxManage modifyvm. The following methods of authentication are available:

- The **null** method means that there is no authentication at all. Any client can connect to the VRDP server and thus the virtual machine. This is very insecure and only to be recommended for private networks.
- The **external** method provides external authentication through a special authentication library. Oracle VM VirtualBox ships with two special authentication libraries:

  1. The default authentication library, `VBoxAuth`, authenticates against user credentials of the hosts. Depending on the host platform, this means the following:

     - On Linux hosts, `VBoxAuth.so` authenticates users against the host's PAM system.
     - On Windows hosts, `VBoxAuth.dll` authenticates users against the host's WinLogon system.
     - On Mac OS X hosts, `VBoxAuth.dylib` authenticates users against the host's directory service.

     In other words, the external method by default performs authentication with the user accounts that exist on the host system. Any user with valid authentication credentials is accepted. For example, the username does not have to correspond to the user running the VM.

2. An additional library called `VBoxAuthSimple` performs authentication against credentials configured in the `extradata` section of a virtual machine's XML settings file. This is probably the simplest way to get authentication that does not depend on a running and supported guest. The following steps are required:

   a. Enable `VBoxAuthSimple` with the following command:

   ```
   VBoxManage setproperty vrdeauthlibrary "VBoxAuthSimple"
   ```

   b. To enable the library for a particular VM, you must switch authentication to external, as follows:

   ```
   VBoxManage modifyvm VM-name --vrdeauthtype external
   ```

   Replace *`VM-name`* with the VM name or UUID.

   c. You then need to configure users and passwords by writing items into the machine's extradata. Since the XML machine settings file, into whose `extradata` section the password needs to be written, is a plain text file, Oracle VM VirtualBox uses hashes to encrypt passwords. The following command must be used:

   ```
   VBoxManage setextradata VM-name "VBoxAuthSimple/users/user" hash
   ```

   Replace *`VM-name`* with the VM name or UUID, *`user`* with the user name who should be allowed to log in and *`hash`* with the encrypted password. The following command example obtains the hash value for the password `secret`:

   ```
   $ VBoxManage internalcommands passwordhash "secret"
   2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b
   ```

   You then use `VBoxManage setextradata` to store this value in the machine's `extradata` section.

   As a combined example, to set the password for the user `john` and the machine `My VM` to `secret`, use this command:

   ```
   VBoxManage setextradata "My VM" "VBoxAuthSimple/users/john"
       2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b
   ```

- The **guest** authentication method performs authentication with a special component that comes with the Guest Additions. As a result, authentication is not performed on the host, but with the guest user accounts.

  This method is currently still in testing and not yet supported.

In addition to the methods described above, you can replace the default external authentication module with any other module. For this, Oracle VM VirtualBox provides a well-defined interface that enables you to write your own authentication module. This is described in detail in the Oracle VM VirtualBox Software Development Kit (SDK) reference. See Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.

1.1.6. RDP Encryption

RDP features data stream encryption, which is based on the RC4 symmetric cipher, with keys up to 128-bit. The RC4 keys are replaced at regular intervals, every 4096 packets.

RDP provides the following different authentication methods:

- **RDP 4** authentication was used historically. With RDP 4, the RDP client does not perform any checks in order to verify the identity of the server it connects to. Since user credentials can be obtained using a man in the middle (MITM) attack, RDP4 authentication is insecure and should generally not be used.
- **RDP 5.1** authentication employs a server certificate for which the client possesses the public key. This way it is guaranteed that the server possess the corresponding private key. However, as this hard-coded private key became public some years ago, RDP 5.1 authentication is also insecure.
- **RDP 5.2 or later** authentication uses Enhanced RDP Security, which means that an external security protocol is used to secure the connection. RDP 4 and RDP 5.1 use Standard RDP Security. The VRDP server supports Enhanced RDP Security with TLS protocol and, as a part of the TLS handshake, sends the server certificate to the client.

  The `Security/Method` VRDE property sets the desired security method, which is used for a connection. Valid values are as follows:

  - **Negotiate.** Both Enhanced (TLS) and Standard RDP Security connections are allowed. The security method is negotiated with the client. This is the default setting.
  - **RDP.** Only Standard RDP Security is accepted.
  - **TLS.** Only Enhanced RDP Security is accepted. The client must support TLS.

    The version of OpenSSL used by Oracle VM VirtualBox supports TLS versions 1.0, 1.1, 1.2, and 1.3.

  For example, the following command enables a client to use either Standard or Enhanced RDP Security connection:

```
vboxmanage modifyvm VM-name --vrdeproperty "Security/Method=negotiate"
```

  If the `Security/Method` property is set to either Negotiate or TLS, the TLS protocol will be automatically used by the server, if the client supports TLS. However, in order to use TLS the server must possess the Server Certificate, the Server Private Key and the Certificate Authority (CA) Certificate. The following example shows how to generate a server certificate.

  1. Create a CA self signed certificate.

```
openssl req -new -x509 -days 365 -extensions v3_ca \
  -keyout ca_key_private.pem -out ca_cert.pem
```

  2. Generate a server private key and a request for signing.

```
openssl genrsa -out server_key_private.pem
openssl req -new -key server_key_private.pem -out server_req.pem
```

  3. Generate the server certificate.

```
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
```

  The server must be configured to access the required files. For example:

```
vboxmanage modifyvm VM-name \
  --vrdeproperty "Security/CACertificate=path/ca_cert.pem"
```

```
vboxmanage modifyvm VM-name \
  --vrdeproperty "Security/ServerCertificate=path/server_cert.pem"
```

```
vboxmanage modifyvm VM-name \
  --vrdeproperty "Security/ServerPrivateKey=path/server_key_private.pem"
```

As the client that connects to the server determines what type of encryption will be used, with `rdesktop`, the Linux RDP viewer, use the `-4` or `-5` options.

1.1.7. Multiple Connections to the VRDP Server

The VRDP server of Oracle VM VirtualBox supports multiple simultaneous connections to the same running VM from different clients. All connected clients see the same screen output and share a mouse pointer and keyboard focus. This is similar to several people using the same computer at the same time, taking turns at the keyboard.

The following command enables multiple connection mode:

```
VBoxManage modifyvm VM-name --vrdemulticon on
```

1.1.8. Multiple Remote Monitors

To access two or more remote VM displays you have to enable the VRDP multiconnection mode. See Section 1.1.7, “Multiple Connections to the VRDP Server”.

The RDP client can select the virtual monitor number to connect to using the `domain` login parameter (`-d`). If the parameter ends with `@` followed by a number, Oracle VM VirtualBox interprets this number as the screen index. The primary guest screen is selected with `@1`, the first secondary screen is `@2`, and so on.

The Microsoft RDP 6 client does not let you specify a separate domain name. Instead, enter *`domain\username`* in the **Username** field. For example, `@2\`*`name`*. *`name`* must be supplied, and must be the name used to log in if the VRDP server is set up to require credentials. If it is not, you may use any text as the username.

1.1.9. VRDP Video Redirection

The VRDP server can redirect video streams from the guest to the RDP client. Video frames are compressed using the JPEG algorithm allowing a higher compression ratio than standard RDP bitmap compression methods. It is possible to increase the compression ratio by lowering the video quality.

The VRDP server automatically detects video streams in a guest as frequently updated rectangular areas. As a result, this method works with any guest operating system without having to install additional software in the guest. In particular, the Guest Additions are not required.

On the client side, however, currently only the Windows 7 Remote Desktop Connection client supports this feature. If a client does not support video redirection, the VRDP server falls back to regular bitmap updates.

The following command enables video redirection:

```
VBoxManage modifyvm VM-name --vrdevideochannel on
```

The quality of the video is defined as a value from 10 to 100 percent, representing a JPEG compression level, where lower numbers mean lower quality but higher compression. The quality can be changed using the following command:

```
VBoxManage modifyvm VM-name --vrdevideochannelquality 75
```

1.1.10. VRDP Customization

You can disable display output, mouse and keyboard input, audio, remote USB, or clipboard individually in the VRDP server.

The following commands change the corresponding server settings:

```
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableDisplay=1
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableInput=1
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableUSB=1
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableAudio=1
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableClipboard=1
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableUpstreamAudio=1
```

To reenable a feature, use a similar command without the trailing 1. For example:

```
$ VBoxManage modifyvm VM-name --vrdeproperty Client/DisableDisplay=
```

1.2. Teleporting

Oracle VM VirtualBox supports *teleporting*. Teleporting is moving a virtual machine over a network from one Oracle VM VirtualBox host to another, while the virtual machine is running. This works regardless of the host operating system that is running on the hosts. You can teleport virtual machines between Oracle Solaris and Mac OS X hosts, for example.

Teleporting requires that a machine be currently running on one host, which is called the *source*. The host to which the virtual machine will be teleported is called the *target*. The machine on the target is then configured to wait for the source to contact the target. The machine's running state will then be transferred from the source to the target with minimal downtime.

Teleporting happens over any TCP/IP network. The source and the target only need to agree on a TCP/IP port which is specified in the teleporting settings.

At this time, there are a few prerequisites for this to work, as follows:

- On the target host, you must configure a virtual machine in Oracle VM VirtualBox with exactly the same hardware settings as the machine on the source that you want to teleport. This does not apply to settings which are merely descriptive, such as the VM name, but obviously for teleporting to work, the target machine must have the same amount of memory and other hardware settings. Otherwise teleporting will fail with an error message.
- The two virtual machines on the source and the target must share the same storage, hard disks as well as floppy disks and CD/DVD images. This means that they either use the same iSCSI targets or that the storage resides somewhere on the network and both hosts have access to it using NFS or SMB/CIFS.

  This also means that neither the source nor the target machine can have any snapshots.

To configure teleporting, perform the following steps:

1. On the *target* host, configure the virtual machine to wait for a teleport request to arrive when it is started, instead of actually attempting to start the machine. This is done with the following `VBoxManage` command:

   ```
   VBoxManage modifyvm targetvmname --teleporter on --teleporterport port
   ```

   `targetvmname` is the name of the virtual machine on the target host and *`port`* is a TCP/IP port number to be used on both the source and the target hosts. For example, use 6000. See VBoxManage modifyvm.

2. Start the VM on the target host. Instead of running, the VM shows a progress dialog, indicating that it is waiting for a teleport request to arrive.

3. Start the VM on the *source* host as usual. When it is running and you want it to be teleported, issue the following command on the source host:

```
VBoxManage controlvm sourcevmname teleport --host targethost --port port
```

where *sourcevmname* is the name of the virtual machine on the source host, which is the machine that is currently running. *targethost* is the host or IP name of the target host on which the machine is waiting for the teleport request, and *port* must be the same number as specified in the command on the target host. See VBoxManage controlvm.

For testing, you can also teleport machines on the same host. In that case, use localhost as the hostname on both the source and the target host.

Note

In rare cases, if the CPUs of the source and the target are very different, teleporting can fail with an error message, or the target may hang. This may happen especially if the VM is running application software that is highly optimized to run on a particular CPU without correctly checking that certain CPU features are actually present. Oracle VM VirtualBox filters what CPU capabilities are presented to the guest operating system. Advanced users can attempt to restrict these virtual CPU capabilities with the `VBoxManage modifyvm --cpuid` command. See VBoxManage modifyvm.

Chapter 2 Advanced Topics

2.1. Automated Guest Logins

Oracle VM VirtualBox provides Guest Addition modules for Windows, Linux, and Oracle Solaris to enable automated logins on the guest.

When a guest operating system is running in a virtual machine, it might be desirable to perform coordinated and automated logins using credentials from a master login system. Credentials are user name, password, and domain name, where each value might be empty.

2.1.1. Automated Windows Guest Logins

Windows provides a modular system login subsystem, called Winlogon, which can be customized and extended by means of so-called GINA (Graphical Identification and Authentication) modules. In Windows Vista and later releases, the GINA modules were replaced with a new mechanism called credential providers. The Oracle VM VirtualBox Guest Additions for Windows come with both, a GINA and a credential provider module, and therefore enable any Windows guest to perform automated logins.

To activate the Oracle VM VirtualBox GINA or credential provider module, install the Guest Additions using the command line switch `/with_autologon`. All the following manual steps required for installing these modules will be then done by the installer.

To manually install the Oracle VM VirtualBox GINA module, extract the Guest Additions as shown in Manual File Extraction, and copy the `VBoxGINA.dll` file to the Windows `SYSTEM32` directory. In the registry, create the following key with a value of `VBoxGINA.dll`:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL
```

Note

The Oracle VM VirtualBox GINA module is implemented as a wrapper around the `MSGINA.DLL` standard Windows GINA module. As a result, it might not work correctly with third-party GINA modules.

To manually install the Oracle VM VirtualBox credential provider module, extract the Guest Additions as shown in Manual File Extraction and copy the `VBoxCredProv.dll` file to the Windows `SYSTEM32` directory. In the registry, create the following keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Authentication\Credential Providers\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}

HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}

HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}\InprocServer32
```

All default values, the key named `Default`, must be set to `VBoxCredProv`.

Create the following string and assign it a value of `Apartment`.

```
HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}\InprocServer32\ThreadingModel
```

To set credentials, use the following command on a *running* VM:

```
$ VBoxManage controlvm "Windows XP" setcredentials "John Doe" "secretpassword" "DOMTEST"
```

While the VM is running, the credentials can be queried by the Oracle VM VirtualBox login modules, GINA or credential provider, using the Oracle VM VirtualBox Guest Additions device driver. When Windows is

in *logged out* mode, the login modules will constantly poll for credentials and if they are present, a login will be attempted. After retrieving the credentials, the login modules will erase them so that the above command will have to be repeated for subsequent logins.

For security reasons, credentials are not stored in any persistent manner and will be lost when the VM is reset. Also, the credentials are write-only. There is no way to retrieve the credentials from the host side. Credentials can be reset from the host side by setting empty values.

Depending on the Windows guest version, the following restrictions apply:

- For **Windows XP guests.** The login subsystem needs to be configured to use the classic login dialog, as the Oracle VM VirtualBox GINA module does not support the Windows XP-style welcome dialog.
- **Windows Vista, Windows 7, Windows 8, and Windows 10 guests.** The login subsystem does not support the so-called Secure Attention Sequence, `Ctrl+Alt+Del`. As a result, the guest's group policy settings need to be changed to not use the Secure Attention Sequence. Also, the user name given is only compared to the true user name, not the user friendly name. This means that when you rename a user, you still have to supply the original user name as Windows never renames user accounts internally.
- Automatic login handling of the built-in **Windows Remote Desktop Service**, formerly known as Terminal Services, is disabled by default. To enable it, create the following registry key with a `DWORD` value of `1`.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox Guest Additions\AutoLogon
```

The following command forces Oracle VM VirtualBox to keep the credentials after they were read by the guest and on VM reset:

```
$ VBoxManage setextradata "Windows XP" VBoxInternal/Devices/VMMDev/0/Config/KeepCredentials 1
```

Note that this is a potential security risk, as a malicious application running on the guest could request this information using the proper interface.

2.1.2. Automated Linux and UNIX Guest Logins

Oracle VM VirtualBox provides a custom PAM module (Pluggable Authentication Module) which can be used to perform automated guest logins on platforms which support this framework. Virtually all modern Linux and UNIX distributions rely on PAM.

For automated logins on Ubuntu, or Ubuntu-derived, distributions using LightDM as the display manager. See Section 2.1.2.1, “Oracle VM VirtualBox Greeter for Ubuntu/LightDM”.

The `pam_vbox.so` module itself *does not* do an actual verification of the credentials passed to the guest OS. Instead it relies on other modules such as `pam_unix.so` or `pam_unix2.so` down in the PAM stack to do the actual validation using the credentials retrieved by `pam_vbox.so`. Therefore `pam_vbox.so` has to be on top of the authentication PAM service list.

Note

The `pam_vbox.so` module only supports the `auth` primitive. Other primitives such as `account`, `session`, or `password` are not supported.

The `pam_vbox.so` module is shipped as part of the Guest Additions but it is not installed and/or activated on the guest OS by default. In order to install it, it has to be copied from `/opt/VBoxGuestAdditions-`*`version`*`/other/` to the security modules directory. This is usually `/lib/`

security/ on 32-bit Linux guests or /lib64/security/ on 64-bit Linux guests. Please refer to your guest OS documentation for the correct PAM module directory.

For example, to use pam_vbox.so with a Ubuntu Linux guest OS and the GNOME Desktop Manager (GDM) to log in users automatically with the credentials passed by the host, configure the guest OS as follows:

1. Copy the pam_vbox.so module to the security modules directory. In this case, /lib/security.
2. Edit the PAM configuration file for GDM, found at /etc/pam.d/gdm. Add the line auth requisite pam_vbox.so at the top. Additionally, in most Linux distributions there is a file called /etc/pam.d/common-auth. This file is included in many other services, like the GDM file mentioned above. There you also have to add the line auth requisite pam_vbox.so.
3. If authentication against the shadow database using pam_unix.so or pam_unix2.so is desired, the argument try_first_pass for pam_unix.so or use_first_pass for pam_unix2.so is needed in order to pass the credentials from the Oracle VM VirtualBox module to the shadow database authentication module. For Ubuntu, this needs to be added to /etc/pam.d/common-auth, to the end of the line referencing pam_unix.so. This argument tells the PAM module to use credentials already present in the stack, such as the ones provided by the Oracle VM VirtualBox PAM module.

Warning

An incorrectly configured PAM stack can effectively prevent you from logging into your guest system.

To make deployment easier, you can pass the argument debug right after the pam_vbox.so statement. Debug log output will then be recorded using syslog.

Note

By default, pam_vbox does not wait for credentials to arrive from the host. When a login prompt is shown, for example by GDM/KDM or the text console, and pam_vbox does not yet have credentials it does not wait until they arrive. Instead the next module in the PAM stack, depending on the PAM configuration, will have the chance for authentication.

pam_vbox supports various guest property parameters that are located in /VirtualBox/GuestAdd/PAM/. These parameters allow pam_vbox to wait for credentials to be provided by the host and optionally can show a message while waiting for those. The following guest properties can be set:

- CredsWait: Set to 1 if pam_vbox should start waiting until credentials arrive from the host. Until then no other authentication methods such as manually logging in will be available. If this property is empty or gets deleted no waiting for credentials will be performed and pam_vbox will act like before. This property must be set read-only for the guest (RDONLYGUEST).
- CredsWaitAbort: Aborts waiting for credentials when set to any value. Can be set from host and the guest.
- CredsWaitTimeout: Timeout, in seconds, to let pam_vbox wait for credentials to arrive. When no credentials arrive within this timeout, authentication of pam_vbox will be set to failed and the next PAM module in chain will be asked. If this property is not specified, set to 0 or an invalid value, an infinite timeout will be used. This property must be set read-only for the guest (RDONLYGUEST).

To customize pam_vbox further there are the following guest properties:

- CredsMsgWaiting: Custom message showed while pam_vbox is waiting for credentials from the host. This property must be set read-only for the guest (RDONLYGUEST).

- `CredsMsgWaitTimeout`: Custom message showed when waiting for credentials by `pam_vbox` has timed out. For example, they did not arrive within time. This property must be set read-only for the guest (`RDONLYGUEST`).

Note

If a `pam_vbox` guest property does not have the correct flag set (`RDONLYGUEST`) the property is ignored and, depending on the property, a default value will be used. This can result in pam_vbox not waiting for credentials. Consult the appropriate syslog file for more information and use the `debug` option.

2.1.2.1. Oracle VM VirtualBox Greeter for Ubuntu/LightDM

Oracle VM VirtualBox comes with a greeter module, named `vbox-greeter`, that can be used with LightDM. LightDM is the default display manager for Ubuntu Linux and therefore can also be used for automated guest logins.

`vbox-greeter` does not need the `pam_vbox` module described in Section 2.1.2, "Automated Linux and UNIX Guest Logins"in order to function. It comes with its own authentication mechanism provided by LightDM. However, to provide maximum flexibility both modules can be used together on the same guest.

As with the `pam_vbox` module, `vbox-greeter` is shipped as part of the Guest Additions but it is not installed or activated on the guest OS by default. To install `vbox-greeter` automatically upon Guest Additions installation, use the `--with-autologon` option when starting the `VBoxLinuxAdditions.run` file:

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

For manual or postponed installation, copy the `vbox-greeter.desktop` file from `/opt/VBoxGuestAdditions-<version>/other/` to the `xgreeters` directory, which is usually `/usr/share/xgreeters/`. See your guest OS documentation for the name of the correct LightDM greeter directory.

The `vbox-greeter` module is installed by the Oracle VM VirtualBox Guest Additions installer and is located in `/usr/sbin/`. To enable `vbox-greeter` as the standard greeter module, edit the file `/etc/lightdm/lightdm.conf` as follows:

```
[SeatDefaults]
greeter-session=vbox-greeter
```

Note

- The LightDM server must be fully restarted in order for `vbox-greeter` to be used as the default greeter. As `root` on Ubuntu, run `service lightdm --full-restart` or restart the guest.
- `vbox-greeter` is independent of the graphical session you choose, such as Gnome, KDE, or Unity. However, `vbox-greeter` does require FLTK 1.3 or later to implement its own user interface.

There are numerous guest properties which can be used to further customize the login experience. For automatically logging in users, the same guest properties apply as for `pam_vbox`. See Section 2.1.2, "Automated Linux and UNIX Guest Logins".

In addition to the previously mentioned guest properties, `vbox-greeter` enables you to further customize its user interface. The following guest properties are located in the `/VirtualBox/GuestAdd/Greeter/` directory:

- `HideRestart`: Set to 1 if `vbox-greeter` should hide the button to restart the guest. This property must be set read-only for the guest (`RDONLYGUEST`).
- `HideShutdown`: Set to 1 if `vbox-greeter` should hide the button to shutdown the guest. This property must be set read-only for the guest (`RDONLYGUEST`).
- `BannerPath`: Path to a `.PNG` file to use as a banner image on the top of the greeter. The image size must be 460 x 90 pixels, any bit depth. This property must be set read-only for the guest (`RDONLYGUEST`).
- `UseTheming`: Set to 1 for turning on the following theming options. This property must be set read-only for the guest (`RDONLYGUEST`).
- `Theme/BackgroundColor`: Hexadecimal RRGGBB color for the background. This property must be set read-only for the guest (`RDONLYGUEST`).
- `Theme/LogonDialog/HeaderColor`: Hexadecimal RRGGBB foreground color for the header text. This property must be set read-only for the guest (`RDONLYGUEST`).
- `Theme/LogonDialog/BackgroundColor`: Hexadecimal RRGGBB color for the login dialog background. This property must be set read-only for the guest (`RDONLYGUEST`).
- `Theme/LogonDialog/ButtonColor`: Hexadecimal RRGGBB background color for the login dialog button. This property must be set read-only for the guest (`RDONLYGUEST`).

Note

The same restrictions for the guest properties above apply as for the ones specified in the `pam_vbox` section.

2.2. Advanced Configuration for Windows Guests

2.2.1. Automated Windows System Preparation

Microsoft offers a system preparation tool called Sysprep, to prepare a Windows system for deployment or redistribution. Some Windows releases include Sysprep on the installation medium, but the tool is also available for download from the Microsoft web site. In a standard For most Windows versions, Sysprep is included in a default installation. Sysprep mainly consists of an executable called `sysprep.exe` which is invoked by the user to put the Windows installation into preparation mode.

The Guest Additions offer a way to launch a system preparation on the guest operating system in an automated way, controlled from the host system. See Guest Control of Applications for details of how to use this feature with the special identifier `sysprep` as the program to execute, along with the user name `sysprep` and password `sysprep` for the credentials. Sysprep is then started with the required system rights.

Note

Specifying the location of `sysprep.exe` is **not possible**. Instead the following paths are used, based on the Windows release:

- `C:\sysprep\sysprep.exe` for Windows XP and earlier
- `%WINDIR%\System32\sysprep\sysprep.exe` for Windows Vista and later

The Guest Additions will automatically use the appropriate path to execute the system preparation tool.

2.3. Advanced Configuration for Linux and Oracle Solaris Guests

2.3.1. Manual Setup of Selected Guest Services on Linux

The Oracle VM VirtualBox Guest Additions contain several different drivers. If you do not want to configure them all, use the following command to install the Guest Additions:

```
$ sh ./VBoxLinuxAdditions.run no_setup
```

After running this script, run the `rcvboxadd setup` command as `root` to compile the kernel modules.

On some 64-bit guests, you must replace `lib` with `lib64`. On older guests that do not run the `udev` service, you must add the `vboxadd` service to the default runlevel to ensure that the modules are loaded.

To set up the time synchronization service, add the `vboxadd-service` service to the default runlevel. To set up the X11 and OpenGL part of the Guest Additions, run the `rcvboxadd-x11 setup` command. Note that you do not need to enable additional services.

Use the `rcvboxadd setup` to recompile the guest kernel modules.

After compilation, reboot your guest to ensure that the new modules are loaded.

2.3.2. Guest Graphics and Mouse Driver Setup in Depth

This section assumes that you are familiar with configuring the X.Org server using xorg.conf and optionally the newer mechanisms using hal or udev and xorg.conf.d. If not you can learn about them by studying the documentation which comes with X.Org.

The Oracle VM VirtualBox Guest Additions includes drivers for X.Org. By default these drivers are in the following directory:

`/opt/VBoxGuestAdditions-`*`version`*`/other/`

The correct versions for the X server are symbolically linked into the X.Org driver directories.

For graphics integration to work correctly, the X server must load the `vboxvideo` driver. Many recent X server versions look for it automatically if they see that they are running in Oracle VM VirtualBox. For an optimal user experience, the guest kernel drivers must be loaded and the Guest Additions tool `VBoxClient` must be running as a client in the X session.

For mouse integration to work correctly, the guest kernel drivers must be loaded. In addition, for legacy X servers the correct `vboxmouse` driver must be loaded and associated with `/dev/mouse` or `/dev/psaux`. For most guests, a driver for a PS/2 mouse must be loaded and the correct vboxmouse driver must be associated with `/dev/vboxguest`.

The Oracle VM VirtualBox guest graphics driver can use any graphics configuration for which the virtual resolution fits into the virtual video memory allocated to the virtual machine, minus a small amount used by the guest driver, as described in Display Settings. The driver will offer a range of standard modes at least up to the default guest resolution for all active guest monitors. The default mode can be changed by setting the output property VBOX_MODE to "<width>x<height>" for any guest monitor. When VBoxClient and the kernel drivers are active this is done automatically when the host requests a mode change. The driver for older versions can only receive new modes by querying the host for requests at regular intervals.

With legacy X Servers before version 1.3, you can also add your own modes to the X server configuration file. Add them to the "Modes" list in the "Display" subsection of the "Screen" section. For example, the following section has a custom 2048x800 resolution mode added:

```
Section "Screen"
        Identifier    "Default Screen"
        Device        "VirtualBox graphics card"
        Monitor       "Generic Monitor"
        DefaultDepth  24
        SubSection "Display"
                Depth         24
                Modes         "2048x800" "800x600" "640x480"
        EndSubSection
EndSection
```

2.4. CPU Hot-Plugging

With virtual machines running modern server operating systems, Oracle VM VirtualBox supports CPU hot-plugging.

On a physical computer CPU hot-plugging would mean that a CPU can be added or removed while the machine is running. Oracle VM VirtualBox supports adding and removing of virtual CPUs while a virtual machine is running.

CPU hot-plugging works only with guest operating systems that support the feature. So far this applies only to Linux and Windows Server. Windows supports only hot-add, while Linux supports hot-add and hot-remove. To use this feature with more than 8 CPUs, a 64-bit Linux guest is required.

CPU hot-plugging is done using the `VBoxManage` command-line interface. First, hot-plugging needs to be enabled for a virtual machine:

```
$ VBoxManage modifyvm VM-name --cpuhotplug on
```

The `--cpus` option is used to specify the maximum number of CPUs that the virtual machine can have:

```
$ VBoxManage modifyvm VM-name --cpus 8
```

When the VM is off, you can then add and remove virtual CPUs with the `VBoxManage modifyvm --plugcpu` and `VBoxManage modifyvm --unplugcpu` commands, which take the number of the virtual CPU as a parameter, as follows:

```
$ VBoxManage modifyvm VM-name --plugcpu 3
$ VBoxManage modifyvm VM-name --unplugcpu 3
```

Note that CPU 0 can never be removed.

While the VM is running, CPUs can be added and removed with the `VBoxManage controlvm plugcpu` and `VBoxManage controlvm unplugcpu` commands instead, as follows:

```
$ VBoxManage controlvm VM-name plugcpu 3
$ VBoxManage controlvm VM-name unplugcpu 3
```

See VBoxManage modifyvm and VBoxManage controlvm for details.

With Linux guests, the following applies:

To prevent ejection while the CPU is still used it has to be ejected from within the guest before. The Linux Guest Additions contain a service which receives hot-remove events and ejects the CPU. Also, after a CPU is added to the VM it is not automatically used by Linux. The Linux Guest Additions service will take care of that if installed. If not a CPU can be started with the following command:

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

2.5. PCI Passthrough

When running on Linux hosts with a kernel version later than `2.6.31`, experimental host PCI devices passthrough is available.

Note

The PCI passthrough module is shipped as an Oracle VM VirtualBox extension package, which must be installed separately. See Installing Oracle VM VirtualBox and Extension Packs.

This feature enables a guest to directly use physical PCI devices on the host, even if host does not have drivers for this particular device. Both, regular PCI and some PCI Express cards, are supported. AGP and certain PCI Express cards are not supported at the moment if they rely on Graphics Address Remapping Table (GART) unit programming for texture management as it does rather non-trivial operations with pages remapping interfering with IOMMU. This limitation may be lifted in future releases.

To be fully functional, PCI passthrough support in Oracle VM VirtualBox depends upon an IOMMU hardware unit. If the device uses bus mastering, for example it performs DMA to the OS memory on its own, then an IOMMU is required. Otherwise such DMA transactions may write to the wrong physical memory address as the device DMA engine is programmed using a device-specific protocol to perform memory transactions. The IOMMU functions as translation unit mapping physical memory access requests from the device using knowledge of the guest physical address to host physical addresses translation rules.

Intel's solution for IOMMU is called Intel Virtualization Technology for Directed I/O (VT-d), and AMD's solution is called AMD-Vi. Check your motherboard datasheet for the appropriate technology. Even if your hardware does not have a IOMMU, certain PCI cards may work, such as serial PCI adapters, but the guest will show a warning on boot and the VM execution will terminate if the guest driver will attempt to enable card bus mastering.

It is very common that the BIOS or the host OS disables the IOMMU by default. So before any attempt to use it please make sure that the following apply:

- Your motherboard has an IOMMU unit.
- Your CPU supports the IOMMU.
- The IOMMU is enabled in the BIOS.
- The VM must run with VT-x/AMD-V and nested paging enabled.
- Your Linux kernel was compiled with IOMMU support, including DMA remapping. See the `CONFIG_DMAR` kernel compilation option. The PCI stub driver (`CONFIG_PCI_STUB`) is required as well.
- Your Linux kernel recognizes and uses the IOMMU unit. The `intel_iommu=on` boot option could be needed. Search for DMAR and PCI-DMA in kernel boot log.

Once you made sure that the host kernel supports the IOMMU, the next step is to select the PCI card and attach it to the guest. To figure out the list of available PCI devices, use the `lspci` command. The output will look as follows:

```
01:00.0 VGA compatible controller: ATI Technologies Inc Cedar PRO [Radeon HD 5450]
01:00.1 Audio device: ATI Technologies Inc Manhattan HDMI Audio [Mobility Radeon HD 5000 Series]
02:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit
        Ethernet controller (rev 03)
03:00.0 SATA controller: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
```

```
03:00.1 IDE interface: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
06:00.0 VGA compatible controller: nVidia Corporation G86 [GeForce 8500 GT] (rev a1)
```

The first column is a PCI address, in the format *bus:device.function*. This address could be used to identify the device for further operations. For example, to attach a PCI network controller on the system listed above to the second PCI bus in the guest, as device 5, function 0, use the following command:

```
$ VBoxManage modifyvm VM-name --pciattach 02:00.0@01:05.0
```

To detach the same device, use:

```
$ VBoxManage modifyvm VM-name --pcidetach 02:00.0
```

Please note that both host and guest could freely assign a different PCI address to the card attached during runtime, so those addresses only apply to the address of the card at the moment of attachment on the host, and during BIOS PCI init on the guest.

If the virtual machine has a PCI device attached, certain limitations apply:

- Only PCI cards with non-shared interrupts, such as those using MSI on the host, are supported at the moment.
- No guest state can be reliably saved or restored. The internal state of the PCI card cannot be retrieved.
- Teleportation, also called live migration, does not work. The internal state of the PCI card cannot be retrieved.
- No lazy physical memory allocation. The host will preallocate the whole RAM required for the VM on startup, as we cannot catch physical hardware accesses to the physical memory.

2.6. Webcam Passthrough

2.6.1. Using a Host Webcam in the Guest

Oracle VM VirtualBox includes a feature called *webcam passthrough*, which enables a guest to use a host webcam. This complements the general USB passthrough support which was the typical way of using host webcams in legacy releases. The webcam passthrough support can handle non-USB video sources in theory, but this is completely untested.

Note

The webcam passthrough module is shipped as part of the Oracle VM VirtualBox extension pack, which must be installed separately. See Installing Oracle VM VirtualBox and Extension Packs.

The host webcam can be attached to the VM using the **Devices** menu in the VM menu bar. The **Webcams** menu contains a list of available video input devices on the host. Clicking on a webcam name attaches or detaches the corresponding host device.

The `VBoxManage` command line tool can be used to enable webcam passthrough. Please see the host-specific sections below for additional details. The following commands are available:

- Get a list of host webcams, or other video input devices:

  ```
  $ VBoxManage list webcams
  ```

  The output format is as follows:

```
alias "user friendly name"
host path or identifier
```

The alias can be used as a shortcut in other commands. Alias '.0' means the default video input device on the host. Alias '.1', '.2'means first, second video input device, and so on. The device order is host-specific.

- Attach a webcam to a running VM, as follows:

```
VBoxManage controlvm VM name webcam attach [host_path|alias [settings]]
```

This attaches a USB webcam device to the guest.

The `settings` parameter is a string `Setting1=Value1;Setting2=Value2`, which enables you to configure the emulated webcam device. The following settings are supported:

- `MaxFramerate`: The highest rate at which video frames are sent to the guest. A higher frame rate requires more CPU power. Therefore sometimes it is useful to set a lower limit. Default is no limit and allow the guest to use all frame rates supported by the host webcam.
- `MaxPayloadTransferSize`: How many bytes the emulated webcam can send to the guest at a time. Default value is 3060 bytes, which is used by some webcams. Higher values can slightly reduce CPU load, if the guest is able to use larger buffers. However, a high `MaxPayloadTransferSize` might be not supported by some guests.

- Detach a webcam from a running VM, as follows:

```
VBoxManage controlvm VM-name webcam detach [host_path|alias]
```

- List the webcams attached to a running VM, as follows:

```
VBoxManage controlvm VM-name webcam list
```

The output contains the path or alias which was used in the `webcam attach` command for each attached webcam.

2.6.2. Windows Hosts

When the webcam device is detached from the host, the emulated webcam device is automatically detached from the guest.

2.6.3. Mac OS X Hosts

Mac OS X version 10.9 or later is required.

When the webcam device is detached from the host, the emulated webcam device remains attached to the guest and must be manually detached using the `VBoxManage controlvm VM-name webcam detach` command.

2.6.4. Linux and Oracle Solaris Hosts

When the webcam is detached from the host the emulated webcam device is automatically detached from the guest only if the webcam is streaming video. If the emulated webcam is inactive it should be manually detached using the `VBoxManage controlvm VM-name webcam detach` command.

Aliases `.0` and `.1` are mapped to `/dev/video0`, alias `.2` is mapped to `/dev/video1` and so forth.

2.7. Advanced Display Configuration

2.7.1. Custom VESA Resolutions

Apart from the standard VESA resolutions, the Oracle VM VirtualBox VESA BIOS enables you to add up to 16 custom video modes which will be reported to the guest operating system. When using Windows guests with the Oracle VM VirtualBox Guest Additions, a custom graphics driver will be used instead of the fallback VESA solution so this information does not apply.

Additional video modes can be configured for each VM using the extra data facility. The extra data key is called `CustomVideoMode`*x* with *x* being a number from 1 to 16. Please note that modes will be read from 1 until either the following number is not defined or 16 is reached. The following example adds a video mode that corresponds to the native display resolution of many notebook computers:

```
$ VBoxManage setextradata VM-name "CustomVideoMode1" "1400x1050x16"
```

The VESA mode IDs for custom video modes start at `0x160`. In order to use the above defined custom video mode, the following command line has to be supplied to Linux:

```
vga = 0x200 | 0x160
vga = 864
```

For guest operating systems with Oracle VM VirtualBox Guest Additions, a custom video mode can be set using the video mode hint feature.

2.7.2. Configuring the Maximum Resolution of Guests When Using the Graphical Frontend

When guest systems with the Guest Additions installed are started using the graphical frontend, the normal Oracle VM VirtualBox application, they will not be allowed to use screen resolutions greater than the host's screen size unless the user manually resizes them by dragging the window, switching to full screen or seamless mode or sending a video mode hint using `VBoxManage`. This behavior is what most users will want, but if you have different needs, you can change it by issuing one of the following commands from the command line:

- Remove all limits on guest resolutions.

  ```
  VBoxManage setextradata global GUI/MaxGuestResolution any
  ```

- Manually specify a maximum resolution.

  ```
  VBoxManage setextradata global GUI/MaxGuestResolution widthxheight
  ```

- Restore the default settings to all guest VMs.

  ```
  VBoxManage setextradata global GUI/MaxGuestResolution auto
  ```

2.8. Advanced Storage Configuration

2.8.1. Using a Raw Host Hard Disk From a Guest

As an alternative to using virtual disk images as described in Virtual Storage, Oracle VM VirtualBox can also present either entire physical hard disks or selected partitions as virtual disks to virtual machines.

With Oracle VM VirtualBox, this type of access is called *raw hard disk access*. It enables a guest operating system to access its virtual hard disk without going through the host OS file system. The actual

performance difference for image files compared to raw disk varies greatly depending on the overhead of the host file system, whether dynamically growing images are used, and on host OS caching strategies. The caching indirectly also affects other aspects such as failure behavior. For example, whether the virtual disk contains all data written before a host OS crash. Consult your host OS documentation for details on this.

Warning

Raw hard disk access is for expert users only. Incorrect use or use of an outdated configuration can lead to **total loss of data** on the physical disk. Most importantly, *do not* attempt to boot the partition with the currently running host operating system in a guest. This will lead to severe data corruption.

Raw hard disk access, both for entire disks and individual partitions, is implemented as part of the VMDK image format support. As a result, you will need to create a special VMDK image file which defines where the data will be stored. After creating such a special VMDK image, you can use it like a regular virtual disk image. For example, you can use the VirtualBox Manager, see The Virtual Media Manager, or `VBoxManage` to assign the image to a virtual machine.

2.8.1.1. Access to Entire Physical Hard Disk

While this variant is the simplest to set up, you must be aware that this will give a guest operating system direct and full access to an *entire physical disk*. If your *host* operating system is also booted from this disk, please take special care to not access the partition from the guest at all. On the positive side, the physical disk can be repartitioned in arbitrary ways without having to recreate the image file that gives access to the raw disk.

On a Linux host, to create an image that represents an entire physical hard disk which will not contain any actual data, as this will all be stored on the physical disk, use the following command:

```
$ VBoxManage internalcommands createrawvmdk -filename \
/path/to/file.vmdk -rawdisk /dev/sda
```

This creates the *`/path-to-file`*`.vmdk` file image that must be an absolute path. All data is read and written from `/dev/sda`.

On a Windows host, instead of the above device specification, for example use `\\.\PhysicalDrive0`. On a Mac OS X host, instead of the above device specification use for example `/dev/disk1`. Note that on Mac OS X you can only get access to an entire disk if no volume is mounted from it.

Creating the image requires read/write access for the given device. Read/write access is also later needed when using the image from a virtual machine. On some host platforms, such as Windows, raw disk access may be restricted and not permitted by the host OS in some situations.

Just like with regular disk images, this does not automatically attach the newly created image to a virtual machine. This can be done as follows:

```
$ VBoxManage storageattach WindowsXP --storagectl "IDE Controller" \
 --port 0 --device 0 --type hdd --medium /path/to/file.vmdk
```

When this is done the selected virtual machine will boot from the specified physical disk.

2.8.1.2. Access to Individual Physical Hard Disk Partitions

This *raw partition support* is quite similar to the full hard disk access described above. However, in this case, any partitioning information will be stored inside the VMDK image. This means that you can install a different boot loader in the virtual hard disk without affecting the host's partitioning information. While the guest will be able to *see* all partitions that exist on the physical disk, access will be filtered in that reading

from partitions for which no access is allowed the partitions will only yield zeroes, and all writes to them are ignored.

To create a special image for raw partition support, which will contain a small amount of data, on a Linux host, use the command:

```
$ VBoxManage internalcommands createrawvmdk -filename \
/path/to/file.vmdk -rawdisk /dev/sda -partitions 1,5
```

The command is identical to the one for full hard disk access, except for the additional `-partitions` parameter. This example would create the image */path-to-file*`.vmdk`, which must be absolute, and partitions 1 and 5 of `/dev/sda` would be made accessible to the guest.

Oracle VM VirtualBox uses the same partition numbering as your Linux host. As a result, the numbers given in the above example would refer to the first primary partition and the first logical drive in the extended partition, respectively.

On a Windows host, instead of the above device specification, use for example `\\.\PhysicalDrive0`. On a Mac OS X host, instead of the above device specification use `/dev/disk1`, for example. Note that on OS X you can only use partitions which are not mounted. Eject the respective volume first. Partition numbers are the same on Linux, Windows, and Mac OS X hosts.

The numbers for the list of partitions can be taken from the output of the following command:

```
$ VBoxManage internalcommands listpartitions -rawdisk /dev/sda
```

The output lists the partition types and sizes to give the user enough information to identify the partitions necessary for the guest.

Images which give access to individual partitions are specific to a particular host disk setup. You cannot transfer these images to another host. Also, whenever the host partitioning changes, the image *must be recreated*.

Creating the image requires read/write access for the given device. Read/write access is also later needed when using the image from a virtual machine. If this is not feasible, there is a special variant for raw partition access, currently only available on Linux hosts, that avoids having to give the current user access to the entire disk. To set up such an image, use:

```
$ VBoxManage internalcommands createrawvmdk -filename \
/path/to/file.vmdk -rawdisk /dev/sda -partitions 1,5 -relative
```

When used from a virtual machine, the image will then refer not to the entire disk, but only to the individual partitions. In this example, `/dev/sda1` and `/dev/sda5`. As a consequence, read/write access is only required for the affected partitions, not for the entire disk. During creation however, read-only access to the entire disk is required to obtain the partitioning information.

In some configurations it may be necessary to change the MBR code of the created image. For example, to replace the Linux boot loader that is used on the host by another boot loader. This enables for example the guest to boot directly to Windows, while the host boots Linux from the "same" disk. For this purpose the `-mbr` option is provided. It specifies a file name from which to take the MBR code. The partition table is not modified at all, so a MBR file from a system with totally different partitioning can be used. An example of this is:

```
$ VBoxManage internalcommands createrawvmdk -filename
/path/to/file.vmdk -rawdisk /dev/sda -partitions 1,5 -mbr winxp.mbr
```

The modified MBR will be stored inside the image, not on the host disk.

The created image can be attached to a storage controller in a VM configuration as usual.

2.8.2. Configuring the Hard Disk Vendor Product Data (VPD)

Oracle VM VirtualBox reports vendor product data for its virtual hard disks which consist of hard disk serial number, firmware revision and model number. These can be changed using the following commands:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/SerialNumber" "serial"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/FirmwareRevision" "firmware"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/ModelNumber" "model"
```

The serial number is a 20 byte alphanumeric string, the firmware revision an 8 byte alphanumeric string and the model number a 40 byte alphanumeric string. Instead of Port0, referring to the first port, specify the desired SATA hard disk port.

The above commands apply to virtual machines with an AHCI (SATA) controller. The commands for virtual machines with an IDE controller are:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/SerialNumber" "serial"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/FirmwareRevision" "firmware"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/ModelNumber" "model"
```

For hard disks, you can mark the drive as having a non-rotational medium by using the following command:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/NonRotational" "1"
```

Additional three parameters are needed for CD/DVD drives to report the vendor product data:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIVendorId" "vendor"
VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIProductId" "product"
VBoxManage setextradata VM-name \
"VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIRevision" "revision"
```

The vendor id is an 8 byte alphanumeric string, the product id an 16 byte alphanumeric string and the revision a 4 byte alphanumeric string. Instead of Port0, referring to the first port, specify the desired SATA hard disk port.

2.8.3. Access iSCSI Targets Using Internal Networking

As an experimental feature, Oracle VM VirtualBox enables access to an iSCSI target running in a virtual machine which is configured to use Internal Networking mode. See iSCSI Servers, Internal Networking, and VBoxManage storageattach.

The IP stack accessing Internal Networking must be configured in the virtual machine which accesses the iSCSI target. A free static IP and a MAC address not used by other virtual machines must be chosen. In the example below, adapt the name of the virtual machine, the MAC address, the IP configuration, and the Internal Networking name (MyIntNet) according to your needs. The following eight commands must first be issued:

```
$ VBoxManage setextradata VM-name \
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-name \
VBoxInternal/Devices/IntNetIP/0/Config/IP 10.0.9.1
$ VBoxManage setextradata VM-name \
VBoxInternal/Devices/IntNetIP/0/Config/Netmask 255.255.255.0
$ VBoxManage setextradata VM-name \
VBoxInternal/Devices/IntNetIP/0/LUN#0/Driver IntNet
$ VBoxManage setextradata VM-name \
VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/Network MyIntNet
$ VBoxManage setextradata VM-name \
VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/TrunkType 2
$ VBoxManage setextradata VM-name \
VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/IsService 1
```

Finally the iSCSI disk must be attached with the `--intnet` option to tell the iSCSI initiator to use internal networking, as follows:

```
$ VBoxManage storageattach ... --medium iscsi --server 10.0.9.30 \
--target iqn.2008-12.com.sun:sampletarget --intnet
```

Compared to a regular iSCSI setup, the IP address of the target *must* be specified as a numeric IP address, as there is no DNS resolver for internal networking.

The virtual machine with the iSCSI target should be started before the VM using it is powered on. If a virtual machine using an iSCSI disk is started without having the iSCSI target powered up, it can take up to 200 seconds to detect this situation. The VM will fail to power up.

2.9. Fine Tuning the Oracle VM VirtualBox NAT Engine

2.9.1. Configuring the Address of a NAT Network Interface

In NAT mode, the guest network interface is assigned to the IPv4 range `10.0.x.0/24` by default where *x* corresponds to the instance of the NAT interface +2. So *x* is 2 when there is only one NAT instance active. In that case the guest is assigned to the address `10.0.2.15`, the gateway is set to `10.0.2.2` and the name server can be found at `10.0.2.3`.

If the NAT network needs to be changed, use the following command:

```
$ VBoxManage modifyvm VM-name \
--natnet1 "192.168/16"
```

This command would reserve the network addresses from `192.168.0.0` to `192.168.254.254` for the first NAT network instance of *VM-name* The guest IP would be assigned to `192.168.0.15` and the default gateway could be found at `192.168.0.2`.

2.9.2. Configuring the Boot Server (Next Server) of a NAT Network Interface

For network booting in NAT mode, by default Oracle VM VirtualBox uses a built-in TFTP server at the IP address 10.0.2.4. This default behavior should work fine for typical remote-booting scenarios. However, it is possible to change the boot server IP and the location of the boot image with the following commands:

```
$ VBoxManage modifyvm VM-name \
--nattftpserver1 10.0.2.2
$ VBoxManage modifyvm VM-name \
--nattftpfile1 /srv/tftp/boot/MyPXEBoot.pxe
```

2.9.3. Tuning TCP/IP Buffers for NAT

The Oracle VM VirtualBox NAT stack performance is often determined by its interaction with the host's TCP/IP stack and the size of several buffers, `SO_RCVBUF` and `SO_SNDBUF`. For certain setups users

might want to adjust the buffer size for a better performance. This can by achieved using the following commands, where values are in kilobytes and can range from 8 to 1024:

```
$ VBoxManage modifyvm VM-name \
--natsettings1 16000,128,128,0,0
```

This example illustrates tuning the NAT settings. The first parameter is the MTU, then the size of the socket's send buffer and the size of the socket's receive buffer, the initial size of the TCP send window, and lastly the initial size of the TCP receive window. Note that specifying zero means fallback to the default value.

Each of these buffers has a default size of 64KB and default MTU is 1500.

2.9.4. Binding NAT Sockets to a Specific Interface

By default, Oracle VM VirtualBox's NAT engine will route TCP/IP packets through the default interface assigned by the host's TCP/IP stack. The technical reason for this is that the NAT engine uses sockets for communication. If you want to change this behavior, you can tell the NAT engine to bind to a particular IP address instead. For example, use the following command:

```
$ VBoxManage modifyvm VM-name \
--natbindip1 "10.45.0.2"
```

After this, all outgoing traffic will be sent through the interface with the IP address 10.45.0.2. Ensure that this interface is up and running before changing the NAT bind address.

2.9.5. Enabling DNS Proxy in NAT Mode

The NAT engine by default offers the same DNS servers to the guest that are configured on the host. In some scenarios, it can be desirable to hide the DNS server IPs from the guest, for example when this information can change on the host due to expiring DHCP leases. In this case, you can tell the NAT engine to act as DNS proxy using the following command:

```
$ VBoxManage modifyvm VM-name --natdnsproxy1 on
```

2.9.6. Using the Host's Resolver as a DNS Proxy in NAT Mode

For resolving network names, the DHCP server of the NAT engine offers a list of registered DNS servers of the host. If for some reason you need to hide this DNS server list and use the host's resolver settings, thereby forcing the Oracle VM VirtualBox NAT engine to intercept DNS requests and forward them to host's resolver, use the following command:

```
$ VBoxManage modifyvm VM-name --natdnshostresolver1 on
```

Note that this setting is similar to the DNS proxy mode, however whereas the proxy mode just forwards DNS requests to the appropriate servers, the resolver mode will interpret the DNS requests and use the host's DNS API to query the information and return it to the guest.

2.9.6.1. User-Defined Host Name Resolving

In some cases it might be useful to intercept the name resolving mechanism, providing a user-defined IP address on a particular DNS request. The intercepting mechanism enables the user to map not only a single host but domains and even more complex naming conventions if required.

The following command sets a rule for mapping a name to a specified IP:

```
VBoxManage setextradata VM-name \
"VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/AttachedDriver/Config/HostResolverMappings/ \
```

```
unique-rule-name-of-interception-rule/HostIP" IPv4

VBoxManage setextradata VM-name \
"VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/AttachedDriver/Config/HostResolverMappings/ \
unique-rule-name/HostName" hostname
```

The following command sets a rule for mapping a pattern name to a specified IP:

```
VBoxManage setextradata VM-name \
"VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/AttachedDriver/Config/HostResolverMappings/ \
unique-rule-name/HostIP" IPv4

VBoxManage setextradata VM-name \
"VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/AttachedDriver/Config/HostResolverMappings/ \
unique-rule-name/HostNamePattern" hostpattern
```

The host name pattern can include the following wildcard characters: pipe (|), question mark (?), and asterisk (*).

This example demonstrates how to instruct the host-resolver mechanism to resolve all domain and probably some mirrors of www.blocked-site.info site with IP 127.0.0.1:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/e1000/0/LUN#0/AttachedDriver/Config/HostResolverMappings/all_blocked_site/HostIP" 127
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/e1000/0/LUN#0/AttachedDriver/Config/HostResolverMappings/all_blocked_site/HostNamePa
```

The host resolver mechanism should be enabled to use user-defined mapping rules, otherwise they do not have any effect.

2.9.7. Configuring Aliasing of the NAT Engine

By default, the NAT core uses aliasing and uses random ports when generating an alias for a connection. This works well for the most protocols like SSH, FTP and so on. Though some protocols might need a more transparent behavior or may depend on the real port number the packet was sent from. You can change the NAT mode by using the following commands:

```
$ VBoxManage modifyvm VM-name \
--nataliasmode1 proxyonly
```

```
$ VBoxManage modifyvm "Linux Guest" --nataliasmode1 sameports
```

The first example disables aliasing and switches NAT into transparent mode, the second example enforces preserving of port values. These modes can be combined if necessary.

2.10. Configuring the BIOS DMI Information

The DMI data that Oracle VM VirtualBox provides to guests can be changed for a specific VM. Use the following commands to configure the DMI BIOS information. In case your VM is configured to use EFI firmware you need to replace `pcbios` by `efi` in the keys.

- DMI BIOS information (type 0)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVendor"        "BIOS Vendor"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVersion"       "BIOS Version"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseDate"   "BIOS Release Date"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMajor"  1
```

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMinor"  2
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMajor" 3
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMinor" 4
```

- DMI system information (type 1)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemVendor"      "System Vendor"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemProduct"     "System Product"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemVersion"     "System Version"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemSerial"      "System Serial"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemSKU"         "System SKU"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemFamily"      "System Family"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemUuid" \
"9852bf98-b83c-49db-a8de-182c42c7226b"
```

- DMI board information (type 2)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardVendor"       "Board Vendor"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardProduct"      "Board Product"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardVersion"      "Board Version"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardSerial"       "Board Serial"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardAssetTag"     "Board Tag"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardLocInChass"   "Board Location"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardBoardType"    10
```

- DMI system enclosure or chassis (type 3)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisVendor"     "Chassis Vendor"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisType"       3
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisVersion"    "Chassis Version"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisSerial"     "Chassis Serial"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisAssetTag"   "Chassis Tag"
```

- DMI processor information (type 4)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiProcManufacturer"  "GenuineIntel"
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiProcVersion"       "Pentium(R) III"
```

- DMI OEM strings (type 11)

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiOEMVBoxVer"        "vboxVer_1.2.3"
```

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiOEMVBoxRev"        "vboxRev_12345"
```

If a DMI string is not set, the default value of Oracle VM VirtualBox is used. To set an empty string use `"<EMPTY>"`.

Note that in the above list, all quoted parameters (DmiBIOSVendor, DmiBIOSVersion but not DmiBIOSReleaseMajor) are expected to be strings. If such a string is a valid number, the parameter is treated as number and the VM will most probably refuse to start with an `VERR_CFGM_NOT_STRING` error. In that case, use `"string:value"`. For example:

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/pcbios/0/Config/DmiSystemSerial"      "string:1234"
```

Changing this information can be necessary to provide the DMI information of the host to the guest to prevent Windows from asking for a new product key. On Linux hosts, the DMI BIOS information can be obtained with the following command:

```
$ dmidecode -t0
```

The DMI system information can be obtained as follows:

```
$ dmidecode -t1
```

2.11. Configuring Custom ACPI Tables

You can configure Oracle VM VirtualBox to present up to four custom ACPI tables to the guest. Use a command such as the following to configure custom ACPI tables. Note that `CustomTable1`, `CustomTable2`, and `CustomTable3` are available in addition to `CustomTable0`.

```
$ VBoxManage setextradata VM-name \
"VBoxInternal/Devices/acpi/0/Config/CustomTable0" "/path/to/table.bin"
```

Configuring custom ACPI tables can for example avoid the need for asking for a new product key on Windows Vista, Windows 7, Windows 8 and later guests. On Linux hosts, one of the system's ACPI tables can be read from `/sys/firmware/acpi/tables/`.

2.12. Fine Tuning Timers and Time Synchronization

2.12.1. Configuring the Guest Time Stamp Counter (TSC) to Reflect Guest Execution

By default, Oracle VM VirtualBox keeps all sources of time visible to the guest synchronized to a single time source, the monotonic host time. This reflects the assumptions of many guest operating systems, which expect all time sources to reflect "wall clock" time. In special circumstances it may be useful however to make the time stamp counter (TSC) in the guest reflect the time actually spent executing the guest.

This special TSC handling mode can be enabled on a per-VM basis, and for best results must be used only in combination with hardware virtualization. To enable this mode use the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/TM/TSCTiedToExecution" 1
```

To revert to the default TSC handling mode use:

```
$ VBoxManage setextradata VM-name "VBoxInternal/TM/TSCTiedToExecution"
```

Note that if you use the special TSC handling mode with a guest operating system which is very strict about the consistency of time sources you may get a warning or error message about the timing

inconsistency. It may also cause clocks to become unreliable with some guest operating systems depending on how they use the TSC.

2.12.2. Accelerate or Slow Down the Guest Clock

For certain purposes it can be useful to accelerate or to slow down the virtual guest clock. This can be achieved as follows:

```
$ VBoxManage setextradata VM-name "VBoxInternal/TM/WarpDrivePercentage" 200
```

The above example will double the speed of the guest clock while

```
$ VBoxManage setextradata VM-name "VBoxInternal/TM/WarpDrivePercentage" 50
```

will halve the speed of the guest clock. Note that changing the rate of the virtual clock can confuse the guest and can even lead to abnormal guest behavior. For instance, a higher clock rate means shorter timeouts for virtual devices with the result that a slightly increased response time of a virtual device due to an increased host load can cause guest failures. Note further that any time synchronization mechanism will frequently try to resynchronize the guest clock with the reference clock, which is the host clock if the Oracle VM VirtualBox Guest Additions are active. Therefore any time synchronization should be disabled if the rate of the guest clock is changed as described above. See Section 2.12.3, “Tuning the Guest Additions Time Synchronization Parameters”.

2.12.3. Tuning the Guest Additions Time Synchronization Parameters

The Oracle VM VirtualBox Guest Additions ensure that the guest's system time is synchronized with the host time. There are several parameters which can be tuned. The parameters can be set for a specific VM using the following command:

```
$ VBoxManage guestproperty set VM-name "/VirtualBox/GuestAdd/VBoxService/property" value
```

property is one of the following:

`--timesync-interval`	Specifies the interval at which to synchronize the time with the host. The default is 10000 ms (10 seconds).
`--timesync-min-adjust`	The minimum absolute drift value measured in milliseconds to make adjustments for. The default is 1000 ms on OS/2 and 100 ms elsewhere.
`--timesync-latency-factor`	The factor to multiply the time query latency with to calculate the dynamic minimum adjust time. The default is 8 times, which means as follows: Measure the time it takes to determine the host time, the guest has to contact the VM host service which may take some time. Multiply this value by 8 and do an adjustment only if the time difference between host and guest is bigger than this value. Do not do any time adjustment otherwise.
`--timesync-max-latency`	The max host timer query latency to accept. The default is 250 ms.
`--timesync-set-threshold`	The absolute drift threshold, given as milliseconds where to start setting the time instead of trying to smoothly adjust it. The default is 20 minutes.
`--timesync-set-start`	Set the time when starting the time sync service.

`--timesync-set-on-restore 0|1`	Set the time after the VM was restored from a saved state when passing 1 as parameter. This is the default. Disable by passing 0. In the latter case, the time will be adjusted smoothly, which can take a long time.

All these parameters can be specified as command line parameters to VBoxService as well.

2.12.4. Disabling the Guest Additions Time Synchronization

Once installed and started, the Oracle VM VirtualBox Guest Additions will try to synchronize the guest time with the host time. This can be prevented by forbidding the guest service from reading the host clock:

```
$ VBoxManage setextradata VM-name "VBoxInternal/Devices/VMMDev/0/Config/GetHostTimeDisabled" 1
```

2.13. Installing the Alternate Bridged Networking Driver on Oracle Solaris 11 Hosts

Oracle VM VirtualBox includes a network filter driver that utilizes Oracle Solaris 11's Crossbow functionality. By default, this new driver is installed for Oracle Solaris 11 hosts that have support for it.

To force installation of the older STREAMS based network filter driver, execute as root the following command before installing the Oracle VM VirtualBox package:

```
$ touch /etc/vboxinst_vboxflt
```

To force installation of the Crossbow based network filter driver, execute as root the following command before installing the Oracle VM VirtualBox package:

```
$ touch /etc/vboxinst_vboxbow
```

To check which driver is currently being used by Oracle VM VirtualBox, execute:

```
$ modinfo | grep vbox
```

If the output contains "vboxbow", it indicates Oracle VM VirtualBox is using the Crossbow network filter driver, while the name "vboxflt" indicates usage of the older STREAMS network filter.

2.14. Oracle VM VirtualBox VNIC Templates for VLANs on Oracle Solaris 11 Hosts

Oracle VM VirtualBox supports Virtual Network Interface (VNIC) templates for configuring VMs over VLANs. An Oracle VM VirtualBox VNIC template is a VNIC whose name starts with `vboxvnic_template`. The string is case-sensitive.

On Oracle Solaris 11 hosts, when Crossbow-based bridged networking is used, a VNIC template may be used to specify the VLAN ID to use while bridging over a network link.

The following is an example of how to use a VNIC template to configure a VM over a VLAN. Create an Oracle VM VirtualBox VNIC template, by executing as root:

```
# dladm create-vnic -t -l nge0 -v 23 vboxvnic_template0
```

This will create a temporary VNIC template over interface `nge0` with the VLAN ID 23. To create VNIC templates that are persistent across host reboots, skip the `-t` parameter in the above command. You may check the current state of links using the following command:

```
$ dladm show-link
LINK        CLASS     MTU    STATE    BRIDGE     OVER
```

```
nge0        phys      1500   up       --         --
nge1        phys      1500   down     --         --
vboxvnic_template0 vnic 1500 up       --         nge0

$ dladm show-vnic
LINK         OVER         SPEED  MACADDRESS        MACADDRTYPE         VID
vboxvnic_template0 nge0   1000   2:8:20:25:12:75   random              23
```

Once the VNIC template is created, any VMs that need to be on VLAN 23 over the interface `nge0` can be configured to bridge using this VNIC template.

VNIC templates makes managing VMs on VLANs simpler and efficient. The VLAN details are not stored as part of every VM's configuration but rather inherited from the VNIC template while starting the VM. The VNIC template itself can be modified anytime using the `dladm` command.

VNIC templates can be created with additional properties such as bandwidth limits and CPU fanout. Refer to your Oracle Solaris network documentation for details. The additional properties are also applied to VMs which bridge using the VNIC template.

2.15. Configuring Multiple Host-Only Network Interfaces on Oracle Solaris Hosts

By default Oracle VM VirtualBox provides you with one host-only network interface. Adding more host-only network interfaces on Oracle Solaris hosts requires manual configuration. Here is how to add another host-only network interface.

Begin by stopping all running VMs. Then, unplumb the existing "vboxnet0" interface by execute the following command as root:

```
# ifconfig vboxnet0 unplumb
```

If you have several vboxnet interfaces, you will need to unplumb all of them. Once all vboxnet interfaces are unplumbed, remove the driver by executing the following command as root:

```
# rem_drv vboxnet
```

Edit the file `/platform/i86pc/kernel/drv/vboxnet.conf` and add a line for the new interface we want to add as shown below:

```
name="vboxnet" parent="pseudo" instance=1;
name="vboxnet" parent="pseudo" instance=2;
```

Add as many of these lines as required with each line having a unique instance number.

Next, reload the vboxnet driver by executing the following command as root:

```
# add_drv vboxnet
```

On Oracle Solaris 11.1 and newer hosts you may want to rename the default vanity interface name. To check what name has been assigned, execute:

```
$ dladm show-phys
LINK              MEDIA               STATE      SPEED  DUPLEX    DEVICE
net0              Ethernet            up         100    full      e1000g0
net2              Ethernet            up         1000   full      vboxnet1
net1              Ethernet            up         1000   full      vboxnet0
```

In the above example, we can rename "net2" to "vboxnet1" before proceeding to plumb the interface. This can be done by executing as root:

```
# dladm rename-link net2 vboxnet1
```

Now plumb all the interfaces using `ifconfig vboxnetX plumb`, where *X* would be 1 in this case. Once the interface is plumbed, it may be configured like any other network interface. Refer to the `ifconfig` documentation for further details.

To make the settings for the newly added interfaces persistent across reboots, you will need to edit the files `/etc/inet/netmasks`, and if you are using NWAM `/etc/nwam/llp` and add the appropriate entries to set the netmask and static IP for each of those interfaces. The Oracle VM VirtualBox installer only updates these configuration files for the one "vboxnet0" interface it creates by default.

2.16. Configuring the Oracle VM VirtualBox CoreDumper on Oracle Solaris Hosts

Oracle VM VirtualBox is capable of producing its own core files for extensive debugging when things go wrong. Currently this is only available on Oracle Solaris hosts.

The Oracle VM VirtualBox CoreDumper can be enabled using the following command:

```
$ VBoxManage setextradata VM-name VBoxInternal2/CoreDumpEnabled 1
```

You can specify which directory to use for core dumps with this command, as follows:

```
$ VBoxManage setextradata VM-name VBoxInternal2/CoreDumpDir path-to-directory
```

Make sure the directory you specify is on a volume with sufficient free space and that the Oracle VM VirtualBox process has sufficient permissions to write files to this directory. If you skip this command and do not specify any core dump directory, the current directory of the Oracle VM VirtualBox executable will be used. This would most likely fail when writing cores as they are protected with root permissions. It is recommended you explicitly set a core dump directory.

You must specify when the Oracle VM VirtualBox CoreDumper should be triggered. This is done using the following commands:

```
$ VBoxManage setextradata VM-name VBoxInternal2/CoreDumpReplaceSystemDump 1
$ VBoxManage setextradata VM-name VBoxInternal2/CoreDumpLive 1
```

At least one of the above two commands will have to be provided if you have enabled the Oracle VM VirtualBox CoreDumper.

Setting `CoreDumpReplaceSystemDump` sets up the VM to override the host's core dumping mechanism and in the event of any crash only the Oracle VM VirtualBox CoreDumper would produce the core file.

Setting `CoreDumpLive` sets up the VM to produce cores whenever the VM process receives a `SIGUSR2` signal. After producing the core file, the VM will not be terminated and will continue to run. You can thus take cores of the VM process using the following command:

```
$ kill -s SIGUSR2 VM-process-id
```

The Oracle VM VirtualBox CoreDumper creates core files of the form `core.vb.process-name.process-ID` such as `core.vb.VBoxHeadless.11321`.

2.17. Oracle VM VirtualBox and Oracle Solaris Kernel Zones

Oracle Solaris kernel zones on x86-based systems make use of hardware-assisted virtualization features like Oracle VM VirtualBox does. However, for kernel zones and Oracle VM VirtualBox to share this hardware resource, they need to cooperate.

By default, due to performance reasons, Oracle VM VirtualBox acquires the hardware-assisted virtualization resource (VT-x/AMD-V) globally on the host machine and uses it until the last Oracle VM VirtualBox VM that requires it is powered off. This prevents other software from using VT-x/AMD-V during the time Oracle VM VirtualBox has taken control of it.

Oracle VM VirtualBox can be instructed to relinquish use of hardware-assisted virtualization features when not executing guest code, thereby allowing kernel zones to make use of them. To do this, shutdown all Oracle VM VirtualBox VMs and execute the following command:

```
$ VBoxManage setproperty hwvirtexclusive off
```

This command needs to be executed only once as the setting is stored as part of the global Oracle VM VirtualBox settings which will continue to persist across host-reboots and Oracle VM VirtualBox upgrades.

2.18. Locking Down the Oracle VM VirtualBox GUI

2.18.1. Customizing the VirtualBox Manager

There are several advanced customization settings for locking down the VirtualBox Manager. Locking down means removing some features that the user should not see.

```
VBoxManage setextradata global GUI/Customizations property[,property ...]
```

property is one of the following properties:

noSelector	Do not allow users to start the VirtualBox Manager. Trying to do so will show a window containing a proper error message.
noMenuBar	VM windows will not contain a menu bar.
noStatusBar	VM windows will not contain a status bar.

To disable any of these VirtualBox Manager customizations use the following command:

```
$ VBoxManage setextradata global GUI/Customizations
```

2.18.2. VM Selector Customization

The following per-machine VM extradata settings can be used to change the behavior of the VM selector window in respect of certain VMs:

```
$ VBoxManage setextradata VM-name property true
```

property can be any of the following:

GUI/HideDetails	Do not show the VM configuration of a certain VM. The details window will remain just empty if this VM is selected.
GUI/PreventReconfiguration	Do not allow the user to open the **Settings** dialog for a certain VM.
GUI/PreventSnapshotOperations	Prevent snapshot operations for a VM from the GUI, either at runtime or when the VM is powered off.
GUI/HideFromManager	Hide a certain VM in the VM selector window.
GUI/PreventApplicationUpdate	Disable the automatic update check and hide the corresponding menu item.

Note that these settings do not prevent the user from reconfiguring the VM by using the `VBoxManage modifyvm` command.

2.18.3. Configure VM Selector Menu Entries

You can disable, or blacklist, certain entries in the global settings page of the VM selector:

```
$ VBoxManage setextradata global GUI/RestrictedGlobalSettingsPages property[,property...]
```

property is one of the following:

`General`	Do not show the **General** settings pane.
`Input`	Do not show the **Input** settings pane.
`Update`	Do not show the **Update** settings pane.
`Language`	Do not show the **Language** settings pane.
`Display`	Do not show the **Display** settings pane.
`Network`	Do not show the **Network** settings pane.
`Extensions`	Do not show the **Extensions** settings pane.
`Proxy`	Do not show the **Proxy** settings pane.

This is a global setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata global GUI/RestrictedGlobalSettingsPages
```

2.18.4. Configure VM Window Menu Entries

You can disable, or blacklist, certain menu actions in the VM window:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeMenus property[,property...]
```

property is one of the following:

`All`	Do not show any menu in the VM window.
`Machine`	Do not show the **Machine** menu in the VM window.
`View`	Do not show the **View** menu in the VM window.
`Devices`	Do not show the **Devices** menu in the VM window.
`Help`	Do not show the **Help** menu in the VM window.
`Debug`	Do not show the **Debug** menu in the VM window. The Debug menu is only visible if the GUI was started with special command line parameters or environment variable settings.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeMenus
```

You can also disable, or blacklist, certain menu actions of certain menus. Use the following command to disable certain actions of the **Application** menu. This is only available on Mac OS X hosts.

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeApplicationMenuActions property[,property...]
```

property is one of the following:

`All`	Do not show any menu item in this menu.
`About`	Do not show the **About** menu item in this menu.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeMenus
```

Use the following command to disable certain actions of the **Machine** menu:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeApplicationMenuActions property[,property ...]
```

property is one of the following:

`All`	Do not show any menu item in this menu.
`SettingsDialog`	Do not show the **Settings** menu item in this menu.
`TakeSnapshot`	Do not show the **Take Snapshot** menu item in this menu.
`TakeScreenshot`	Do not show the **Take Screenshot** menu item in this menu.
`InformationDialog`	Do not show the **Session Information** menu item in this menu.
`MouseIntegration`	Do not show the **Disable Mouse Integration** menu item in this menu.
`TypeCAD`	Do not show the **Insert Ctrl+Alt+Del** menu item in this menu.
`TypeCABS`	Do not show the **Insert Ctrl+Alt+Backspace** menu item in this menu. Available on X11 hosts only.
`Pause`	Do not show the **Pause** menu item in this menu.
`Reset`	Do not show the **Reset** menu item in this menu.
`SaveState`	Do not show the **Save the machine state** menu item in this menu.
`Shutdown`	Do not show the **ACPI Shutdown** menu item in this menu.
`PowerOff`	Do not show the **Power Off the machine** menu item in this menu.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeApplicationMenuActions
```

Use the following command to disable certain actions of the **View** menu:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeViewMenuActions property[,property...]
```

property is one of the following:

`All`	Do not show any menu item in this menu.
`Fullscreen`	Do not show the **Switch to Fullscreen** menu item in this menu.
`Seamless`	Do not show the **Switch to Seamless Mode** menu item in this menu.

`Scale`	Do not show the **Switch to Scaled Mode** menu item in this menu.
`GuestAutoresize`	Do not show the **Auto-resize Guest Display** menu item in this menu.
`AdjustWindow`	Do not show the **Adjust Window Size** menu item in this menu.
`Multiscreen`	Do not show the **Multiscreen** menu item in this menu. Only visible in full screen/seamless mode.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeViewMenuActions
```

Use the following command to disable certain actions of the **View** menu:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeDevicesMenuActions property[,property...]
```

property is one of the following properties to disable actions in the **Devices** menu:

`All`	Do not show any menu item in this menu.
`OpticalDevices`	Do not show the **CD/DVD Devices** menu item in this menu.
`FloppyDevices`	Do not show the **Floppy Devices** menu item in this menu.
`USBDevices`	Do not show the **USB Devices** menu item in this menu.
`SharedClipboard`	Do not show the **Shared Clipboard** menu item in this menu.
`DragAndDrop`	Do not show the **Drag and Drop** menu item in this menu.
`NetworkSettings`	Do not show the **Network Settings...** menu item in this menu.
`SharedFoldersSettings`	Do not show the **Shared Folders Settings...** menu item in this menu.
`VRDEServer`	Do not show the **Remove Display** menu item in this menu.
`InstallGuestTools`	Do not show the **Insert Guest Additions CD image...** menu item in this menu.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeDevicesMenuActions
```

Use the following command to disable certain actions of the **View** menu:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeDebuggerMenuActions property[,property...]
```

property is one of the following properties to disable actions in the *Debug* menu, which is completely disabled by default:

`All`	Do not show any menu item in this menu.
`Statistics`	Do not show the **Statistics...** menu item in this menu.
`CommandLine`	Do not show the **Command Line...** menu item in this menu.
`Logging`	Do not show the **Logging...** menu item in this menu.
`LogDialog`	Do not show the **Show Log...** menu item in this menu.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeDebuggerMenuActions
```

Use the following command to disable certain actions of the **View** menu:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeHelpMenuActions property[,property...]
```

property is one of the following properties to disable actions in the **Help** menu, which is completely disabled by default:

`All`	Do not show any menu item in this menu.
`Contents`	Do not show the **Contents...** menu item in this menu.
`WebSite`	Do not show the **VirtualBox Web Site...** menu item in this menu.
`ResetWarnings`	Do not show the **Reset All Warnings** menu item in this menu.
`NetworkAccessManager`	Do not show the **Network Operations Manager** menu item in this menu.
`About`	Do not show the **About** menu item in this menu. Only for non-Mac OS X hosts.
`Contents`	Do not show the **Contents...** menu item in this menu.
`Contents`	Do not show the **Contents...** menu item in this menu.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedRuntimeHelpMenuActions
```

2.18.5. Configure VM Window Status Bar Entries

You can disable, or blacklist, certain status bar items:

```
$ VBoxManage setextradata VM-name GUI/RestrictedStatusBarIndicators property[,property...]
```

property is one of the following:

`HardDisks`	Do not show the hard disk icon in the VM window status bar. By default the hard disk icon is only shown if the VM configuration contains one or more hard disks.
`OpticalDisks`	Do not show the CD icon in the VM window status bar. By default the CD icon is only shown if the VM configuration contains one or more CD drives.
`FloppyDisks`	Do not show the floppy icon in the VM window status bar. By default the floppy icon is only shown if the VM configuration contains one or more floppy drives.
`Network`	Do not show the network icon in the VM window status bar. By default the network icon is only shown if the VM configuration contains one or more active network adapters.
`USB`	Do not show the USB icon in the status bar.

`SharedFolders`	Do not show the shared folders icon in the status bar.
`Capture`	Do not show the capture icon in the status bar.
`Features`	Do not show the CPU features icon in the status bar.
`Mouse`	Do not show the mouse icon in the status bar.
`Keyboard`	Do not show the keyboard icon in the status bar.

This is a per-VM setting. You can specify any combination of properties. If all options are specified, no icons are shown in the status bar of the VM window. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedStatusBarIndicators
```

2.18.6. Configure VM Window Visual Modes

You can disable, or blacklist, certain VM visual modes:

```
$ VBoxManage setextradata VM-name GUI/RestrictedVisualStates property[,property...]
```

property is one of the following:

`Fullscreen`	Do not allow to switch the VM into full screen mode.
`Seamless`	Do not allow to switch the VM into seamless mode.
`Scale`	Do not allow to switch the VM into scale mode.

This is a per-VM setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedVisualStates
```

2.18.7. Host Key Customization

To disable all Host key combinations, open the preferences and change the Host key to None. This might be useful when using Oracle VM VirtualBox in a kiosk mode.

To redefine or disable certain Host key actions, use the following command:

```
$ VBoxManage setextradata global GUI/Input/MachineShortcuts "FullscreenMode=F,...."
```

The following table shows the possible Host key actions, together with their default Host key shortcut. Setting an action to None will disable that Host key action.

Table 2.1 Host Key Customization

Action	Default Key	Action
`TakeSnapshot`	T	Take a snapshot
`TakeScreenshot`	E	Take a screenshot
`MouseIntegration`	I	Toggle mouse integration
`TypeCAD`	Del	Inject Ctrl+Alt+Del
`TypeCABS`	Backspace	Inject Ctrl+Alt+Backspace
`Pause`	P	Pause the VM

Action	Default Key	Action
Reset	R	Hard reset the guest
SaveState		Save the VM state and terminate the VM
Shutdown	H	Press the virtual ACPI power button
PowerOff		Power off the VM without saving the state
Close	Q	Show the Close VM dialog
FullscreenMode	F	Switch the VM into full screen mode
SeamlessMode	L	Switch the VM into seamless mode
ScaleMode	C	Switch the VM into scaled mode
GuestAutoResize	G	Automatically resize the guest window
WindowAdjust	A	Immediately resize the guest window
PopupMenu	Home	Show the popup menu in full screen mode and seamless mode
SettingsDialog	S	Open the VM Settings dialog
InformationDialog	N	Show the VM Session Information window
NetworkAdaptersDialog		Show the VM Network Adapters dialog
SharedFoldersDialog		Show the VM Shared Folders dialog
InstallGuestAdditions	D	Mount the ISO containing the Guest Additions

To disable full screen mode and seamless mode, use the following command:

```
$ VBoxManage setextradata global GUI/Input/MachineShortcuts "FullscreenMode=None,SeamlessMode=None"
```

2.18.8. Action when Terminating the VM

You can disallow, or blacklist, certain actions when terminating a VM. To disallow specific actions, use the following command:

```
$ VBoxManage setextradata VM-name GUI/RestrictedCloseActions property[,property...]
```

property is one of the following:

SaveState — Do not allow the user to save the VM state when terminating the VM.

Shutdown — Do not allow the user to shutdown the VM by sending the ACPI power-off event to the guest.

PowerOff — Do not allow the user to power off the VM.

`PowerOffRestoringSnapshot`	Do not allow the user to return to the last snapshot when powering off the VM.
`Detach`	Do not allow the user to detach from the VM process if the VM was started in separate mode.

This is a per-VM setting. You can specify any combination of properties. If all properties are specified, the VM cannot be shut down.

2.18.9. Default Action when Terminating the VM

You can define a specific action for terminating a VM. In contrast to the setting decribed in the previous section, this setting allows only one action when the user terminates the VM. No exit menu is shown. Use the following command:

```
$ VBoxManage setextradata VM-name GUI/DefaultCloseAction action
```

`action` is one of the following:

`SaveState`	Save the VM state before terminating the VM process.
`Shutdown`	The VM is shut down by sending the ACPI power-off event to the guest.
`PowerOff`	The VM is powered off.
`PowerOffRestoringSnapshot`	The VM is powered off and the saved state returns to the last snapshot.
`Detach`	Terminate the frontend but leave the VM process running.

This is a per-VM setting. You can specify any combination of properties. If all properties are specified, the VM cannot be shut down.

2.18.10. Action for Handling a Guru Meditation

A VM runs into a Guru Meditation if there is a problem which cannot be fixed by other means than terminating the process. The default is to show a message window which instructs the user to open a bug report.

This behavior can be configured as follows:

```
$ VBoxManage setextradata VM-name GUI/GuruMeditationHandler mode
```

`mode` is one of the following:

`Default`	A message window is shown. After the user confirmed, the VM is terminated.
`PowerOff`	The VM is immediately powered-off without showing any message window. The VM logfile will show information about what happened.
`Ignore`	The VM is left in stuck mode. Execution is stopped but no message window is shown. The VM has to be powered off manually.

This is a per-VM setting.

2.18.11. Configuring Automatic Mouse Capturing

By default, the mouse is captured if the user clicks on the guest window and the guest expects relative mouse coordinates at this time. This happens if the pointing device is configured as PS/2 mouse and the

guest has not yet started the Oracle VM VirtualBox Guest Additions. For instance, the guest is booting or the Guest Additions are not installed, or if the pointing device is configured as a USB tablet but the guest has no USB driver loaded yet. Once the Guest Additions become active or the USB guest driver is started, the mouse capture is automatically released.

The default behavior is sometimes not desired. Therefore it can be configured as follows:

```
VBoxManage setextradata VM-name GUI/MouseCapturePolicy mode
```

mode is one of the following:

`Default`	The default behavior as described above.
`HostComboOnly`	The mouse is only captured if the Host Key is toggled.
`Disabled`	The mouse is never captured, also not by toggling the Host Key

This is a per-VM setting.

2.18.12. Requesting Legacy Full-Screen Mode

Oracle VM VirtualBox uses special window manager facilities to switch a multi-screen machine to full-screen on a multi-monitor host system. However, not all window managers provide these facilities correctly. Oracle VM VirtualBox can be configured to use a legacy method of switching to full-screen mode instead, by using the command:

```
VBoxManage setextradata global GUI/Fullscreen/LegacyMode true
```

You can go back to the default method by using the following command:

```
VBoxManage setextradata global GUI/Fullscreen/LegacyMode
```

This is a global setting.

2.18.13. Removing Certain Modes of Networking From the GUI

It is possible to remove networking modes from Oracle VM VirtualBox GUI. To do this, use the following command:

```
VBoxManage setextradata global GUI/RestrictedNetworkAttachmentTypes property[,property...]
```

property is one of the following:

`NAT`	Remove the **NAT** option from the GUI.
`NATNetwork`	Remove the **NAT network** option from the GUI.
`BridgedAdapter`	Remove the **Bridged networking** option from the GUI.
`InternalNetwork`	Remove the **Internal networking** option from the GUI.
`HostOnlyAdapter`	Remove the **Host Only networking** option from the GUI.
`GenericDriver`	Remove the **Generic networking** option from the GUI.

This is a global setting. You can specify any combination of properties. To restore the default behavior, use the following command:

```
VBoxManage setextradata global GUI/RestrictedNetworkAttachmentTypes
```

2.19. Starting the Oracle VM VirtualBox Web Service Automatically

The Oracle VM VirtualBox web service, `vboxwebsrv`, is used for controlling Oracle VM VirtualBox remotely. It is documented in detail in the Oracle VM VirtualBox Software Development Kit (SDK). See Chapter 4, *Oracle VM VirtualBox Programming Interfaces*. Web service start scripts are available for supported host operating systems. The following sections describe how to use the scripts. The Oracle VM VirtualBox web service is never started automatically as a result of a standard installation.

2.19.1. Linux: Starting the Web Service With init

On Linux, the web service can be automatically started during host boot by adding appropriate parameters to the file `/etc/default/virtualbox`. There is one mandatory parameter, `VBOXWEB_USER`, which must be set to the user which will later start the VMs. The parameters in the following table all start with the `VBOXWEB_` prefix string. For example: `VBOXWEB_HOST` and `VBOXWEB_PORT`.

Table 2.2 Web Service Configuration Parameters

Parameter	Description	Default
`USER`	The user which the web service runs as	
`HOST`	The host to bind the web service to	localhost
`PORT`	The port to bind the web service to	18083
`SSL_KEYFILE`	Server key and certificate file, in PEM format	
`SSL_PASSWORDFILE`	File name for password to server key	
`SSL_CACERT`	CA certificate file, in PEM format	
`SSL_CAPATH`	CA certificate path	
`SSL_DHFILE`	DH file name or DH key length in bits	
`SSL_RANDFILE`	File containing seed for random number generator	
`TIMEOUT`	Session timeout in seconds, 0 disables timeouts	300
`CHECK_INTERVAL`	Frequency of timeout checks in seconds	5
`THREADS`	Maximum number of worker threads to run in parallel	100
`KEEPALIVE`	Maximum number of requests before a socket will be closed	100
`ROTATE`	Number of log files, 0 disables log rotation	10
`LOGSIZE`	Maximum log file size to trigger rotation, in bytes	1MB
`LOGINTERVAL`	Maximum time interval to trigger log rotation, in seconds	1 day

Setting the parameter `SSL_KEYFILE` enables the SSL/TLS support. Using encryption is strongly encouraged, as otherwise everything, including passwords, is transferred in clear text.

2.19.2. Oracle Solaris: Starting the Web Service With SMF

On Oracle Solaris hosts, the Oracle VM VirtualBox web service daemon is integrated into the SMF framework. You can change the parameters, but do not have to if the defaults below already match your needs:

```
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
```

The table in Section 2.19.1, “Linux: Starting the Web Service With init” showing the parameter names and defaults also applies for Oracle Solaris. The parameter names must be changed to lowercase and a prefix of `config/` has to be added. For example: `config/user` or `config/ssl_keyfile`. If you make any change, do not forget to run the following command to put the changes into effect immediately:

```
svcadm refresh svc:/application/virtualbox/webservice:default
```

If you forget the above command then the previous settings are used when enabling the service. Check the current property settings as follows:

```
svcprop -p config svc:/application/virtualbox/webservice:default
```

When everything is configured correctly you can start the Oracle VM VirtualBox web service with the following command:

```
svcadm enable svc:/application/virtualbox/webservice:default
```

For more information about SMF, please refer to the Oracle Solaris documentation.

2.19.3. Mac OS X: Starting the Web Service With launchd

On Mac OS X, launchd is used to start the Oracle VM VirtualBox webservice. An example configuration file can be found in `$HOME/Library/LaunchAgents/org.virtualbox.vboxwebsrv.plist`. It can be enabled by changing the `Disabled` key from `true` to `false`. To manually start the service use the following command:

```
launchctl load ~/Library/LaunchAgents/org.virtualbox.vboxwebsrv.plist
```

For additional information on how launchd services could be configured see:

https://developer.apple.com/library/mac/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html.

2.20. Oracle VM VirtualBox Watchdog

The memory ballooning service, formerly known as `VBoxBalloonCtrl`, was renamed to VBoxWatchdog. This service now incorporates the following host services that are meant to be run in a server environment:

- **Memory ballooning control.** This service automatically takes care of a VM's configured memory balloon. See Memory Ballooning. This service is useful for server environments where VMs may dynamically require more or less memory during runtime.

  The service periodically checks a VM's current memory balloon and its free guest RAM and automatically adjusts the current memory balloon by inflating or deflating it accordingly. This handling only applies to running VMs having recent Guest Additions installed.

- **Host isolation detection.** This service provides a way to detect whether the host cannot reach the specific Oracle VM VirtualBox server instance anymore and take appropriate actions, such as shutting down, saving the current state or even powering down certain VMs.

All configuration values can be either specified using the command line or global extradata, whereas command line values always have a higher priority when set. Some of the configuration values also be specified on a per-VM basis. So the overall lookup order is: command line, per-VM basis extradata if available, global extradata.

2.20.1. Memory Ballooning Control

The memory ballooning control inflates and deflates the memory balloon of VMs based on the VMs free memory and the desired maximum balloon size.

To set up the memory ballooning control the maximum ballooning size a VM can reach needs to be set. This can be specified using the command line, as follows:

```
--balloon-max <Size in MB>
```

Using a per-VM basis extradata value, as follows:

```
VBoxManage setextradata <VM-Name> VBoxInternal2/Watchdog/BalloonCtrl/BalloonSizeMax <Size in MB>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonSizeMax <Size in MB>
```

Note

If no maximum ballooning size is specified by at least one of the parameters above, no ballooning will be performed at all.

Setting the ballooning increment in MB can be either done using command line, as follows:

```
--balloon-inc <Size in MB>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonIncrementMB <Size in MB>
```

The default ballooning increment is 256 MB if not specified.

The same options apply for a ballooning decrement. Using the command line, as follows:

```
--balloon-dec <Size in MB>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonDecrementMB <Size in MB>
```

The default ballooning decrement is 128 MB if not specified.

The lower limit in MB for a balloon can be defined using the command line, as follows:

```
--balloon-lower-limit <Size in MB>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonLowerLimitMB <Size in MB>
```

The default lower limit is 128 MB if not specified.

2.20.2. Host Isolation Detection

To detect whether a host is being isolated, that is, the host cannot reach the Oracle VM VirtualBox server instance anymore, the host needs to set an alternating value to a global extradata value within a time period. If this value is not set within that time period a timeout occurred and the so-called host isolation response will be performed to the VMs handled. Which VMs are handled can be controlled by defining VM groups and assigning VMs to those groups. By default no groups are set, meaning that all VMs on the server will be handled when no host response is received within 30 seconds.

Set the groups handled by the host isolation detection using the following command line:

```
--apimon-groups=<string[,stringN]>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/Groups <string[,stringN]>
```

Set the host isolation timeout using the following command line:

```
--apimon-isln-timeout=<ms>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/IsolationTimeoutMS <ms>
```

Set the actual host isolation response using the following command line:

```
--apimon-isln-response=<cmd>
```

Using a global extradata value, as follows:

```
VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/IsolationResponse <cmd>
```

The following response commands are available:

- `none`. This has no effect.
- `pause`. Pauses the execution of a VM.
- `poweroff`. Shuts down the VM by pressing the virtual power button. The VM will not have the chance of saving any data or veto the shutdown process.
- `save`. Saves the current machine state and powers off the VM afterwards. If saving the machine state fails the VM will be paused.
- `shutdown`. Shuts down the VM in a gentle way by sending an `ACPI` shutdown event to the VM's operating system. The OS then has the chance of doing a clean shutdown.

2.20.3. More Information

For more advanced options and parameters like verbose logging check the built-in command line help accessible with `--help`.

2.20.4. Linux: Starting the Watchdog Service With init

On Linux, the watchdog service can be automatically started during host boot by adding appropriate parameters to the file `/etc/default/virtualbox`. There is one mandatory parameter,

VBOXWATCHDOG_USER, which must be set to the user which will later start the VMs. For backward compatibility you can also specify VBOXBALLOONCTRL_USER.

The parameters in the following table all start with the VBOXWATCHDOG_ prefix string. For example: VBOXWATCHDOG_BALLOON_INTERVAL and VBOXWATCHDOG_LOGSIZE. Legacy parameters such as VBOXBALLOONCTRL_INTERVAL can still be used.

Table 2.3 Oracle VM VirtualBox Watchdog Configuration Parameters

Parameter	Description	Default
USER	The user which the watchdog service runs as	
ROTATE	Number of log files, 0 disables log rotation	10
LOGSIZE	Maximum log file size to trigger rotation, in bytes	1MB
LOGINTERVAL	Maximum time interval to trigger log rotation, in seconds	1 day
BALLOON_INTERVAL	Interval for checking the balloon size, in milliseconds	30000
BALLOON_INCREMENT	Balloon size increment, in megabytes	256
BALLOON_DECREMENT	Balloon size decrement, in megabytes	128
BALLOON_LOWERLIMIT	Balloon size lower limit, in megabytes	64
BALLOON_SAFETYMARGIN	Free memory required for decreasing the balloon size, in megabytes	1024

2.20.5. Oracle Solaris: Starting the Watchdog Service With SMF

On Oracle Solaris hosts, the Oracle VM VirtualBox watchdog service daemon is integrated into the SMF framework. You can change the parameters, but do not have to if the defaults already match your needs:

```
svccfg -s svc:/application/virtualbox/balloonctrl:default setprop \
  config/balloon_interval=10000
svccfg -s svc:/application/virtualbox/balloonctrl:default setprop \
config/balloon_safetymargin=134217728
```

Table 2.3, “Oracle VM VirtualBox Watchdog Configuration Parameters” also applies for Oracle Solaris. The parameter names must be changed to lowercase and a prefix of config/ has to be added. For example: config/user or config/balloon_safetymargin. If you made any change, do not forget to run the following command to put the changes into effect immediately:

```
svcadm refresh svc:/application/virtualbox/balloonctrl:default
```

If you forget the above command then the previous settings will be used when enabling the service. Check the current property settings with the following command:

```
svcprop -p config svc:/application/virtualbox/balloonctrl:default
```

When everything is configured correctly you can start the Oracle VM VirtualBox watchdog service with the following command:

```
svcadm enable svc:/application/virtualbox/balloonctrl:default
```

For more information about SMF, please refer to the Oracle Solaris documentation.

2.21. Other Extension Packs

Another extension pack called VNC is available. This extension pack is open source and replaces the previous integration of the VNC remote access protocol. This is experimental code, and is initially available in the Oracle VM VirtualBox source code package only. It is to a large portion code contributed by users, and is not supported in any way by Oracle.

The keyboard handling is severely limited, and only the US keyboard layout works. Other keyboard layouts will have at least some keys which produce the wrong results, often with quite surprising effects, and for layouts which have significant differences to the US keyboard layout it is most likely unusable.

It is possible to install both the Oracle VM VirtualBox Extension Pack and VNC, but only one VRDE module can be active at any time. The following command switches to the VNC VRDE module in VNC:

```
VBoxManage setproperty vrdeextpack VNC
```

Configuring the remote access works very similarly to VRDP, see Section 1.1, “Remote Display (VRDP Support)”, with some limitations. VNC does not support specifying several port numbers, and the authentication is done differently. VNC can only deal with password authentication, and there is no option to use password hashes. This leaves no other choice than having a clear-text password in the VM configuration, which can be set with the following command:

```
VBoxManage modifyvm VM-name --vrdeproperty VNCPassword=secret
```

The user is responsible for keeping this password secret, and it should be removed when a VM configuration is passed to another person, for whatever purpose. Some VNC servers claim to have encrypted passwords in the configuration. This is not true encryption, it is only concealing the passwords, which is only as secure as using clear-text passwords.

The following command switches back to VRDP, if installed:

```
VBoxManage setproperty vrdeextpack "Oracle VM VirtualBox Extension Pack"
```

2.22. Starting Virtual Machines During System Boot

You can start VMs automatically during system boot on Linux, Oracle Solaris, and Mac OS X platforms for all users.

2.22.1. Linux: Starting the Autostart Service With init

On Linux, the autostart service is activated by setting two variables in `/etc/default/virtualbox`. The first one is `VBOXAUTOSTART_DB` which contains an absolute path to the autostart database directory. The directory should have write access for every user who should be able to start virtual machines automatically. Furthermore the directory should have the sticky bit set. The second variable is `VBOXAUTOSTART_CONFIG` which points the service to the autostart configuration file which is used during boot to determine whether to allow individual users to start a VM automatically and configure startup delays. The configuration file can be placed in `/etc/vbox` and contains several options. One is `default_policy` which controls whether the autostart service allows or denies to start a VM for users which are not in the exception list. The exception list starts with `exception_list` and contains a comma separated list with usernames. Furthermore a separate startup delay can be configured for every user to avoid overloading the host. A sample configuration is given below:

```
# Default policy is to deny starting a VM, the other option is "allow".
default_policy = deny

# Bob is allowed to start virtual machines but starting them
# will be delayed for 10 seconds
bob = {
    allow = true
    startup_delay = 10
}

# Alice is not allowed to start virtual machines, useful to exclude certain users
# if the default policy is set to allow.
alice = {
    allow = false
}
```

Any user who wants to enable autostart for individual machines must set the path to the autostart database directory with the following command:

```
VBoxManage setproperty autostartdbpath autostart-directory
```

2.22.2. Oracle Solaris: Starting the Autostart Service With SMF

On Oracle Solaris hosts, the Oracle VM VirtualBox autostart daemon is integrated into the SMF framework. To enable it you must point the service to an existing configuration file which has the same format as on Linux, see Section 2.22.1, “Linux: Starting the Autostart Service With init”. For example:

```
# svccfg -s svc:/application/virtualbox/autostart:default setprop \
  config/config=/etc/vbox/autostart.cfg
```

When everything is configured correctly you can start the Oracle VM VirtualBox autostart service with the following command:

```
# svcadm enable svc:/application/virtualbox/autostart:default
```

For more information about SMF, see the Oracle Solaris documentation.

2.22.3. Mac OS X: Starting the Autostart Service With launchd

On Mac OS X, launchd is used to start the Oracle VM VirtualBox autostart service. An example configuration file can be found in `/Applications/VirtualBox.app/Contents/MacOS/org.virtualbox.vboxautostart.plist`. To enable the service copy the file to `/Library/LaunchDaemons` and change the `Disabled` key from `true` to `false`. Furthermore replace the second parameter to an existing configuration file which has the same format as on Linux, see Section 2.22.1, “Linux: Starting the Autostart Service With init”.

To manually start the service use the following command:

```
# launchctl load /Library/LaunchDaemons/org.virtualbox.vboxautostart.plist
```

For additional information on how launchd services can be configured see:

http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPSystemStartup/BPSystemStartup.html.

2.22.4. Windows: Starting the Autostart Service With a Windows service

On Windows, autostarting is implemented as a Windows service. The service is installed for every user with their own credentials. Before installing any autostart services on a system you first have to define

the VBOXAUTOSTART_CONFIG environment variable in the system variables with the path to the autostart configuration file. The configuration file has the same format as on Linux, see Section 2.22.1, “Linux: Starting the Autostart Service With init”, except the user name can be specified using the following formats: "user", "domain\user", ".\user" and "user@domain".

To enable autostarting for a particular user, a member of the administrators group must run the following command:

```
VBoxAutostartSvc install --user=<user> [--password-file=<password_file>]
```

The password file should contain the password followed by a line break. The rest of the file is ignored. The user will be asked for a password if the password file is not specified.

To disable autostarting for particular user, a member of the administrators group must run the following command:

```
VBoxAutostartSvc delete --user=<user>
```

If a user has changed their password then a member of the administrators group must either reinstall the service or change the service credentials using Windows Service Manager. Due to Windows security policies, the autostart service cannot be installed for users with empty passwords.

2.23. Oracle VM VirtualBox Expert Storage Management

In case the snapshot model of Oracle VM VirtualBox is not sufficient it is possible to enable a special mode which makes it possible to reconfigure storage attachments while the VM is paused. The user has to make sure that the disk data stays consistent to the guest because unlike with hotplugging the guest is not informed about detached or newly attached media.

The expert storage management mode can be enabled per VM executing:

```
$ VBoxManage setextradata VM-name "VBoxInternal2/SilentReconfigureWhilePaused" 1
```

You can reconfigure storage attachments later while the VM is paused by using the VBoxManage storageattach command.

2.24. Handling of Host Power Management Events

Some host power management events are handled by Oracle VM VirtualBox. The actual behavior depends on the platform:

- **Host Suspends.** This event is generated when the host is about to suspend, that is, the host saves the state to some non-volatile storage and powers off.

  This event is currently only handled on Windows hosts and Mac OS X hosts. When this event is generated, Oracle VM VirtualBox will pause all running VMs.

- **Host Resumes.** This event is generated when the host woke up from the suspended state.

  This event is currently only handled on Windows hosts and Mac OS X hosts. When this event is generated, Oracle VM VirtualBox will resume all VMs which are where paused before.

- **Battery Low.** The battery level reached a critical level, usually less than 5 percent charged.

  This event is currently only handled on Windows hosts and Mac OS X hosts. When this event is generated, Oracle VM VirtualBox will save the state and terminate all VMs in preparation of a potential host powerdown.

The behavior can be configured. By executing the following command, no VM is saved:

```
$ VBoxManage setextradata global "VBoxInternal2/SavestateOnBatteryLow" 0
```

This is a global setting as well as a per-VM setting. The per-VM value has higher precedence than the global value. The following command will save the state of all VMs but will not save the state of VM "foo":

```
$ VBoxManage setextradata global "VBoxInternal2/SavestateOnBatteryLow" 1
$ VBoxManage setextradata "foo" "VBoxInternal2/SavestateOnBatteryLow" 0
```

The first line is actually not required as by default the savestate action is performed.

2.25. Passing Through SSE4.1/SSE4.2 Instructions

To provide SSE 4.1/SSE 4.2 support to guests, the host CPU has to implement these instruction sets. The instruction sets are exposed to guests by default, but it is possible to disable the instructions for certain guests by using the following commands:

```
$ VBoxManage setextradata VM-name \
VBoxInternal/CPUM/IsaExts/SSE4.1 0
$ VBoxManage setextradata VM-name \
VBoxInternal/CPUM/IsaExts/SSE4.2 0
```

These are per-VM settings which are enabled by default.

2.26. Support for Keyboard Indicator Synchronization

This feature makes the host keyboard indicators (LEDs) match those of the VM's emulated keyboard when the machine window is active. It is currently implemented for Mac OS X and Windows hosts. This feature is enabled by default on supported host OSes. You can disable this feature by running the following command:

```
$ VBoxManage setextradata VM-name GUI/HidLedsSync 0
```

This is a per-VM setting that is enabled by default.

2.27. Capturing USB Traffic for Selected Devices

You can capture USB traffic for single USB devices or on the root hub level, which captures the traffic of all USB devices attached to the root hub. Oracle VM VirtualBox stores the traffic in a format which is compatible with Wireshark. To capture the traffic of a specific USB device it must be attached to the VM with `VBoxManage` using the following command:

```
VBoxManage controlvm VM-name usbattach device uuid|address --capturefile filename
```

In order to enable capturing on the root hub use the following command while the VM is not running:

```
VBoxManage setextradata VM-name \
VBoxInternal/Devices/usb-ehci/0/LUN#0/Config/CaptureFilename filename
```

The command above enables capturing on the root hub attached to the EHCI controller. To enable it for the OHCI or XHCI controller replace `usb-ehci` with `usb-ohci` or `usb-xhci`, respectively.

2.28. Configuring the Heartbeat Service

Oracle VM VirtualBox ships a simple heartbeat service. Once the Guest Additions are active, the guest sends frequent heartbeat pings to the host. If the guest stops sending the heartbeat pings without properly

terminating the service, the VM process will log this event in the VBox.log file. In the future it might be possible to configure dedicated actions but for now there is only a warning in the log file.

There are two parameters to configure. The *heartbeat interval* defines the time between two heartbeat pings. The default value is 2 seconds, that is, the heartbeat service of the Oracle VM VirtualBox Guest Additions will send a heartbeat ping every two seconds. The value in nanoseconds can be configured like this:

```
VBoxManage setextradata VM-name \
VBoxInternal/Devices/VMMDev/0/Config/HeartbeatInterval 2000000000
```

The *heartbeat timeout* defines the time the host waits starting from the last heartbeat ping before it defines the guest as unresponsive. The default value is 2 times the heartbeat interval (4 seconds) and can be configured as following, in nanoseconds:

```
VBoxManage setextradata VM-name \
VBoxInternal/Devices/VMMDev/0/Config/HeartbeatTimeout 4000000000
```

If the heartbeat timeout expires, there will be a log message like *VMMDev: HeartBeatCheckTimer: Guest seems to be unresponsive. Last heartbeat received 5 seconds ago.* If another heartbeat ping arrives after this warning, there will be a log message like *VMMDev: GuestHeartBeat: Guest is alive.*

2.29. Encryption of Disk Images

Oracle VM VirtualBox enables you to transparently encrypt the data stored in hard disk images for the guest. It does not depend on a specific image format to be used. Images which have the data encrypted are not portable between Oracle VM VirtualBox and other virtualization software.

Oracle VM VirtualBox uses the AES algorithm in XTS mode and supports 128-bit or 256-bit data encryption keys (DEK). The DEK is stored encrypted in the medium properties and is decrypted during VM startup by entering a password which was chosen when the image was encrypted.

Since the DEK is stored as part of the VM configuration file, it is important that it is kept safe. Losing the DEK means that the data stored in the disk images is lost irrecoverably. Having complete and up to date backups of all data related to the VM is the responsibility of the user.

2.29.1. Limitations of Disk Encryption

There are some limitations the user needs to be aware of when using this feature:

- This feature is part of the Oracle VM VirtualBox Extension Pack, which needs to be installed. Otherwise disk encryption is unavailable.
- Since encryption works only on the stored user data, it is currently not possible to check for metadata integrity of the disk image. Attackers might destroy data by removing or changing blocks of data in the image or change metadata items such as the disk size.
- Exporting appliances which contain encrypted disk images is not possible because the OVF specification does not support this. All images are therefore decrypted during export.
- The DEK is kept in memory while the VM is running to be able to decrypt data read and encrypt data written by the guest. While this should be obvious the user needs to be aware of this because an attacker might be able to extract the key on a compromised host and decrypt the data.
- When encrypting or decrypting the images, the password is passed in clear text using the Oracle VM VirtualBox API. This needs to be kept in mind, especially when using third party API clients which make use of the webservice where the password might be transmitted over the network. The use of HTTPS is mandatory in such a case.

- Encrypting images with differencing images is only possible if there are no snapshots or a linear chain of snapshots. This limitation may be addressed in a future Oracle VM VirtualBox version.

2.29.2. Encrypting Disk Images

Encrypting disk images can be done either using the GUI or `VBoxManage`. While the GUI is easier to use, it works on a per VM basis and encrypts all disk images attached to the specific VM. With `VBoxManage` one can encrypt individual images, including all differencing images. To encrypt an unencrypted medium with `VBoxManage`, use:

```
VBoxManage encryptmedium uuid|filename \
--newpassword filename|- --cipher cipher-ID --newpasswordid "ID
```

To supply the encryption password point `VBoxManage` to the file where the password is stored or specify `-` to let VBoxManage ask you for the password on the command line.

The cipher parameter specifies the cipher to use for encryption and can be either `AES-XTS128-PLAIN64` or `AES-XTS256-PLAIN64`. The specified password identifier can be freely chosen by the user and is used for correct identification when supplying multiple passwords during VM startup.

If the user uses the same password when encrypting multiple images and also the same password identifier, the user needs to supply the password only once during VM startup.

2.29.3. Starting a VM with Encrypted Images

When a VM is started using the GUI, a dialog will open where the user needs to enter all passwords for all encrypted images attached to the VM. If another frontend like VBoxHeadless is used, the VM will be paused as soon as the guest tries to access an encrypted disk. The user needs to provide the passwords through `VBoxManage` using the following command:

```
VBoxManage controlvm uuid|vmname addencpassword ID password [--removeonsuspend yes|no]
```

`ID` must be the same as the password identifier supplied when encrypting the images. *`password`* is the password used when encrypting the images. Optionally, you can specify `--removeonsuspend yes|no` to specify whether to remove the password from VM memory when the VM is suspended. Before the VM can be resumed, the user needs to supply the passwords again. This is useful when a VM is suspended by a host suspend event and the user does not want the password to remain in memory.

2.29.4. Decrypting Encrypted Images

In some circumstances it might be required to decrypt previously encrypted images. This can be done in the GUI for a complete VM or using `VBoxManage` with the following command:

```
VBoxManage encryptmedium uuid|filename --oldpassword file|-
```

The only required parameter is the password the image was encrypted with. The options are the same as for encrypting images.

2.30. Paravirtualized Debugging

This section covers debugging of guest operating systems using interfaces supported by paravirtualization providers.

Note

Paravirtualized debugging significantly alter guest operating system behaviour and should only be used by expert users for debugging and diagnostics.

These debug options are specified as a string of key-value pairs separated by commas. An empty string disables paravirtualized debugging.

2.30.1. Hyper-V Debug Options

All of the options listed below are optional, and thus the default value specified will be used when the corresponding key-value pair is not specified.

- Key: **`enabled`**

  Value: `0` or `1`

  Default: `0`

  Specify `1` to enable the Hyper-V debug interface. If this key-value pair is not specified or the value is not `1`, the Hyper-V debug interface is disabled regardless of other key-value pairs being present.

- Key: **`address`**

  Value: IPv4 address

  Default: 127.0.0.1

  Specify the IPv4 address where the remote debugger is connected.

- Key: **`port`**

  Value: UDP port number

  Default: 50000

  Specify the UDP port number where the remote debugger is connected.

- Key: **`vendor`**

  Value: Hyper-V vendor signature reported by CPUID to the guest

  Default: When debugging is enabled: `Microsoft Hv`, otherwise: `VBoxVBoxVBox`

  Specify the Hyper-V vendor signature which is exposed to the guest by CPUID. For debugging Microsoft Windows guests, it is required the hypervisor reports the Microsoft vendor.

- Key: **`hypercallinterface`**

  Value: `0` or `1`

  Default: `0`

  Specify whether hypercalls should be suggested for initiating debug data transfers between host and guest rather than MSRs when requested by the guest.

- Key: **`vsinterface`**

  Value: `0` or `1`

  Default: When debugging is enabled, `1`, otherwise `0`

  Specify whether to expose the VS#1 virtualization service interface to the guest. This interface is required for debugging Microsoft Windows 10 32-bit guests, but is optional for other Windows versions.

2.30.1.1. Setting up Windows Guests for Debugging with the Hyper-V Paravirtualization Provider

Windows supports debugging over a serial cable, USB, IEEE 1394 Firewire, and Ethernet. USB and IEEE 1394 are not applicable for virtual machines, and Ethernet requires Windows 8 or later. While a serial connection is universally usable, it is slow.

Debugging using the Hyper-V debug transport, supported on Windows Vista and later, offers significant benefits. It provides excellent performance due to direct host-to-guest transfers, it is easy to set up and requires minimal support from the hypervisor. It can be used with the debugger running on the same host as the VM or with the debugger and VM on separate machines connected over a network.

Prerequisites

- A VM configured for Hyper-V paravirtualization running a Windows Vista or newer Windows guest. You can check the effective paravirtualization provider for your VM with the output of the following `VBoxManage` command:

```
$ VBoxManage showvminfo VM-name
```

- A sufficiently up-to-date version of the Microsoft WinDbg debugger required to debug the version of Windows in your VM.

- While Windows 8 and newer Windows guests ship with Hyper-V debug support, Windows 7 and Vista do not. To use Hyper-V debugging with a Windows 7 or Vista guest, copy the file `kdvm.dll` from a Windows 8.0 installation. This file is typically located in `C:\Windows\System32`. Copy it to the same location in your Windows 7/Vista guest. Make sure you copy the 32-bit or 64-bit version of the DLL which matches your guest OS.

Note

Only Windows 8.0 ships `kdvm.dll`. Windows 8.1 and newer Windows versions do not.

VM and Guest Configuration

1. Power off the VM.

2. Enable the debug options with the following `VBoxManage` command:

```
$ VBoxManage modifyvm VM-name --paravirtdebug "enabled=1"
```

   The above command assumes your debugger will connect to your host machine on UDP port 50000. However, if you need to run the debugger on a remote machine you may specify the remote address and port here. For example:

```
$ VBoxManage modifyvm VM-name \
--paravirtdebug "enabled=1,address=192.168.32.1,port=55000"
```

   See Section 2.30.1, “Hyper-V Debug Options” for the complete set of options.

3. Start the VM.

4. In the guest, start an elevated command prompt and execute the following commands:

   - For a Windows 8 or newer Windows guest:

```
bcdedit /dbgsettings net hostip:5.5.5.5 port:50000 key:1.2.3.4
```

- For a Windows 7 or Vista guest:

```
bcdedit /set loadoptions host_ip=5.5.5.5,host_port=50000,encryption_key=1.2.3.4
```

```
bcdedit /set dbgtransport kdvm.dll
```

The IP address and port in the `bcdedit` command are ignored when using the Hyper-V debug transport. Any valid IP and a port number greater than 49151 and lower than 65536 can be entered.

The encryption key in the `bcdedit` command is relevant and must be valid. The key "1.2.3.4" used in the above example is valid and may be used if security is not a concern. If you do not specify any encryption key, `bcdedit` will generate one for you and you will need to copy this key to later enter in Microsoft WinDbg on the remote end. This encryption key is used to encrypt the debug data exchanged between Windows and the debugger.

- Run one or more of the following commands to enable debugging for the appropriate phase or component of your Windows guest:

```
bcdedit /set debug on
```

```
bcdedit /set bootdebug on
```

```
bcdedit /set {bootmgr} bootdebug on
```

Please note that the `bootdebug` options are only effective on Windows 8 or newer when using the Hyper-V debug transport. Refer to Microsoft Windows documentation for detailed explanation of `bcdedit` options.

5. Start Microsoft WinDbg on your host machine or remote host.

   From the **File** menu, select **Kernel Debug**. On the **NET** tab, specify the UDP port number you used in the `paravirtdebug` options. If you did not specify any, leave it as 50000. Ensure that the UDP port is not blocked by a firewall or other security software.

   In the **Key** field, enter `1.2.3.4` or the encryption key from the `bcdedit` command in your Windows guest.

   Click **OK** to start listening for connections. Microsoft WinDbg typically shows a Waiting to Reconnect message during this phase.

   Alternatively, to directly start a debug session, run WinDbg from the command line as follows :

```
windbg.exe -k net:port=50000,key=1.2.3.4
```

   See the WinDbg documentation for the complete command line syntax.

6. Reboot your Windows guest and it should then connect as a debuggee with Microsoft WinDbg.

2.31. PC Speaker Passthrough

As an experimental feature, primarily due to being limited to Linux host only and unknown Linux distribution coverage, Oracle VM VirtualBox supports passing through the PC speaker to the host. The PC speaker, sometimes called the system speaker, is a way to produce audible feedback such as beeps without the need for regular audio and sound card support.

The PC speaker passthrough feature in Oracle VM VirtualBox handles beeps only. Advanced PC speaker use by the VM, such as PCM audio, will not work, resulting in undefined host behavior.

Producing beeps on Linux is a very complex topic. Oracle VM VirtualBox offers a collection of options, in an attempt to make this work deterministically and reliably on as many Linux distributions and system configurations as possible. These are summarized in the following table.

Table 2.4 PC Speaker Configuration Options

Code	Device	Notes
1	`/dev/input/by-path/platform-pcspkr-event-spkr`	Direct host PC speaker use.
2	`/dev/tty`	Uses the terminal association of the VM process. VM needs to be started on a virtual console.
3	`/dev/tty0` or `/dev/vc/0`	Can only be used by user `root` or users with `cap_sys_tty_config` capability.
9	A user-specified console or evdev device path.	As for codes 1 to 3, but with a custom device path.
70	`/dev/tty`	Standard beep only. Loses frequency and length. See code 2.
79	A user-specified terminal device path.	As for code 70, but with a custom device path.
100	All of the above.	Tries all the available codes.

To enable PC speaker passthrough use the following command:

```
VBoxManage setextradata VM-name "VBoxInternal/Devices/i8254/0/Config/PassthroughSpeaker" N
```

Replace *N* with the code representing the case you want to use. Changing this setting takes effect when you next start the VM. It is safe to enable PC speaker passthrough on all host OSes. It will only have an effect on Linux.

The VM log file, `VBox.log`, contains lines with the prefix `PIT: speaker:` showing the PC speaker passthrough setup activities. It gives hints which device it picked or why it failed.

Enabling PC speaker passthrough for the VM is usually the simple part. The real difficulty is making sure that Oracle VM VirtualBox can access the necessary device, because in a typical Linux install most of them can only be accessed by user `root`. You should follow the preferred way to persistently change this, such as by referring to your distribution's documentation. Since there are countless Linux distribution variants, we can only give the general hints that there is often a way to give the X11 session user access to additional devices, or you need to find a working solution using a udev configuration file. If everything fails you might try setting the permissions using a script which is run late enough in the host system startup.

Sometimes additional rules are applied by the kernel to limit access. For example, that the VM process must have the same controlling terminal as the device configured to be used for beeping, something which is often very difficult to achieve for GUI applications such as Oracle VM VirtualBox. The table above contains some hints, but in general refer to the Linux documentation.

If you have trouble getting any beeps even if the device permissions are set up and VBox.log confirms that it uses evdev or console for the PC speaker control, check if your system has a PC speaker. Some systems do not have one. Other complications can arise from Linux rerouting the PC speaker output to a sound card. Check if the beeps are audible if you connect speakers to your sound card. Today almost all

systems have one. Finally, check if the audio mixer control has a channel named "beep", which could be hidden in the mixer settings, and that it is not muted.

2.32. Accessing USB devices Exposed Over the Network with USB/IP

Oracle VM VirtualBox supports passing through USB devices which are exposed over the network using the USB over IP protocol without the need to configure the client side provided by the kernel and usbip tools. Furthermore, this feature works with Oracle VM VirtualBox running on any supported host, rather than just Linux alone, as is the case with the official client.

To enable support for passing through USB/IP devices, use the following command to add the device server that exports the devices:

```
VBoxManage usbdevsource add unique-name --backend USB-IP --address device-server[:port]
```

USB devices exported on the device server are then accessible through the GUI or `VBoxManage`, like any USB devices attached locally. This can be used multiple times to access different device servers.

To remove a device server, the following command can be used:

```
$ VBoxManage usbdevsource remove unique-name
```

2.32.1. Setting up USB/IP Support on a Linux System

This section gives a brief overview on how to set up a Linux based system to act as a USB device server. The system on the server requires that the `usbip-core.ko` and `usbip-host.ko` kernel drivers are available, and that the USB/IP tools package is installed. The particular installation method for the necessary tools depends on which distribution is used. For example, for Debian based systems, use the following command to install the required tools:

```
$ apt-get install usbip-utils
```

To check whether the necessary tools are already installed use the following command:

```
$ usbip list -l
```

This should produce output similar to that shown in the example below:

```
 - busid 4-2 (0bda:0301)
   Realtek Semiconductor Corp. : multicard reader (0bda:0301)

 - busid 5-1 (046d:c52b)
   Logitech, Inc. : Unifying Receiver (046d:c52b)
```

If everything is installed, the USB/IP server needs to be started as `root` using the following command:

```
# usbipd -D
```

See the documentation for the installed distribution to determine how to start the service when the system boots.

By default, no device on the server is exported. This must be done manually for each device. To export a device use the following command:

```
# usbip bind -b "bus identifier"
```

To export the multicard reader in the previous example:

```
# usbip bind -b 4-2
```

2.32.2. Security Considerations

The communication between the server and client is unencrypted and there is no authorization required to access exported devices. An attacker might sniff sensitive data or gain control over a device. To mitigate this risk, the device should be exposed over a local network to which only trusted clients have access. To access the device remotely over a public network, a VPN solution should be used to provide the required level of security protection.

2.33. Using Hyper-V with Oracle VM VirtualBox

Oracle VM VirtualBox can be used on a Windows host where Hyper-V is running. This is an experimental feature.

No configuration is required. Oracle VM VirtualBox detects Hyper-V automatically and uses Hyper-V as the virtualization engine for the host system. The CPU icon in the VM window status bar indicates that Hyper-V is being used.

Note

When using this feature, some host systems might experience significant Oracle VM VirtualBox performance degradation.

2.34. Nested Virtualization

Oracle VM VirtualBox supports *nested virtualization* on host systems that run AMD and Intel CPUs. This feature enables the passthrough of hardware virtualization functions to the guest VM. That means that you can install a hypervisor, such as Oracle VM VirtualBox, Oracle VM Server or KVM, on an Oracle VM VirtualBox guest. You can then create and run VMs within the guest VM.

You can enable the nested virtualization feature in one of the following ways:

- From the VirtualBox Manager, select the **Enable Nested VT-x/AMD-V** check box on the **Processor** tab. To disable the feature, deselect the check box.
- Use the `--nested-hw-virt` option of the `VBoxManage modifyvm` command to enable or disable nested virtualization. See VBoxManage modifyvm.

2.35. VISO file format / RTIsoMaker

ISO image maker.

2.35.1. Synopsis

`RTIsoMaker` [*`options`*] [@*`commands.rsp`*] { *`filespec`* ...}

2.35.2. Description

Construct a virtual ISO 9660 / Joliet / UDF / HFS hybrid image and either write it to a file (RTIsoMaker) or serve it as a virtual image (VISO).

2.35.2.1. VISO file format

A VISO file is a virtual ISO image, i.e. constructed in memory from a bunch of files on the host. A VISO is just the recipe describing how to go about this using a syntax vaguely similar to mkisofs and genisoimage.

One requirement is that the VISO file must start with one of the `--iprt-iso-maker-file-marker` options. Which of the options you use will dictate the quoting and escaping rules used when reading the file. The option takes the image UUID as an argument.

The VISO files are treated as UTF-8 and must not contain any byte order marker (BOM). There is currently no way to comment out lines in a VISO file.

2.35.2.2. File specifications and --name-setup

All non-options that does not start with '@' are taken to indicate a file, directory, or similar that is should be added to the ISO image. Directories are added recursively and content is subject to filtering options.

Since there can be up to six different namespaces on an ISO, it is handy to be able to control the names used in each and be able to exclude an object from one or more namespaces. The `--name-setup` option specifies the file specification format to use forthwith.

The default setup is:

```
--name-setup iso+joliet+udf+hfs
```

Which means you specify one on-ISO name for all namespaces followed by '=' and the source file system name. Only specifying the source file system will add the file/dir/whatever to the root of the ISO image.

Lets look at the following two examples:

```
/docs/readme.txt=/home/user/Documents/product-x-readme.txt
```

```
/home/user/Documents/product-x-readme.txt
```

In the first case the file `'/home/user/Documents/product-x-readme.txt'` is added to the ISO image as `'/docs/readme.txt'` in all enabled namespaces. In the primary ISO 9660 namespace, the file name will by default be converted to upper case because it's required by the spec.

In the second case the file is added to the root under the name `'product-x-readme.txt'` in all namespaces. Though, in the primary ISO 9660 namespace the name will be transformed to apply with the current ISO level, probably uppercased, possibly truncated too.

Given `--name-setup iso,joliet,udf` you can specify the name individually for each of the three namespace, if you like. If you omit any, they will use last name given. Any names left blank (`==`) will be considered omitted.

A different name in each namespace:

```
/ISO.TXT=/Joliet.TxT=/UDF.txt=/tmp/iso/real.txt
```

Specific name in the ISO 9660 namespace, same in the rest:

```
/ISO.TXT=/OtherNamespaces.TxT=/tmp/iso/real.txt
```

Omit the file from the ISO 9660 namespace:

```
=/OtherNamespaces.TxT=/tmp/iso/real.txt
```

Omit the file from the joliet namespace:

```
/ISO.TXT==/UDF.TxT=/tmp/iso/real.txt
```

Use the same file name as the source everywhere:

`/tmp/iso/real.txt`

Using for instance `--name-setup udf` you can add a files/dirs/whatever to select namespace(s) without the more complicated empty name syntax above.

When adding directories, you can only control the naming and omitting of the directory itself, not any recursively added files and directories below it.

2.35.3. Options

2.35.3.1. General

`-o output-file`, `--output=output-file`

The output file name. This option is not supported in VISO mode.

`--name-setup spec`

Configures active namespaces and how file specifications are to be interpreted. The specification is a comma separated list. Each element in the list is a sub-list separated by space, `'+'` or `'|'` giving the namespaces that elements controls. Namespaces are divied into two major and minor ones, you cannot specifying a minor before the major it belongs to.

Major namespaces and aliases in parentheses:
- iso (primary, iso9660, iso-9660, primary-iso, iso-primary)
- joliet
- udf
- hfs (hfs-plus)

Minor namespaces:
- rock: rock ridge on previous major namespace (iso / joliet)
- iso-rock: rock ridge extensions on primary ISO 9660 namespace
- joliet-rock: rock ridge on joliet namespace (just for fun)
- trans-tbl: translation table file on previous major namespace
- iso-trans-tbl
- joliet-trans-tbl
- udf-trans-tbl
- hfs-trans-tbl

`--push-iso=iso-file`, `--push-iso-no-joliet=iso-file`, `--push-iso-no-rock-iso-file`, `--push-iso-no-rock-no-joliet=iso-file`

Open the specified ISO file and use it as source file system until the corresponding `--pop` options is encountered. The variations are for selecting which namespace on the ISO to (not) access. These options are handy for copying files/directories/stuff from an ISO without having to extract them first or using the `:iprtvfs:` syntax.

`--pop`

Pops a `--push-iso` of the source file system stack.

`--import-iso=iso-file`

Imports everything on the given ISO file, including boot configuration and system area (first 16 sectors) content. You can use `--name-setup` to omit namespaces.

2.35.3.2. Namespaces

`--iso-level=0|1|2|3`

Sets the ISO level:
- 0: Disable primary ISO namespace.
- 1: ISO level 1: Filenames 8.3 format and limited to 4GB - 1.

- 2: ISO level 2: 31 char long names and limited to 4GB - 1.
- 3: ISO level 3: 31 char long names and support for >=4GB files. (default)
- 4: Fictive level used by other tools. Not yet implemented.

`--rock-ridge`, `--limited-rock-ridge`, `--no-rock-ridge`
: Enables or disables rock ridge support for the primary ISO 9660 namespace. The `--limited-rock-ridge` option omits a couple of bits in the root directory that would make Linux pick rock ridge over joliet.

  Default: `--limited-rock-ridge`

`-J`, `--joliet`, `--no-joliet`
: Enables or disable the joliet namespace. This option must precede any file specifications.

  Default: `--joliet`

`--joliet-ucs-level=1|2|3`, `--ucs-level=1|2|3`
: Set the Joliet UCS support level. This is currently only flagged in the image but not enforced on the actual path names.

  Default level: 3

2.35.3.3. File Attributes

`--rational-attribs`
: Enables rational file attribute handling (default):
  - Owner ID is set to zero
  - Group ID is set to zero
  - Mode is set to 0444 for non-executable files.
  - Mode is set to 0555 for executable files.
  - Mode is set to 0555 for directories, preserving stick bits.

`--strict-attribs`
: Counters `--rational-attribs` and causes attributes to be recorded exactly as they appear in the source.

`--file-mode=mode`, `--no-file-mode`
: Controls the forced file mode mask for rock ridge, UDF and HFS.

`--dir-mode=mode`, `--no-dir-mode`
: Controls the forced directory mode mask for rock ridge, UDF and HFS.

`--new-dir-mode=mode`
: Controls the default mode mask (rock ridge, UDF, HFS) for directories that are created implicitly. The `--dir-mode` option overrides this.

`--chmod=mode:on-iso-file`
: Explictily sets the rock ridge, UDF and HFS file mode for a file/dir/whatever that has already been added to the ISO. The mode can be octal, `ra+x`, `a+r`, or `a+rx`. (Support for more complicated mode specifications may be implemented at a later point.)

  Note that only namespaces in the current --name-setup are affected.

`--chown=owner-id:on-iso-file`
: Explictily sets the rock ridge, UDF and HFS file owner ID (numeric) for a file/dir/whatever that has already been added to the ISO.

  Note that only namespaces in the current --name-setup are affected.

--chgrp=*group-id*:*on-iso-file*
: Explictily sets the rock ridge, UDF and HFS file group ID (numeric) for a file/dir/whatever that has already been added to the ISO.

Note that only namespaces in the current --name-setup are affected.

2.35.3.4. Booting

`--eltorito-new-entry`, `--eltorito-alt-boot` — Starts a new El Torito boot entry.

`--eltorito-add-image=`*`filespec`* — File specification of a file that should be added to the image and used as the El Torito boot image of the current boot entry.

`-b` *`on-iso-file`*, `--eltorito-boot=`*`on-iso-file`* — Specifies a file on the ISO as the El Torito boot image for the current boot entry.

`--eltorito-floppy-12`, `--eltorito-floppy-144`, `--eltorito-floppy-288`, `--no-emulation-boot`, `--hard-disk-boot` — Sets the boot image emulation type of the current El Torito boot entry.

`--boot-load-seg=`*`seg`* — Specify the image load segment for the current El Torito boot entry.

Default: 0x7c0

`--boot-load-size=`*`sectors`* — Specify the image load size in emulated sectors for the current El Torito boot entry.

Default: 4 (sectors of 512 bytes)

`--no-boot` — Indicates that the current El Torito boot entry isn't bootable. (The BIOS will allegedly configure the emulation, but not attempt booting.)

`--boot-info-table` — Write a isolinux/syslinux boot info table into the boot image for the current El Torito boot entry.

`--eltorito-platform-id=`*`id`* — Set the El Torito platform ID of the current entry, a new entry of the verification entry depending on when it's used. The ID must be one of: `x86`, `PPC`, `Mac`, `efi`

`-c` *`namespec`*, `--boot-catalog=`*`namespec`* — Enters the El Torito boot catalog into the namespaces as a file. The *`namespec`* uses the same format as a 'filespec', but omits the final source file system name component.

`-G` *`file`*, `--generic-boot=`*`file`* — Specifies a file that should be loaded at offset 0 in the ISO image. The file must not be larger than 32KB. When creating a hybrid image, parts of this may be regenerated by partition tables and such.

2.35.3.5. String properties (applied to active namespaces only)

`--abstract=`*`file-id`* — The name of the abstract file in the root dir.

`-A` *`text|_file-id`*, `--application-id=`*`text|_file-id`* — Application ID string or root file name. The latter must be prefixed with an underscore.

`--biblio=`*`file-id`* — The name of the bibliographic file in the root dir.

`--copyright=`*`file-id`*
: The name of the copyright file in the root dir.

`-P` *`text|_file-id`*, `--publisher=`*`text|_file-id`*
: Publisher ID string or root file name. The latter must be prefixed with an underscore.

`-p` *`text|_file-id`*, `--preparer=`*`text|_file-id`*
: Data preparer ID string or root file name. The latter must be prefixed with an underscore.

`--sysid=`*`text`*
: System ID string.

`--volid=`*`text`*, `--volume-id=`*`text`*
: Volume ID string (label). (It is possible to set different labels for primary ISO 9660, joliet, UDF and HFS by changing the active namespaces using the `--name-setup` option between `--volume-id` occurences.)

`--volset=`*`text`*
: Volume set ID string.

2.35.3.6. Compatibility:

`--graft-points`
: Alias for --name-setup iso+joliet+udf+hfs.

`-l`, `--long-names`
: Allow 31 charater file names. Just ensure ISO level >= 2 here.

`-R`, `--rock`
: Same as `--rock-ridge` and `--strict-attribs`.

`-r`, `--rational-rock`
: Same as `--rock-ridge` and `--rational-attribs`.

2.35.3.7. VISO Specific:

`--iprt-iso-maker-file-marker=`*`UUID`*, `--iprt-iso-maker-file-marker-bourne=`*`UUID`*, `--iprt-iso-maker-file-marker-bourne-sh=`*`UUID`*
: Used as first option in a VISO file to specify the file UUID and that it is formatted using bourne-shell argument quoting & escaping style.

`--iprt-iso-maker-file-marker-ms=`*`UUID`*, `--iprt-iso-maker-file-marker-ms-sh=`*`UUID`*
: Used as first option in a VISO file to specify the file UUID and that it is formatted using microsoft CRT argument quoting & escaping style.

2.35.3.8. Testing (not applicable to VISO):

`--output-buffer-size=`*`bytes`*
: Selects a specific output buffer size for testing virtual image reads.

`--random-output-buffer-size`
: Enables randomized buffer size for each virtual image read, using the current output buffer size (`--output-buffer-size`) as maximum.

`--random-order-verification=`*`size`*
: Enables verification pass of the image that compares blocks of the given size in random order from the virtual and output images.

Chapter 3 Technical Background

This chapter provides additional information for readers who are familiar with computer architecture and technology and wish to find out more about how Oracle VM VirtualBox works *under the hood*. The contents of this chapter are not required reading in order to use Oracle VM VirtualBox successfully.

3.1. Where Oracle VM VirtualBox Stores its Files

In Oracle VM VirtualBox, a virtual machine and its settings are described in a virtual machine settings file in XML format. In addition, most virtual machines have one or more virtual hard disks. These are typically represented by disk images, such as those in VDI format. The location of these files may vary, depending on the host operating system. See Section 3.1.1, “The Machine Folder”.

Global configuration data for Oracle VM VirtualBox is maintained in another location on the host. See Section 3.1.2, “Global Settings”.

3.1.1. The Machine Folder

By default, each virtual machine has a directory on your host computer where all the files of that machine are stored: the XML settings file, with a `.vbox` file extension, and its disk images. This is called the *machine folder*.

By default, this machine folder is located in a common folder called `VirtualBox VMs`, which Oracle VM VirtualBox creates in the current system user's home directory. The location of this home directory depends on the conventions of the host operating system, as follows:

- On Windows, this is the location returned by the `SHGetFolderPath` function of the Windows system library Shell32.dll, asking for the user profile. A typical location is `C:\Users\`*`username`*.
- On Linux, Mac OS X, and Oracle Solaris, this is generally taken from the environment variable `$HOME`, except for the user `root` where it is taken from the account database. This is a workaround for the frequent trouble caused by users using Oracle VM VirtualBox in combination with the tool `sudo`, which by default does not reset the environment variable `$HOME`.

  A typical location on Linux and Oracle Solaris is `/home/`*`username`* and on Mac OS X is `/Users/`*`username`*.

For simplicity, we abbreviate the location of the home directory as `$HOME`. Using that convention, the common folder for all virtual machines is `$HOME/VirtualBox VMs`.

As an example, when you create a virtual machine called "Example VM", Oracle VM VirtualBox creates the following:

- A machine folder: `$HOME/VirtualBox VMs/Example VM/`
- In the machine folder, a settings file: `Example VM.vbox`
- In the machine folder, a virtual disk image: `Example VM.vdi`.

This is the default layout if you use the **Create New Virtual Machine** wizard described in Creating Your First Virtual Machine. Once you start working with the VM, additional files are added. Log files are in a subfolder called `Logs`, and if you have taken snapshots, they are in a `Snapshots` subfolder. For each VM, you can change the location of its snapshots folder in the VM settings.

You can change the default machine folder by selecting **Preferences** from the **File** menu in the Oracle VM VirtualBox main window. Then, in the displayed window, click on the **General** tab. Alternatively, use the `VBoxManage setproperty machinefolder` command. See VBoxManage setproperty.

3.1.2. Global Settings

In addition to the files for the virtual machines, Oracle VM VirtualBox maintains global configuration data in the following directory:

- **Linux and Oracle Solaris:** `$HOME/.config/VirtualBox`.
- **Windows:** `$HOME/.VirtualBox`.
- **Mac OS X:** `$HOME/Library/VirtualBox`.

Oracle VM VirtualBox creates this configuration directory automatically, if necessary. You can specify an alternate configuration directory by either setting the `VBOX_USER_HOME` environment variable, or on Linux or Oracle Solaris by using the standard `XDG_CONFIG_HOME` variable. Since the global `VirtualBox.xml` settings file points to all other configuration files, this enables switching between several Oracle VM VirtualBox configurations.

In this configuration directory, Oracle VM VirtualBox stores its global settings file, an XML file called `VirtualBox.xml`. This file includes global configuration options and a list of registered virtual machines with pointers to their XML settings files.

3.1.3. Summary of Configuration Data Locations

The following table gives a brief overview of the configuration data locations on an Oracle VM VirtualBox host.

Table 3.1 Configuration File Locations

Setting	Location
Default machines folder	`$HOME/VirtualBox VMs`
Default disk image location	In each machine's folder
Machine settings file extension	`.vbox`
Media registry	Each machine settings file Media registration is done automatically when a storage medium is attached to a VM

3.1.4. Oracle VM VirtualBox XML Files

Oracle VM VirtualBox uses XML for both the machine settings files and the global configuration file, `VirtualBox.xml`.

All Oracle VM VirtualBox XML files are versioned. When a new settings file is created, for example because a new virtual machine is created, Oracle VM VirtualBox automatically uses the settings format of the current Oracle VM VirtualBox version. These files may not be readable if you downgrade to an earlier version of Oracle VM VirtualBox. However, when Oracle VM VirtualBox encounters a settings file from an earlier version, such as after upgrading Oracle VM VirtualBox, it attempts to preserve the settings format as much as possible. It will only silently upgrade the settings format if the current settings cannot be expressed in the old format, for example because you enabled a feature that was not present in an earlier version of Oracle VM VirtualBox.

In such cases, Oracle VM VirtualBox backs up the old settings file in the virtual machine's configuration directory. If you need to go back to the earlier version of Oracle VM VirtualBox, then you will need to manually copy these backup files back.

We intentionally do not document the specifications of the Oracle VM VirtualBox XML files, as we must reserve the right to modify them in the future. We therefore strongly suggest that you do not edit these files manually. Oracle VM VirtualBox provides complete access to its configuration data through its the `VBoxManage` command line tool, see VBoxManage and its API, see Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.

3.2. Oracle VM VirtualBox Executables and Components

Oracle VM VirtualBox was designed to be modular and flexible. When the Oracle VM VirtualBox graphical user interface (GUI) is opened and a VM is started, at least the following three processes are running:

- `VBoxSVC`, the Oracle VM VirtualBox service process which always runs in the background. This process is started automatically by the first Oracle VM VirtualBox client process and exits a short time after the last client exits. The first Oracle VM VirtualBox service can be the GUI, `VBoxManage`, `VBoxHeadless`, the web service amongst others. The service is responsible for bookkeeping, maintaining the state of all VMs, and for providing communication between Oracle VM VirtualBox components. This communication is implemented using COM/XPCOM.

Note

When we refer to *clients* here, we mean the local clients of a particular `VBoxSVC` server process, not clients in a network. Oracle VM VirtualBox employs its own client/server design to allow its processes to cooperate, but all these processes run under the same user account on the host operating system, and this is totally transparent to the user.

- The GUI process, `VirtualBoxVM`, a client application based on the cross-platform Qt library. When started without the `--startvm` option, this application acts as the VirtualBox Manager, displaying the VMs and their settings. It then communicates settings and state changes to `VBoxSVC` and also reflects changes effected through other means, such as the `VBoxManage` command.
- If the `VirtualBoxVM` client application is started with the `--startvm` argument, it loads the VMM library which includes the actual hypervisor and then runs a virtual machine and provides the input and output for the guest.

Any Oracle VM VirtualBox front-end, or client, will communicate with the service process and can both control and reflect the current state. For example, either the VM selector or the VM window or VBoxManage can be used to pause the running VM, and other components will always reflect the changed state.

The Oracle VM VirtualBox GUI application is only one of several available front ends, or clients. The complete list shipped with Oracle VM VirtualBox is as follows:

- `VirtualBoxVM`: The Qt front end implementing the VirtualBox Manager and running VMs.
- `VBoxManage`: A less user-friendly but more powerful alternative. See VBoxManage.
- `VBoxHeadless`: A VM front end which does not directly provide any video output and keyboard or mouse input, but enables redirection through the VirtualBox Remote Desktop Extension. See Section 1.1.2, “VBoxHeadless, the Remote Desktop Server”.
- `vboxwebsrv`: The Oracle VM VirtualBox web service process which enables control of an Oracle VM VirtualBox host remotely. This is described in detail in the Oracle VM VirtualBox Software Development Kit (SDK) reference. See Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.
- The Oracle VM VirtualBox Python shell: A Python alternative to `VBoxManage`. This is also described in the SDK reference.

Internally, Oracle VM VirtualBox consists of many more or less separate components. You may encounter these when analyzing Oracle VM VirtualBox internal error messages or log files. These include the following:

- IPRT: A portable runtime library which abstracts file access, threading, and string manipulation. Whenever Oracle VM VirtualBox accesses host operating features, it does so through this library for cross-platform portability.
- VMM (Virtual Machine Monitor): The heart of the hypervisor.
- EM (Execution Manager): Controls execution of guest code.
- REM (Recompiled Execution Monitor): Provides software emulation of CPU instructions.
- TRPM (Trap Manager): Intercepts and processes guest traps and exceptions.
- HM (Hardware Acceleration Manager): Provides support for VT-x and AMD-V.
- GIM (Guest Interface Manager): Provides support for various paravirtualization interfaces to the guest.
- PDM (Pluggable Device Manager): An abstract interface between the VMM and emulated devices which separates device implementations from VMM internals and makes it easy to add new emulated devices. Through PDM, third-party developers can add new virtual devices to Oracle VM VirtualBox without having to change Oracle VM VirtualBox itself.
- PGM (Page Manager): A component that controls guest paging.
- PATM (Patch Manager): Patches guest code to improve and speed up software virtualization.
- TM (Time Manager): Handles timers and all aspects of time inside guests.
- CFGM (Configuration Manager): Provides a tree structure which holds configuration settings for the VM and all emulated devices.
- SSM (Saved State Manager): Saves and loads VM state.
- VUSB (Virtual USB): A USB layer which separates emulated USB controllers from the controllers on the host and from USB devices. This component also enables remote USB.
- DBGF (Debug Facility): A built-in VM debugger.
- Oracle VM VirtualBox emulates a number of devices to provide the hardware environment that various guests need. Most of these are standard devices found in many PC compatible machines and widely supported by guest operating systems. For network and storage devices in particular, there are several options for the emulated devices to access the underlying hardware. These devices are managed by PDM.
- Guest Additions for various guest operating systems. This is code that is installed from within a virtual machine. See Guest Additions.
- The "Main" component is special. It ties all the above bits together and is the only public API that Oracle VM VirtualBox provides. All the client processes listed above use only this API and never access the hypervisor components directly. As a result, third-party applications that use the Oracle VM VirtualBox Main API can rely on the fact that it is always well-tested and that all capabilities of Oracle VM VirtualBox are fully exposed. It is this API that is described in the Oracle VM VirtualBox SDK. See Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.

3.3. Hardware vs. Software Virtualization

Oracle VM VirtualBox enables software in the virtual machine to run directly on the processor of the host, but an array of complex techniques is employed to intercept operations that would interfere with your host. Whenever the guest attempts to do something that could be harmful to your computer and its data, Oracle VM VirtualBox steps in and takes action. In particular, for lots of hardware that the guest believes to be accessing, Oracle VM VirtualBox simulates a certain "virtual" environment according to how you have configured a virtual machine. For example, when the guest attempts to access a hard disk, Oracle VM VirtualBox redirects these requests to whatever you have configured to be the virtual machine's virtual hard disk. This is normally an image file on your host.

Unfortunately, the x86 platform was never designed to be virtualized. Detecting situations in which Oracle VM VirtualBox needs to take control over the guest code that is executing, as described above, is difficult. There are two ways in which to achieve this:

- Intel and AMD processors have support for so-called *hardware virtualization*. This means that these processors can help Oracle VM VirtualBox to intercept potentially dangerous operations that a guest operating system may be attempting and also makes it easier to present virtual hardware to a virtual machine.

  These hardware features differ between Intel and AMD processors. Intel named its technology VT-x, AMD calls theirs AMD-V. The Intel and AMD support for virtualization is very different in detail, but not very different in principle.

  

  Note

  On many systems, the hardware virtualization features first need to be enabled in the BIOS before Oracle VM VirtualBox can use them.

- As opposed to other virtualization software, for many usage scenarios, Oracle VM VirtualBox does not *require* hardware virtualization features to be present. Through sophisticated techniques, Oracle VM VirtualBox virtualizes many guest operating systems entirely in *software*. This means that you can run virtual machines even on older processors which do not support hardware virtualization.

Even though Oracle VM VirtualBox does not always require hardware virtualization, enabling it is *required* in the following scenarios:

- Certain rare guest operating systems like OS/2 make use of very esoteric processor instructions that are not supported with our software virtualization. For virtual machines that are configured to contain such an operating system, hardware virtualization is enabled automatically.

- Oracle VM VirtualBox's 64-bit guest and multiprocessing (SMP) support both require hardware virtualization to be enabled. This is not much of a limitation since the vast majority of 64-bit and multicore CPUs ship with hardware virtualization. The exceptions to this rule are some legacy Intel and AMD CPUs.

Warning

Do not run other hypervisors, either open source or commercial virtualization products, together with Oracle VM VirtualBox. While several hypervisors can normally be *installed* in parallel, do not attempt to *run* several virtual machines from competing hypervisors at the same time. Oracle VM VirtualBox cannot track what another hypervisor is currently attempting to do on the same host, and especially if several products attempt to use hardware virtualization features such as VT-x, this can crash the entire host. Also, within Oracle VM VirtualBox, you can mix

software and hardware virtualization when running multiple VMs. In certain cases a small performance penalty will be unavoidable when mixing VT-x and software virtualization VMs. We recommend not mixing virtualization modes if maximum performance and low overhead are essential. This does *not* apply to AMD-V.

3.4. Paravirtualization Providers

Oracle VM VirtualBox enables the exposure of a paravirtualization interface, to facilitate accurate and efficient execution of software within a virtual machine. These interfaces require the guest operating system to recognize their presence and make use of them in order to leverage the benefits of communicating with the Oracle VM VirtualBox hypervisor.

Most modern mainstream guest operating systems, including Windows and Linux, ship with support for one or more paravirtualization interfaces. Hence, there is typically no need to install additional software in the guest to take advantage of this feature.

Exposing a paravirtualization provider to the guest operating system does not rely on the choice of host platforms. For example, the *Hyper-V* paravirtualization provider can be used for VMs to run on any host platform supported by Oracle VM VirtualBox and not just Windows.

Oracle VM VirtualBox provides the following interfaces:

- **Minimal**: Announces the presence of a virtualized environment. Additionally, reports the TSC and APIC frequency to the guest operating system. This provider is mandatory for running any Mac OS X guests.
- **KVM**: Presents a Linux KVM hypervisor interface which is recognized by Linux kernels version 2.6.25 or later. Oracle VM VirtualBox's implementation currently supports paravirtualized clocks and SMP spinlocks. This provider is recommended for Linux guests.
- **Hyper-V**: Presents a Microsoft Hyper-V hypervisor interface which is recognized by Windows 7 and newer operating systems. Oracle VM VirtualBox's implementation currently supports paravirtualized clocks, APIC frequency reporting, guest debugging, guest crash reporting and relaxed timer checks. This provider is recommended for Windows guests.

3.5. Details About Software Virtualization

Implementing virtualization on x86 CPUs with no hardware virtualization support is an extraordinarily complex task because the CPU architecture was not designed to be virtualized. The problems can usually be solved, but at the cost of reduced performance. Thus, there is a constant clash between virtualization performance and accuracy.

The x86 instruction set was originally designed in the 1970s and underwent significant changes with the addition of protected mode in the 1980s with the 286 CPU architecture and then again with the Intel 386 and its 32-bit architecture. Whereas the 386 did have limited virtualization support for real mode operation with V86 mode, as used by the "DOS Box" of Windows 3.x and OS/2 2.x, no support was provided for virtualizing the entire architecture.

In theory, software virtualization is not overly complex. There are four privilege levels, called *rings*, provided by the hardware. Typically only two rings are used: ring 0 for kernel mode and ring 3 for user mode. Additionally, one needs to differentiate between *host context* and *guest context*.

In host context, everything is as if no hypervisor was active. This might be the active mode if another application on your host has been scheduled CPU time. In that case, there is a host ring 3 mode and a host ring 0 mode. The hypervisor is not involved.

In guest context, however, a virtual machine is active. So long as the guest code is running in ring 3, this is not much of a problem since a hypervisor can set up the page tables properly and run that code natively on the processor. The problems mostly lie in how to intercept what the guest's kernel does.

There are several possible solutions to these problems. One approach is full software emulation, usually involving recompilation. That is, all code to be run by the guest is analyzed, transformed into a form which will not allow the guest to either modify or see the true state of the CPU, and only then executed. This process is obviously highly complex and costly in terms of performance. Oracle VM VirtualBox contains a recompiler based on QEMU which can be used for pure software emulation, but the recompiler is only activated in special situations, described below.

Another possible solution is paravirtualization, in which only specially modified guest OSes are allowed to run. This way, most of the hardware access is abstracted and any functions which would normally access the hardware or privileged CPU state are passed on to the hypervisor instead. Paravirtualization can achieve good functionality and performance on standard x86 CPUs, but it can only work if the guest OS can actually be modified, which is obviously not always the case.

Oracle VM VirtualBox chooses a different approach. When starting a virtual machine, through its ring-0 support kernel driver, Oracle VM VirtualBox has set up the host system so that it can run most of the guest code natively, but it has inserted itself at the "bottom" of the picture. It can then assume control when needed. If a privileged instruction is executed, the guest traps, in particular because an I/O register was accessed and a device needs to be virtualized, or external interrupts occur. Oracle VM VirtualBox may then handle this and either route a request to a virtual device or possibly delegate handling such things to the guest or host OS. In guest context, Oracle VM VirtualBox can therefore be in one of three states:

- Guest ring 3 code is run unmodified, at full speed, as much as possible. The number of faults will generally be low, unless the guest allows port I/O from ring 3. This is something we cannot do as we do not want the guest to be able to access real ports. This is also referred to as *raw mode*, as the guest ring-3 code runs unmodified.

- For guest code in ring 0, Oracle VM VirtualBox employs a clever trick. It actually reconfigures the guest so that its ring-0 code is run in ring 1 instead, which is normally not used in x86 operating systems). As a result, when guest ring-0 code, actually running n ring 1, such as a guest device driver attempts to write to an I/O register or execute a privileged instruction, the Oracle VM VirtualBox hypervisor in the "real" ring 0 can take over.

- The hypervisor (VMM) can be active. Every time a fault occurs, Oracle VM VirtualBox looks at the offending instruction and can relegate it to a virtual device or the host OS or the guest OS or run it in the recompiler.

  In particular, the recompiler is used when guest code disables interrupts and Oracle VM VirtualBox cannot figure out when they will be switched back on. In these situations, Oracle VM VirtualBox actually analyzes the guest code using its own disassembler. Also, certain privileged instructions such as LIDT need to be handled specially. Finally, any real-mode or protected-mode code, such as BIOS code, a DOS guest, or any operating system startup, is run in the recompiler entirely.

Unfortunately this only works to a degree. Among others, the following situations require special handling:

- Running ring 0 code in ring 1 causes a lot of additional instruction faults, as ring 1 is not allowed to execute any privileged instructions, of which guest's ring-0 contains plenty. With each of these faults, the VMM must step in and emulate the code to achieve the desired behavior. While this works, emulating thousands of these faults is very expensive and severely hurts the performance of the virtualized guest.

- There are certain flaws in the implementation of ring 1 in the x86 architecture that were never fixed. Certain instructions that *should* trap in ring 1 do not. This affects, for example, the LGDT/SGDT, LIDT/SIDT, or POPF/PUSHF instruction pairs. Whereas the "load" operation is privileged and can therefore

be trapped, the "store" instruction always succeed. If the guest is allowed to execute these, it will see the true state of the CPU, not the virtualized state. The CPUID instruction also has the same problem.

- A hypervisor typically needs to reserve some portion of the guest's address space, both linear address space and selectors, for its own use. This is not entirely transparent to the guest OS and may cause clashes.
- The SYSENTER instruction, used for system calls, executed by an application running in a guest OS always transitions to ring 0. But that is where the hypervisor runs, not the guest OS. In this case, the hypervisor must trap and emulate the instruction even when it is not desirable.
- The CPU segment registers contain a "hidden" descriptor cache which is not software-accessible. The hypervisor cannot read, save, or restore this state, but the guest OS may use it.
- Some resources must, and can, be trapped by the hypervisor, but the access is so frequent that this creates a significant performance overhead. An example is the TPR (Task Priority) register in 32-bit mode. Accesses to this register must be trapped by the hypervisor. But certain guest operating systems, notably Windows and Oracle Solaris, write this register very often, which adversely affects virtualization performance.

To fix these performance and security issues, Oracle VM VirtualBox contains a Code Scanning and Analysis Manager (CSAM), which disassembles guest code, and the Patch Manager (PATM), which can replace it at runtime.

Before executing ring 0 code, CSAM scans it recursively to discover problematic instructions. PATM then performs *in-situ* patching. It replaces the instruction with a jump to hypervisor memory where an integrated code generator has placed a more suitable implementation. In reality, this is a very complex task as there are lots of odd situations to be discovered and handled correctly. So, with its current complexity, one could argue that PATM is an advanced *in-situ* recompiler.

In addition, every time a fault occurs, Oracle VM VirtualBox analyzes the offending code to determine if it is possible to patch it in order to prevent it from causing more faults in the future. This approach works well in practice and dramatically improves software virtualization performance.

3.6. Details About Hardware Virtualization

With Intel VT-x, there are two distinct modes of CPU operation: VMX root mode and non-root mode.

- In root mode, the CPU operates much like older generations of processors without VT-x support. There are four privilege levels, called rings, and the same instruction set is supported, with the addition of several virtualization specific instruction. Root mode is what a host operating system without virtualization uses, and it is also used by a hypervisor when virtualization is active.
- In non-root mode, CPU operation is significantly different. There are still four privilege rings and the same instruction set, but a new structure called VMCS (Virtual Machine Control Structure) now controls the CPU operation and determines how certain instructions behave. Non-root mode is where guest systems run.

Switching from root mode to non-root mode is called "VM entry", the switch back is "VM exit". The VMCS includes a guest and host state area which is saved/restored at VM entry and exit. Most importantly, the VMCS controls which guest operations will cause VM exits.

The VMCS provides fairly fine-grained control over what the guests can and cannot do. For example, a hypervisor can allow a guest to write certain bits in shadowed control registers, but not others. This enables efficient virtualization in cases where guests can be allowed to write control bits without disrupting

the hypervisor, while preventing them from altering control bits over which the hypervisor needs to retain full control. The VMCS also provides control over interrupt delivery and exceptions.

Whenever an instruction or event causes a VM exit, the VMCS contains information about the exit reason, often with accompanying detail. For example, if a write to the CR0 register causes an exit, the offending instruction is recorded, along with the fact that a write access to a control register caused the exit, and information about source and destination register. Thus the hypervisor can efficiently handle the condition without needing advanced techniques such as CSAM and PATM described above.

VT-x inherently avoids several of the problems which software virtualization faces. The guest has its own completely separate address space not shared with the hypervisor, which eliminates potential clashes. Additionally, guest OS kernel code runs at privilege ring 0 in VMX non-root mode, obviating the problems by running ring 0 code at less privileged levels. For example the SYSENTER instruction can transition to ring 0 without causing problems. Naturally, even at ring 0 in VMX non-root mode, any I/O access by guest code still causes a VM exit, allowing for device emulation.

The biggest difference between VT-x and AMD-V is that AMD-V provides a more complete virtualization environment. VT-x requires the VMX non-root code to run with paging enabled, which precludes hardware virtualization of real-mode code and non-paged protected-mode software. This typically only includes firmware and OS loaders, but nevertheless complicates VT-x hypervisor implementation. AMD-V does not have this restriction.

Of course hardware virtualization is not perfect. Compared to software virtualization, the overhead of VM exits is relatively high. This causes problems for devices whose emulation requires high number of traps. One example is a VGA device in 16-color mode, where not only every I/O port access but also every access to the framebuffer memory must be trapped.

3.7. Nested Paging and VPIDs

In addition to normal hardware virtualization, your processor may also support the following additional sophisticated techniques:

- Nested paging implements some memory management in hardware, which can greatly accelerate hardware virtualization since these tasks no longer need to be performed by the virtualization software.

  With nested paging, the hardware provides another level of indirection when translating linear to physical addresses. Page tables function as before, but linear addresses are now translated to "guest physical" addresses first and not physical addresses directly. A new set of paging registers now exists under the traditional paging mechanism and translates from guest physical addresses to host physical addresses, which are used to access memory.

  Nested paging eliminates the overhead caused by VM exits and page table accesses. In essence, with nested page tables the guest can handle paging without intervention from the hypervisor. Nested paging thus significantly improves virtualization performance.

  On AMD processors, nested paging has been available starting with the Barcelona (K10) architecture. They now call it rapid virtualization indexing (RVI). Intel added support for nested paging, which they call extended page tables (EPT), with their Core i7 (Nehalem) processors.

  If nested paging is enabled, the Oracle VM VirtualBox hypervisor can also use *large pages* to reduce TLB usage and overhead. This can yield a performance improvement of up to 5%. To enable this feature for a VM, you use the `VBoxManage modifyvm --largepages` command. See VBoxManage modifyvm.

  If you have an Intel CPU with EPT, please consult Section 6.4.1, “CVE-2018-3646” for security concerns regarding EPT.

- On Intel CPUs, a hardware feature called Virtual Processor Identifiers (VPIDs) can greatly accelerate context switching by reducing the need for expensive flushing of the processor's Translation Lookaside Buffers (TLBs).

  To enable these features for a VM, you use the `VBoxManage modifyvm --vtxvpid` and `VBoxManage modifyvm --largepages` commands. See VBoxManage modifyvm.

Chapter 4 Oracle VM VirtualBox Programming Interfaces

Oracle VM VirtualBox comes with comprehensive support for third-party developers. The so-called *Main API* of Oracle VM VirtualBox exposes the entire feature set of the virtualization engine. It is completely documented and available to anyone who wishes to control Oracle VM VirtualBox programmatically.

The Main API is made available to C++ clients through COM on Windows hosts or XPCOM on other hosts. Bridges also exist for SOAP, Java and Python.

All programming information such as documentation, reference information, header and other interface files, as well as samples have been split out to a separate **Software Development Kit (SDK)**. The SDK is available for download from http://www.virtualbox.org. In particular, the SDK comes with a Programming Guide and Reference manual in PDF format. This manual contains, among other things, the information that was previously in this chapter of the User Manual.

Chapter 5 Troubleshooting

This chapter provides answers to commonly asked questions. In order to improve your user experience with Oracle VM VirtualBox, it is recommended to read this section to learn more about common pitfalls and get recommendations on how to use the product.

5.1. Procedures and Tools

5.1.1. Categorizing and Isolating Problems

More often than not, a virtualized guest behaves like a physical system. Any problems that a physical machine would encounter, a virtual machine will encounter as well. If, for example, Internet connectivity is lost due to external issues, virtual machines will be affected just as much as physical ones.

If a true Oracle VM VirtualBox problem is encountered, it helps to categorize and isolate the problem first. Here are some of the questions that should be answered before reporting a problem:

- Is the problem specific to a certain guest OS? Or a specific release of a guest OS? Especially with Linux guest related problems, the issue may be specific to a certain distribution and version of Linux.
- Is the problem specific to a certain host OS? Problems are usually not host OS specific, because most of the Oracle VM VirtualBox code base is shared across all supported platforms, but especially in the areas of networking and USB support, there are significant differences between host platforms. Some GUI related issues are also host specific.
- Is the problem specific to certain host hardware? This category of issues is typically related to the host CPU. Because of significant differences between VT-x and AMD-V, problems may be specific to one or the other technology. The exact CPU model may also make a difference, even for software virtualization, because different CPUs support different features, which may affect certain aspects of guest CPU operation.
- Is the problem specific to a certain virtualization mode? Some problems may only occur in software virtualization mode, others may be specific to hardware virtualization.
- Is the problem specific to guest SMP? That is, is it related to the number of virtual CPUs (VCPUs) in the guest? Using more than one CPU usually significantly affects the internal operation of a guest OS.
- Is the problem specific to the Guest Additions? In some cases, this is obvious, such as a shared folders problem. In other cases such as display problems, it may be less obvious. If the problem is Guest Additions specific, is it also specific to a certain version of the Guest Additions?
- Is the problem specific to a certain environment? Some problems are related to a particular environment external to the VM. This usually involves network setup. Certain configurations of external servers such as DHCP or PXE may expose problems which do not occur with other, similar servers.
- Is the problem a regression? Knowing that an issue is a regression usually makes it significantly easier to find the solution. In this case, it is crucial to know which version is affected and which is not.

5.1.2. Collecting Debugging Information

For problem determination, it is often important to collect debugging information which can be analyzed by Oracle VM VirtualBox support. This section contains information about what kind of information can be obtained.

Every time Oracle VM VirtualBox starts up a VM, a so-called *release log file* is created, containing lots of information about the VM configuration and runtime events. The log file is called `VBox.log` and resides in the VM log file folder, which is `$HOME/VirtualBox VMs/VM-name/Logs` by default.

When starting a VM, the configuration file of the last run will be renamed to `.1`, up to `.3`. Sometimes when there is a problem, it is useful to have a look at the logs. Also when requesting support for Oracle VM VirtualBox, supplying the corresponding log file is mandatory.

For convenience, for each virtual machine, the VirtualBox Manager window can show these logs in a window. To access it, select a virtual machine from the list on the left and select **Show Log** from the **Machine** menu.

The release log file, `VBox.log`, contains a wealth of diagnostic information, such as Host OS type and version, Oracle VM VirtualBox version and build. It also includes a complete dump of the guest's configuration (CFGM), detailed information about the host CPU type and supported features, whether hardware virtualization is enabled, information about VT-x/AMD-V setup, state transitions (such as creating, running, paused, stopping), guest BIOS messages, Guest Additions messages, device-specific log entries and, at the end of execution, final guest state and condensed statistics.

In case of crashes, it is very important to collect *crash dumps*. This is true for both host and guest crashes. For information about enabling core dumps on Linux, Oracle Solaris, and Mac OS X systems, refer to the following core dump article on the Oracle VM VirtualBox website:

http://www.virtualbox.org/wiki/Core_dump.

You can also use `VBoxManage debugvm` to create a dump of a complete virtual machine. See VBoxManage debugvm.

For network related problems, it is often helpful to capture a trace of network traffic. If the traffic is routed through an adapter on the host, it is possible to use Wireshark or a similar tool to capture the traffic there. However, this often also includes a lot of traffic unrelated to the VM.

Oracle VM VirtualBox provides an ability to capture network traffic only on a specific VM's network adapter. Refer to the following network tracing article on the Oracle VM VirtualBox website for information on enabling this capture:

http://www.virtualbox.org/wiki/Network_tips.

The trace files created by Oracle VM VirtualBox are in `.pcap` format and can be easily analyzed with Wireshark.

5.1.3. Using the VBoxBugReport Command to Collect Debug Information Automatically

The `VBoxBugReport` command is used to collect debug information automatically for an Oracle VM VirtualBox installation. This command can be useful when you need to gather information to send to Oracle Support.

The following examples show how to use `VBoxBugReport`.

By default, the command collects `VBoxSVC` process logs, device settings, and global configuration data for an Oracle VM VirtualBox host.

```
$ VBoxBugReport
  ...
  0% - collecting VBoxSVC.log.10...
  7% - collecting VBoxSVC.log.9...
  ...
 64% - collecting VBoxSVC.log.1...
 71% - collecting VBoxSVC.log...
```

```
 78% - collecting VirtualBox.xml...
 85% - collecting HostUsbDevices...
 92% - collecting HostUsbFilters...
100% - compressing...

Report was written to '2019-03-26-13-32-02-bugreport.tgz'
```

The results are saved as a compressed tar file archive in the same directory where the command is run.

To specify a different output file location:

```
$ VBoxBugReport --output ~/debug/bug004.tgz
```

To output all debug information to a single text file, rather than a `tgz` file:

```
$ VBoxBugReport --text
```

To collect information for a specific VM, called `Windows_10`:

```
$ VBoxBugReport Windows_10
```

This command collects machine settings, guest properties, and log files for the specified VM. Global configuration information for the host is also included.

To collect information for several VMs, called `Windows_7`, `Windows_8`, and `Windows_10`:

```
$ VBoxBugReport Windows_7 Windows_8 Windows_10
```

To collect information for all VMs:

```
$ VBoxBugReport --all
```

To show a full list of the available command options, run `VBoxBugReport --help`.

5.1.4. The Built-In VM Debugger

Oracle VM VirtualBox includes a built-in VM debugger, which advanced users may find useful. This debugger enables you to examine and, to some extent, control the VM state.

Warning

Use the VM debugger at your own risk. There is no support for it, and the following documentation is only made available for advanced users with a very high level of familiarity with the x86/AMD64 machine instruction set, as well as detailed knowledge of the PC architecture. A degree of familiarity with the internals of the guest OS in question may also be very helpful.

The VM debugger is available in all regular production versions of Oracle VM VirtualBox, but it is disabled by default because the average user will have little use for it. There are two ways to access the debugger:

- Using a debugger console window displayed alongside the VM
- Using the `telnet` protocol on port 5000

The debugger can be enabled in the following ways:

- Start the VM directly using `VirtualBoxVM --startvm`, with an additional `--dbg`, `--debug`, or `--debug-command-line` argument. See the `VirtualBoxVM --help` command usage help for details.

- Set the `VBOX_GUI_DBG_ENABLED` or `VBOX_GUI_DBG_AUTO_SHOW` environment variable to `true` before launching the Oracle VM VirtualBox process. Setting these variables, only their presence is checked, is effective even when the first Oracle VM VirtualBox process is the VM selector window. VMs subsequently launched from the selector will have the debugger enabled.
- Set the `GUI/Dbg/Enabled` extra data item to `true` before launching the VM. This can be set globally or on a per VM basis.

A new **Debug** menu entry is added to the Oracle VM VirtualBox application. This menu enables the user to open the debugger console.

The VM debugger command syntax is loosely modeled on Microsoft and IBM debuggers used on DOS, OS/2, and Windows. Users familiar with symdeb, CodeView, or the OS/2 kernel debugger will find the Oracle VM VirtualBox VM debugger familiar.

The most important command is `help`. This will print brief usage help for all debugger commands. The set of commands supported by the VM debugger changes frequently and the `help` command is always up-to-date.

A brief summary of frequently used commands is as follows:

- `stop`: Stops the VM execution and enables single stepping
- `g`: Continue VM execution
- `t`: Single step an instruction
- `rg`, `rh`, and `r`: Print the guest, hypervisor, and current registers
- `kg`, `kh`, and `k`: Print the guest, hypervisor, and current call stack
- `da`, `db`, `dw`, `dd`, `dq`: Print memory contents as ASCII, bytes, words, dwords, and qwords
- `u`: Unassemble memory
- `dg`: Print the guest's GDT
- `di`: Print the guest's IDT
- `dl`: Print the guest's LDT
- `dt`: Print the guest's TSS
- `dp*`: Print the guest's page table structures
- `bp` and `br`: Set a normal and recompiler breakpoint
- `bl`: List breakpoints
- `bc`: Clear a breakpoint
- `writecore`: Write a VM core file to disk. See Section 5.1.5, “VM Core Format”

See the built-in `help` for other available commands.

The VM debugger supports symbolic debugging, although symbols for guest code are often not available. For Oracle Solaris guests, the `detect` command automatically determines the guest OS version and locates kernel symbols in guest's memory. Symbolic debugging is then available. For Linux guests,

the `detect` commands also determines the guest OS version, but there are no symbols in the guest's memory. Kernel symbols are available in the file `/proc/kallsyms` on Linux guests. This file must be copied to the host, for example using `scp`. The `loadmap` debugger command can be used to make the symbol information available to the VM debugger. Note that the `kallsyms` file contains the symbols for the currently loaded modules. If the guest's configuration changes, the symbols will change as well and must be updated.

For all guests, a simple way to verify that the correct symbols are loaded is the `k` command. The guest is normally idling and it should be clear from the symbolic information that the guest operating system's idle loop is being executed.

Another group of debugger commands is the set of `info` commands. Running `info help` provides complete usage information. The information commands provide ad-hoc data pertinent to various emulated devices and aspects of the VMM. There is no general guideline for using the `info` commands, the right command to use depends entirely on the problem being investigated. Some of the `info` commands are as follows:

- `cfgm`: Print a branch of the configuration tree
- `cpuid`: Display the guest CPUID leaves
- `ioport`: Print registered I/O port ranges
- `mmio`: Print registered MMIO ranges
- `mode`: Print the current paging mode
- `pit`: Print the i8254 PIT state
- `pic`: Print the i8259A PIC state
- `ohci`, `ehci`, `xhci`: Print a subset of the OHCI, EHCI, and xHCI USB controller state
- `pcnet0`: Print the PCnet state
- `vgatext`: Print the contents of the VGA framebuffer formatted as standard text mode
- `timers`: Print all VM timers

The output of the `info` commands generally requires in-depth knowledge of the emulated device or Oracle VM VirtualBox VMM internals. However, when used properly, the information provided can be invaluable.

5.1.5. VM Core Format

Oracle VM VirtualBox uses the 64-bit ELF format for its VM core files created by `VBoxManage debugvm`, see VBoxManage debugvm. The VM core file contain the memory and CPU dumps of the VM and can be useful for debugging your guest OS. The 64-bit ELF object format specification can be obtained at:

http://downloads.openwatcom.org/ftp/devel/docs/elf-64-gen.pdf.

The overall layout of the VM core format is as follows:

```
[ ELF 64 Header]
[ Program Header, type PT_NOTE ]
  → offset to COREDESCRIPTOR
[ Program Header, type PT_LOAD ] - one for each contiguous physical memory range
  → Memory offset of range
  → File offset
```

```
[ Note Header, type NT_VBOXCORE ]
[ COREDESCRIPTOR ]
  → Magic
  → VM core file version
  → VBox version
  → Number of vCPUs etc.
[ Note Header, type NT_VBOXCPU ] - one for each vCPU
[ vCPU 1 Note Header ]
  [ DBGFCORECPU - vCPU 1 dump ]
[ Additional Notes + Data ] - currently unused
[ Memory dump ]
```

The memory descriptors contain physical addresses relative to the guest and not virtual addresses. Regions of memory such as MMIO regions are not included in the core file.

The relevant data structures and definitions can be found in the Oracle VM VirtualBox sources under the following header files: `include/VBox/dbgfcorefmt.h`, `include/iprt/x86.h` and `src/VBox/Runtime/include/internal/ldrELFCommon.h`.

The VM core file can be inspected using `elfdump` and GNU `readelf` or other similar utilities.

5.2. General Troubleshooting

5.2.1. Guest Shows IDE/SATA Errors for File-Based Images on Slow Host File System

Occasionally, some host file systems provide very poor writing performance and as a consequence cause the guest to time out IDE/SATA commands. This is normal behavior and should normally cause no real problems, as the guest should repeat commands that have timed out. However, guests such as some Linux versions have severe problems if a write to an image file takes longer than about 15 seconds. Some file systems however require more than a minute to complete a single write, if the host cache contains a large amount of data that needs to be written.

The symptom for this problem is that the guest can no longer access its files during large write or copying operations, usually leading to an immediate hang of the guest.

In order to work around this problem, the true fix is to use a faster file system that does not exhibit such unacceptable write performance, it is possible to flush the image file after a certain amount of data has been written. This interval is normally infinite, but can be configured individually for each disk of a VM.

For IDE disks use the following command:

```
VBoxManage setextradata VM-name
"VBoxInternal/Devices/piix3ide/0/LUN#[x]/Config/FlushInterval" [b]
```

For SATA disks use the following command:

```
VBoxManage setextradata VM-name
"VBoxInternal/Devices/ahci/0/LUN#[x]/Config/FlushInterval" [b]
```

`[x]` specifies the disk for IDE. `0` represents the master device on the first channel, `1` represents the slave device on the first channel, `2` represents the master device on the second channel, and `3` represents the slave device on the second channel. For SATA, use values between `0` and `29`. This configuration option applies to disks only. Do not use this option for CD or DVD drives.

The unit of the interval (`[b]`) is the number of bytes written since the last flush. The value for it must be selected so that the occasional long write delays do not occur. Since the proper flush interval depends

on the performance of the host and the host filesystem, finding the optimal value that makes the problem disappear requires some experimentation. Values between 1000000 and 10000000 (1 to 10 megabytes) are a good starting point. Decreasing the interval both decreases the probability of the problem and the write performance of the guest. Setting the value unnecessarily low will cost performance without providing any benefits. An interval of 1 will cause a flush for each write operation and should solve the problem in any case, but has a severe write performance penalty.

Providing a value of `0` for `[b]` is treated as an infinite flush interval, effectively disabling this workaround. Removing the extra data key by specifying no value for `[b]` has the same effect.

5.2.2. Responding to Guest IDE/SATA Flush Requests

If desired, the virtual disk images can be flushed when the guest issues the IDE FLUSH CACHE command. Normally these requests are ignored for improved performance. The parameters below are only accepted for disk drives. They must not be set for DVD drives.

To enable flushing for IDE disks, issue the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/Devices/piix3ide/0/LUN#[x]/Config/IgnoreFlush" 0
```

`[x]` specifies the disk. is 0 for the master device on the first channel, 1 for the slave device on the first channel, 2 for the master device on the second channel or 3 for the master device on the second channel.

To enable flushing for SATA disks, issue the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/Devices/ahci/0/LUN#[x]/Config/IgnoreFlush" 0
```

The value [x] that selects the disk can be a value between 0 and 29.

Note that this does not affect the flushes performed according to the configuration described in Section 5.2.1, “Guest Shows IDE/SATA Errors for File-Based Images on Slow Host File System”. Restoring the default of ignoring flush commands is possible by setting the value to 1 or by removing the key.

5.2.3. Performance Variation with Frequency Boosting

Many multicore processors support some form of frequency boosting, which means that if only one core is utilized, it can run possibly 50% faster or even more than the rated CPU frequency. This causes measured performance to vary somewhat as a function of the momentary overall system load. The exact behavior depends strongly on the specific processor model.

As a consequence, benchmarking on systems which utilize frequency boosting may produce unstable and non-repeatable results. This is especially true if benchmark runs are short, of the order of seconds. To obtain stable results, benchmarks must be run over longer periods of time and with a constant system load apart from the VM being tested.

5.2.4. Frequency Scaling Effect on CPU Usage

On some hardware platforms and operating systems, CPU frequency scaling may cause CPU usage reporting to be highly misleading. This happens in situations when the host CPU load is significant but not heavy, such as between 15% to 30% of the maximum.

Most operating systems determine CPU usage in terms of time spent, measuring for example how many nanoseconds the systems or a process was active within one second. However, in order to save energy, systems can significantly scale down CPU speed when the system is not fully loaded. When the CPU is

running at for example one half of its maximum speed, the same number of instructions will take roughly twice as long to execute compared to running at full speed.

Depending on the specific hardware and host OS, this effect can very significantly skew the CPU usage reported by the OS. The reported CPU usage can be several times higher than what it would have been had the CPU been running at full speed. The effect can be observed both on the host OS and in a guest OS.

5.2.5. Inaccurate Windows CPU Usage Reporting

CPU usage reporting tools which come with Windows, such as Task Manager or Resource Monitor, do not take the time spent processing hardware interrupts into account. If the interrupt load is heavy, with thousands of interrupts per second, CPU usage may be significantly underreported.

This problem affects Windows as both host and guest OS. Sysinternals tools, such as Process Explorer, do not suffer from this problem.

5.2.6. Poor Performance Caused by Host Power Management

On some hardware platforms and operating systems, virtualization performance is negatively affected by host CPU power management. The symptoms may be choppy audio in the guest or erratic guest clock behavior.

Some of the problems may be caused by firmware and/or host operating system bugs. Therefore, updating the firmware and applying operating systems fixes is recommended.

For optimal virtualization performance, the C1E power state support in the system's BIOS should be disabled, if such a setting is available. Not all systems support the C1E power state. On Intel systems, the `Intel C State` setting should be disabled. Disabling other power management settings may also improve performance. However, a balance between performance and power consumption must always be considered.

5.2.7. GUI: 2D Video Acceleration Option is Grayed Out

To use 2D Video Acceleration within Oracle VM VirtualBox, your host's video card should support certain OpenGL extensions. On startup, Oracle VM VirtualBox checks for those extensions, and, if the test fails, this option is silently grayed out.

To find out why it has failed, you can manually execute the following command:

```
$ VBoxTestOGL --log "log_file_name" --test 2D
```

It will list the required OpenGL extensions one by one and will show you which one failed the test. This usually means that you are running an outdated or misconfigured OpenGL driver on your host. It can also mean that your video chip is lacking required functionality.

5.3. Windows Guests

5.3.1. No USB 3.0 Support in Windows 7 Guests

If a Windows 7 or Windows Server 2008 R2 guest is configured for USB 3.0 (xHCI) support, the guest OS will not have any USB support at all. This happens because Windows 7 predates USB 3.0 and therefore does not ship with any xHCI drivers. Microsoft also does not offer any vendor-provided xHCI drivers through Windows Update.

To solve this problem, it is necessary to download and install the Intel xHCI driver in the guest. Intel offers the driver as the USB 3.0 eXtensible Host Controller (xHCI) driver for Intel 7 Series/C216 chipsets.

Note that the driver only supports Windows 7 and Windows Server 2008 R2. The driver package includes support for both 32-bit and 64-bit OS variants.

5.3.2. Windows Bluescreens After Changing VM Configuration

Changing certain virtual machine settings can cause Windows guests to fail during start up with a bluescreen. This may happen if you change VM settings after installing Windows, or if you copy a disk image with an already installed Windows to a newly created VM which has settings that differ from the original machine.

This applies in particular to the following settings:

- The ACPI and I/O APIC settings should never be changed after installing Windows. Depending on the presence of these hardware features, the Windows installation program chooses special kernel and device driver versions and will fail to startup should these hardware features be removed. Enabling them for a Windows VM which was installed without them does not cause any harm. However, Windows will not use these features in this case.

- Changing the storage controller hardware will cause bootup failures as well. This might also apply to you if you copy a disk image from an older version of Oracle VM VirtualBox to a new virtual machine. The default subtype of IDE controller hardware used by Oracle VM VirtualBox is PIIX4. Make sure that the storage controller settings are identical.

5.3.3. Windows 0x101 Bluescreens with SMP Enabled (IPI Timeout)

If a VM is configured to have more than one processor (symmetrical multiprocessing, SMP), some configurations of Windows guests crash with an 0x101 error message, indicating a timeout for interprocessor interrupts (IPIs). These interrupts synchronize memory management between processors.

According to Microsoft, this is due to a race condition in Windows. A hotfix is available from Microsoft.

If this does not help, please reduce the number of virtual processors to 1.

5.3.4. Windows 2000 Installation Failures

When installing Windows 2000 guests, you might run into one of the following issues:

- Installation reboots, usually during component registration.

- Installation fills the whole hard disk with empty log files.

- Installation complains about a failure installing `msgina.dll`.

These problems are all caused by a bug in the hard disk driver of Windows 2000. After issuing a hard disk request, there is a race condition in the Windows driver code which leads to corruption if the operation completes too fast. For example, the hardware interrupt from the IDE controller arrives too soon. With physical hardware, there is a guaranteed delay in most systems so the problem is usually hidden there. However, it should be possible to also reproduce it on physical hardware. In a virtual environment, it is possible for the operation to be done immediately, especially on very fast systems with multiple CPUs, and the interrupt is signaled sooner than on a physical system. The solution is to introduce an artificial delay before delivering such interrupts. This delay can be configured for a VM using the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/Devices/piix3ide/0/Config/IRQDelay" 1
```

This sets the delay to one millisecond. In case this does not help, increase it to a value between 1 and 5 milliseconds. Please note that this slows down disk performance. After installation, you should be able to remove the key, or set it to 0.

5.3.5. How to Record Bluescreen Information from Windows Guests

When Windows guests run into a kernel crash, they display a bluescreen error. Depending on how Windows is configured, the information will remain on the screen until the machine is restarted or it will reboot automatically. During installation, Windows is usually configured to reboot automatically. With automatic reboots, there is no chance to record the bluescreen information which might be important for problem determination.

Oracle VM VirtualBox provides a method of halting a guest when it wants to perform a reset. In order to enable this feature, use the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/PDM/HaltOnReset" 1
```

5.3.6. No Networking in Windows Vista Guests

With Windows Vista, Microsoft dropped support for the AMD PCNet card that legacy versions of Oracle VM VirtualBox used to provide as the default virtual network card. For Windows Vista guests, Oracle VM VirtualBox now uses an Intel E1000 card by default.

If, for some reason, you still want to use the AMD card, you need to download the PCNet driver from the AMD website. This driver is available for 32-bit Windows only. You can transfer it into the virtual machine using a shared folder. See Shared Folders.

5.3.7. Windows Guests may Cause a High CPU Load

Several background applications of Windows guests, especially virus scanners, are known to increase the CPU load notably even if the guest appears to be idle. We recommend to deactivate virus scanners within virtualized guests if possible.

5.3.8. Long Delays When Accessing Shared Folders

The performance for accesses to shared folders from a Windows guest might be decreased due to delays during the resolution of the Oracle VM VirtualBox shared folders name service. To fix these delays, add the following entries to the file `\windows\system32\drivers\etc\lmhosts` of the Windows guest:

```
255.255.255.255         VBOXSVR #PRE
255.255.255.255         VBOXSRV #PRE
```

After doing this change, a reboot of the guest is required.

5.3.9. USB Tablet Coordinates Wrong in Windows 98 Guests

If a Windows 98 VM is configured to use the emulated USB tablet (absolute pointing device), the coordinate translation may be incorrect and the pointer is restricted to the upper left quarter of the guest's screen.

The USB HID (Human Interface Device) drivers in Windows 98 are very old and do not handle tablets in the same way as modern operating systems do. To work around the problem, use the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal/USB/HidMouse/0/Config/CoordShift" 0
```

To restore the default behavior, remove the key or set its value to 1.

5.3.10. Windows Guests are Removed From an Active Directory Domain After Restoring a Snapshot

If a Windows guest is a member of an Active Directory domain and the snapshot feature of Oracle VM VirtualBox is used, it could be removed from the Active Direcory domain after you restore an older snapshot.

This is caused by automatic machine password changes performed by Windows at regular intervals for security purposes. You can disable this feature as shown in the following article from Microsoft: http://support.microsoft.com/kb/154501.

5.3.11. Windows 3.x Limited to 64 MB RAM

Windows 3.x guests are typically limited to 64 MB RAM, even if a VM is assigned much more memory. While Windows 3.1 is theoretically capable of using up to 512 MB RAM, it only uses memory available through the XMS interface. Versions of HIMEM.SYS, the Microsoft XMS manager, shipped with MS-DOS and Microsoft Windows 3.x can only use up to 64 MB on standard PCs.

This is a known HIMEM.SYS limitation. Windows 3.1 memory limits are described in detail in Microsoft Knowledge base article KB 84388.

It is possible for Windows 3.x guests to utilize more than 64 MB RAM if a different XMS provider is used. That could be a newer HIMEM.SYS version, such as that shipped with Windows 98, or a more capable third-party memory manager, such as QEMM.

5.4. Linux and X11 Guests

5.4.1. Linux Guests May Cause a High CPU load

Some Linux guests may cause a high CPU load even if the guest system appears to be idle. This can be caused by a high timer frequency of the guest kernel. Some Linux distributions, for example Fedora, ship a Linux kernel configured for a timer frequency of 1000Hz. We recommend to recompile the guest kernel and to select a timer frequency of 100Hz.

Linux kernels shipped with Red Hat Enterprise Linux, as well as kernels of related Linux distributions, such as CentOS and Oracle Linux, support a kernel parameter *divider=N*. Hence, such kernels support a lower timer frequency without recompilation. We suggest you add the kernel parameter *divider=10* to select a guest kernel timer frequency of 100Hz.

5.4.2. Buggy Linux 2.6 Kernel Versions

The following bugs in Linux kernels prevent them from executing correctly in Oracle VM VirtualBox, causing VM boot crashes:

- The Linux kernel version 2.6.18, and some 2.6.17 versions, introduced a race condition that can cause boot crashes in Oracle VM VirtualBox. Please use a kernel version 2.6.19 or later.
- With hardware virtualization and the I/O APIC enabled, kernels before 2.6.24-rc6 may panic on boot with the following message:

```
Kernel panic - not syncing: IO-APIC + timer doesn't work!  Boot with
apic=debug and send a report.  Then try booting with the 'noapic' option
```

If you see this message, either disable hardware virtualization or the I/O APIC as described in System Settings, or upgrade the guest to a newer kernel.

See http://www.mail-archive.com/git-commits-head@vger.kernel.org/msg30813.html for details about the kernel fix.

5.4.3. Shared Clipboard, Auto-Resizing, and Seamless Desktop in X11 Guests

Guest desktop services in guests running the X11 window system such as Oracle Solaris and Linux, are provided by a guest service called `VBoxClient`, which runs under the ID of the user who started the desktop session and is automatically started using the following command lines when your X11 user session is started if you are using a common desktop environment such as Gnome or KDE.

```
$ VBoxClient --clipboard
$ VBoxClient --display
$ VBoxClient --seamless
```

If a particular desktop service is not working correctly, it is worth checking whether the process which should provide it is running.

The `VBoxClient` processes create files in the user's home directory with names of the form `.vboxclient-*.pid` when they are running in order to prevent a given service from being started twice. It can happen due to misconfiguration that these files are created owned by root and not deleted when the services are stopped, which will prevent them from being started in future sessions. If the services cannot be started, you may wish to check whether these files still exist.

5.5. Oracle Solaris Guests

5.5.1. Certain Oracle Solaris 10 Releases May Take a Long Time to Boot with SMP

When using more than one CPU, Oracle Solaris 10 10/08, and Oracle Solaris 10 5/09 may take a long time to boot and may print warnings on the system console regarding failures to read from disk. This is a bug in Oracle Solaris 10 which affects specific physical and virtual configurations. It is caused by trying to read microcode updates from the boot disk when the disk interrupt is reassigned to a not yet fully initialized secondary CPU. Disk reads will time out and fail, triggering delays of about 45 seconds and warnings.

The recommended solution is upgrading to at least Oracle Solaris 10 10/09 which includes a fix for this problem. Alternative solutions include restricting the number of virtual CPUs to one or possibly using a different storage controller.

5.6. Windows Hosts

5.6.1. VBoxSVC Out-of-Process COM Server Issues

Oracle VM VirtualBox makes use of the Microsoft Component Object Model (COM) for interprocess and intraprocess communication. This enables Oracle VM VirtualBox to share a common configuration among different virtual machine processes and provide several user interface options based on a common architecture. All global status information and configuration is maintained by the process `VBoxSVC.exe`, which is an out-of-process COM server. Whenever an Oracle VM VirtualBox process is started, it requests access to the COM server and Windows automatically starts the process. Note that it should never be started by the end user.

When the last process disconnects from the COM server, it will terminate itself after some seconds. The Oracle VM VirtualBox configuration XML files are maintained and owned by the COM server and the files are locked whenever the server runs.

In some cases, such as when a virtual machine is terminated unexpectedly, the COM server will not notice that the client is disconnected and stay active for a longer period of 10 minutes or so, keeping the configuration files locked. In other rare cases the COM server might experience an internal error and subsequently other processes fail to initialize it. In these situations, it is recommended to use the Windows task manager to kill the process `VBoxSVC.exe`.

5.6.2. CD and DVD Changes Not Recognized

In case you have assigned a physical CD or DVD drive to a guest and the guest does not notice when the medium changes, make sure that the Windows media change notification (MCN) feature is not turned off. This is represented by the following key in the Windows registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Cdrom\Autorun
```

Certain applications may disable this key against Microsoft's advice. If it is set to 0, change it to 1 and reboot your system. Oracle VM VirtualBox relies on Windows notifying it of media changes.

5.6.3. Sluggish Response When Using Microsoft RDP Client

If connecting to a Virtual Machine using the Microsoft RDP client, called a Remote Desktop Connection, there can be large delays between input such as moving the mouse over a menu and output. This is because this RDP client collects input for a certain time before sending it to the RDP server.

The interval can be decreased by setting a Windows registry key to smaller values than the default of 100. The key does not exist initially and must be of type DWORD. The unit for its values is milliseconds. Values around 20 are suitable for low-bandwidth connections between the RDP client and server. Values around 4 can be used for a gigabit Ethernet connection. Generally values below 10 achieve a performance that is very close to that of the local input devices and screen of the host on which the Virtual Machine is running.

Depending whether the setting should be changed for an individual user or for the system, set either of the following.

```
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Min Send Interval
```

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Terminal Server Client\Min Send Interval
```

5.6.4. Running an iSCSI Initiator and Target on a Single System

Deadlocks can occur on a Windows host when attempting to access an iSCSI target running in a guest virtual machine with an iSCSI initiator, such as a Microsoft iSCSI Initiator, that is running on the host. This is caused by a flaw in the Windows cache manager component, and causes sluggish host system response for several minutes, followed by a "Delayed Write Failed" error message in the system tray or in a separate message window. The guest is blocked during that period and may show error messages or become unstable.

Setting the `VBOX_DISABLE_HOST_DISK_CACHE` environment variable to `1` enables a workaround for this problem until Microsoft addresses the issue. For example, open a command prompt window and start Oracle VM VirtualBox like this:

```
set VBOX_DISABLE_HOST_DISK_CACHE=1
VirtualBox
```

While this will decrease guest disk performance, especially writes, it does not affect the performance of other applications running on the host.

5.6.5. Bridged Networking Adapters Missing

If no bridged adapters show up in the **Networking** section of the VM settings, this typically means that the bridged networking driver was not installed properly on your host. This could be due to the following reasons:

- The maximum allowed filter count was reached on the host. In this case, the MSI log would mention the `0x8004a029` error code returned on NetFlt network component install, as follows:

  ```
  VBoxNetCfgWinInstallComponent: Install failed, hr (0x8004a029)
  ```

  You can try to increase the maximum filter count in the Windows registry using the following key:

  ```
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Network\MaxNumFilters
  ```

  The maximum number allowed is 14. After a reboot, try to reinstall Oracle VM VirtualBox.

- The INF cache is corrupt. In this case, the install log at `%windir%\inf\setupapi.dev.log` would typically mention the failure to find a suitable driver package for either the `sun_VBoxNetFlt` or `sun_VBoxNetFltmp` components. The solution then is to uninstall Oracle VM VirtualBox, remove the INF cache (`%windir%\inf\INFCACHE.1`), reboot and try to reinstall Oracle VM VirtualBox.

5.6.6. Host-Only Networking Adapters Cannot be Created

If a host-only adapter cannot be created, either with the VirtualBox Manager or the `VBoxManage` command, then the INF cache is probably corrupt. In this case, the install log at `%windir%\inf\setupapi.dev.log` would typically mention the failure to find a suitable driver package for the `sun_VBoxNetAdp` component. Again, as with the bridged networking problem described above, the solution is to uninstall Oracle VM VirtualBox, remove the INF cache (`%windir%\inf\INFCACHE.1`), reboot and try to reinstall Oracle VM VirtualBox.

5.7. Linux Hosts

5.7.1. Linux Kernel Module Refuses to Load

If the Oracle VM VirtualBox kernel module, `vboxdrv`, refuses to load you may see an `Error inserting vboxdrv: Invalid argument` message. As root, check the output of the `dmesg` command to find out why the load failed. Most probably the kernel disagrees with the version of `gcc` used to compile the module. Make sure that you use the same compiler that was used to build the kernel.

5.7.2. Linux Host CD or DVD Drive Not Found

If you have configured a virtual machine to use the host's CD or DVD drive, but this does not appear to work, make sure that the current user has permission to access the corresponding Linux device file. This is `/dev/hdc`, `/dev/scd0`, `/dev/cdrom` or similar. On most distributions, the user must be added to a corresponding group, usually called `cdrom` or `cdrw`.

5.7.3. Linux Host CD or DVD Drive Not Found (Older Distributions)

On older Linux distributions, if your CD or DVD device has a different name, Oracle VM VirtualBox may be unable to find it. On older Linux hosts, Oracle VM VirtualBox performs the following steps to locate your CD or DVD drives:

1. Oracle VM VirtualBox checks if the environment variable `VBOX_CDROM` is defined. If so, Oracle VM VirtualBox omits all the following checks.
2. Oracle VM VirtualBox tests if `/dev/cdrom` works.
3. Oracle VM VirtualBox checks if any CD or DVD drives are currently mounted by checking `/etc/mtab`.
4. Oracle VM VirtualBox checks if any of the entries in `/etc/fstab` point to CD or DVD devices.

You can set the VBOX_CDROM environment variable to contain a list of your CD or DVD devices, separated by colons. For example:

```
export VBOX_CDROM='/dev/cdrom0:/dev/cdrom1'
```

On modern Linux distributions, Oracle VM VirtualBox uses the hardware abstraction layer (HAL) to locate CD and DVD hardware.

5.7.4. Linux Host Floppy Not Found

Section 5.7.3, “Linux Host CD or DVD Drive Not Found (Older Distributions)” appplies also to floppy disks, except that on older distributions Oracle VM VirtualBox tests for `/dev/fd*` devices by default. This can be overridden with the `VBOX_FLOPPY` environment variable.

5.7.5. Strange Guest IDE Error Messages When Writing to CD or DVD

If the experimental CD or DVD writer support is enabled with an incorrect Oracle VM VirtualBox, host or guest configuration, it is possible that any attempt to access the CD or DVD writer fails and simply results in guest kernel error messages for Linux guests or application error messages for Windows guests. Oracle VM VirtualBox performs the usual consistency checks when a VM is powered up. In particular, it aborts with an error message if the device for the CD or DVD writer is not writable by the user starting the VM. But Oracle VM VirtualBox cannot detect all misconfigurations. The necessary host and guest OS configuration is not specific for Oracle VM VirtualBox, but a few frequent problems are listed here which occurred in connection with Oracle VM VirtualBox.

Special care must be taken to use the correct device. The configured host CD or DVD device file name, in most cases `/dev/cdrom`, must point to the device that allows writing to the CD or DVD unit. For CD or DVD writer units connected to a SCSI controller or to a IDE controller that interfaces to the Linux SCSI subsystem, common for some SATA controllers, this must refer to the SCSI device node, such as `/dev/scd0`. Even for IDE CD or DVD writer units this must refer to the appropriate SCSI CD-ROM device node, such as `/dev/scd0`, if the `ide-scsi` kernel module is loaded. This module is required for CD or DVD writer support with some early 2.6 kernels. Many Linux distributions load this module whenever a CD or DVD writer is detected in the system, even if the kernel would support CD or DVD writers without the module. Oracle VM VirtualBox supports the use of IDE device files, such as `/dev/hdc`, provided the kernel supports this and the `ide-scsi` module is not loaded.

Similar rules, except that within the guest the CD or DVD writer is always an IDE device, apply to the guest configuration. Since this setup is very common, it is likely that the default configuration of the guest works as expected.

5.7.6. VBoxSVC IPC Issues

On Linux, Oracle VM VirtualBox makes use of a custom version of Mozilla XPCOM (cross platform component object model) for interprocess and intraprocess communication (IPC). The process `VBoxSVC` serves as a communication hub between different Oracle VM VirtualBox processes and maintains the global configuration, such as the XML database. When starting an Oracle VM VirtualBox component, the

processes VBoxSVC and VBoxXPCOMIPCD are started automatically. They are only accessible from the user account they are running under. VBoxSVC owns the Oracle VM VirtualBox configuration database which normally resides in ~/.config/VirtualBox, or the appropriate configuration directory for your operating system. While it is running, the configuration files are locked. Communication between the various Oracle VM VirtualBox components and VBoxSVC is performed through a local domain socket residing in /tmp/.vbox-*username*-ipc. In case there are communication problems, such as an Oracle VM VirtualBox application cannot communicate with VBoxSVC, terminate the daemons and remove the local domain socket directory.

5.7.7. USB Not Working

If USB is not working on your Linux host, make sure that the current user is a member of the vboxusers group. Please keep in mind that group membership does not take effect immediately but rather at the next login. If available, the newgrp command may avoid the need for a logout and login.

5.7.8. PAX/grsec Kernels

Linux kernels including the grsec patch, see http://www.grsecurity.net/, and derivates have to disable PAX_MPROTECT for the VBox binaries to be able to start a VM. The reason is that Oracle VM VirtualBox has to create executable code on anonymous memory.

5.7.9. Linux Kernel vmalloc Pool Exhausted

When running a large number of VMs with a lot of RAM on a Linux system, say 20 VMs with 1 GB of RAM each, additional VMs might fail to start with a kernel error saying that the vmalloc pool is exhausted and should be extended. The error message also tells you to specify vmalloc=256MB in your kernel parameter list. If adding this parameter to your GRUB or LILO configuration makes the kernel fail to boot, with an error message such as failed to mount the root partition, then you have probably run into a memory conflict of your kernel and initial RAM disk. This can be solved by adding the following parameter to your GRUB configuration:

```
uppermem 524288
```

5.8. Oracle Solaris Hosts

5.8.1. Cannot Start VM, Not Enough Contiguous Memory

The ZFS file system is known to use nearly all available RAM as cache if the default system settings are not changed. This may lead to a heavy fragmentation of the host memory preventing Oracle VM VirtualBox VMs from being started. We recommend to limit the ZFS cache by adding the following line to /etc/system, where *xxxx* bytes is the amount of memory usable for the ZFS cache.

```
set zfs:zfs_arc_max = xxxx
```

Chapter 6 Security Guide

6.1. General Security Principles

The following principles are fundamental to using any application securely.

- **Keep software up to date**. One of the principles of good security practise is to keep all software versions and patches up to date. Activate the Oracle VM VirtualBox update notification to get notified when a new Oracle VM VirtualBox release is available. When updating Oracle VM VirtualBox, do not forget to update the Guest Additions. Keep the host operating system as well as the guest operating system up to date.
- **Restrict network access to critical services.** Use proper means, for instance a firewall, to protect your computer and your guests from accesses from the outside. Choosing the proper networking mode for VMs helps to separate host networking from the guest and vice versa.
- **Follow the principle of least privilege.** The principle of least privilege states that users should be given the least amount of privilege necessary to perform their jobs. Always execute Oracle VM VirtualBox as a regular user. We strongly discourage anyone from executing Oracle VM VirtualBox with system privileges.

  Choose restrictive permissions when creating configuration files, for instance when creating /etc/default/virtualbox, see Automatic Installation Options. Mode 0600 is preferred.
- **Monitor system activity.** System security builds on three pillars: good security protocols, proper system configuration and system monitoring. Auditing and reviewing audit records address the third requirement. Each component within a system has some degree of monitoring capability. Follow audit advice in this document and regularly monitor audit records.
- **Keep up to date on latest security information.** Oracle continually improves its software and documentation. Check this note yearly for revisions.

6.2. Secure Installation and Configuration

6.2.1. Installation Overview

The Oracle VM VirtualBox base package should be downloaded only from a trusted source, for instance the official website http://www.virtualbox.org. The integrity of the package should be verified with the provided SHA256 checksum which can be found on the official website.

General Oracle VM VirtualBox installation instructions for the supported hosts can be found in Installation Details.

On Windows hosts, the installer can be used to disable USB support, support for bridged networking, support for host-only networking and the Python language binding. See Installing on Windows Hosts. All these features are enabled by default but disabling some of them could be appropriate if the corresponding functionality is not required by any virtual machine. The Python language bindings are only required if the Oracle VM VirtualBox API is to be used by external Python applications. In particular USB support and support for the two networking modes require the installation of Windows kernel drivers on the host. Therefore disabling those selected features can not only be used to restrict the user to certain functionality but also to minimize the surface provided to a potential attacker.

The general case is to install the complete Oracle VM VirtualBox package. The installation must be done with system privileges. All Oracle VM VirtualBox binaries should be executed as a regular user and never as a privileged user.

The Oracle VM VirtualBox Extension Pack provides additional features and must be downloaded and installed separately, see Installing Oracle VM VirtualBox and Extension Packs. As for the base package, the SHA256 checksum of the extension pack should be verified. As the installation requires system privileges, Oracle VM VirtualBox will ask for the system password during the installation of the extension pack.

6.2.2. Post Installation Configuration

Normally there is no post installation configuration of Oracle VM VirtualBox components required. However, on Oracle Solaris and Linux hosts it is necessary to configure the proper permissions for users executing VMs and who should be able to access certain host resources. For instance, Linux users must be member of the *vboxusers* group to be able to pass USB devices to a guest. If a serial host interface should be accessed from a VM, the proper permissions must be granted to the user to be able to access that device. The same applies to other resources like raw partitions, DVD/CD drives, and sound devices.

6.3. Security Features

This section outlines the specific security mechanisms offered by Oracle VM VirtualBox.

6.3.1. The Security Model

One property of virtual machine monitors (VMMs) like Oracle VM VirtualBox is to encapsulate a guest by executing it in a protected environment, a virtual machine, running as a user process on the host operating system. The guest cannot communicate directly with the hardware or other computers but only through the VMM. The VMM provides emulated physical resources and devices to the guest which are accessed by the guest operating system to perform the required tasks. The VM settings control the resources provided to the guest, for example the amount of guest memory or the number of guest processors and the enabled features for that guest. For example remote control, certain screen settings and others. See General Settings.

6.3.2. Secure Configuration of Virtual Machines

Several aspects of a virtual machine configuration are subject to security considerations.

6.3.2.1. Networking

The default networking mode for VMs is NAT which means that the VM acts like a computer behind a router, see Network Address Translation (NAT). The guest is part of a private subnet belonging to this VM and the guest IP is not visible from the outside. This networking mode works without any additional setup and is sufficient for many purposes.

If bridged networking is used, the VM acts like a computer inside the same network as the host, see Bridged Networking. In this case, the guest has the same network access as the host and a firewall might be necessary to protect other computers on the subnet from a potential malicious guest as well as to protect the guest from a direct access from other computers. In some cases it is worth considering using a forwarding rule for a specific port in NAT mode instead of using bridged networking.

Some setups do not require a VM to be connected to the public network at all. Internal networking, see Internal Networking, or host-only networking, see Host-Only Networking, are often sufficient to connect VMs among each other or to connect VMs only with the host but not with the public network.

6.3.2.2. VRDP Remote Desktop Authentication

When using the Oracle VM VirtualBox Extension Pack provided by Oracle for VRDP remote desktop support, you can optionally use various methods to configure RDP authentication. The "null" method is very insecure and should be avoided in a public network. See Section 1.1.5, “RDP Authentication”.

6.3.2.3. Clipboard

The shared clipboard enables users to share data between the host and the guest. Enabling the clipboard in Bidirectional mode enables the guest to read and write the host clipboard. The Host to Guest mode and the Guest to Host mode limit the access to one direction. If the guest is able to access the host clipboard it can also potentially access sensitive data from the host which is shared over the clipboard.

If the guest is able to read from and/or write to the host clipboard then a remote user connecting to the guest over the network will also gain this ability, which may not be desirable. As a consequence, the shared clipboard is disabled for new machines.

6.3.2.4. Shared Folders

If any host folder is shared with the guest then a remote user connected to the guest over the network can access these files too as the folder sharing mechanism cannot be selectively disabled for remote users.

6.3.2.5. 3D Graphics Acceleration

Enabling 3D graphics using the Guest Additions exposes the host to additional security risks. See Hardware 3D Acceleration (OpenGL and Direct3D 8/9).

6.3.2.6. CD/DVD Passthrough

Enabling CD/DVD passthrough enables the guest to perform advanced operations on the CD/DVD drive, see CD/DVD Support. This could induce a security risk as a guest could overwrite data on a CD/DVD medium.

6.3.2.7. USB Passthrough

Passing USB devices to the guest provides the guest full access to these devices, see USB Settings. For instance, in addition to reading and writing the content of the partitions of an external USB disk the guest will be also able to read and write the partition table and hardware data of that disk.

6.3.3. Configuring and Using Authentication

The following components of Oracle VM VirtualBox can use passwords for authentication:

- When using remote iSCSI storage and the storage server requires authentication, an initiator secret can optionally be supplied with the `VBoxManage storageattach` command. As long as no settings password is provided, by using the command line option `--settingspwfile`, then this secret is stored *unencrypted* in the machine configuration and is therefore potentially readable on the host. See iSCSI Servers and VBoxManage storageattach.
- When using the Oracle VM VirtualBox web service to control an Oracle VM VirtualBox host remotely, connections to the web service are authenticated in various ways. This is described in detail in the Oracle VM VirtualBox Software Development Kit (SDK) reference. See Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.

6.3.4. Potentially Insecure Operations

The following features of Oracle VM VirtualBox can present security problems:

- Enabling 3D graphics using the Guest Additions exposes the host to additional security risks. See Hardware 3D Acceleration (OpenGL and Direct3D 8/9).

- When teleporting a machine, the data stream through which the machine's memory contents are transferred from one host to another is not encrypted. A third party with access to the network through which the data is transferred could therefore intercept that data. An SSH tunnel could be used to secure the connection between the two hosts. But when considering teleporting a VM over an untrusted network the first question to answer is how both VMs can securely access the same virtual disk image with a reasonable performance.
- When Page Fusion, see Page Fusion, is enabled, it is possible that a side-channel opens up that enables a malicious guest to determine the address space of another VM running on the same host layout. For example, where DLLs are typically loaded. This information leak in itself is harmless, however the malicious guest may use it to optimize attack against that VM through unrelated attack vectors. It is recommended to only enable Page Fusion if you do not think this is a concern in your setup.
- When using the Oracle VM VirtualBox web service to control an Oracle VM VirtualBox host remotely, connections to the web service, over which the API calls are transferred using SOAP XML, are not encrypted. They use plain HTTP by default. This is a potential security risk. For details about the web service, see Chapter 4, *Oracle VM VirtualBox Programming Interfaces*.

  The web services are not started by default. See Section 2.19, “Starting the Oracle VM VirtualBox Web Service Automatically” to find out how to start this service and how to enable SSL/TLS support. It has to be started as a regular user and only the VMs of that user can be controlled. By default, the service binds to localhost preventing any remote connection.
- Traffic sent over a UDP Tunnel network attachment is not encrypted. You can either encrypt it on the host network level, with IPsec, or use encrypted protocols in the guest network, such as SSH. The security properties are similar to bridged Ethernet.
- Because of shortcomings in older Windows versions, using Oracle VM VirtualBox on Windows versions older than Vista with Service Pack 1 is not recommended.

6.3.5. Encryption

The following components of Oracle VM VirtualBox use encryption to protect sensitive data:

- When using the Oracle VM VirtualBox Extension Pack provided by Oracle for VRDP remote desktop support, RDP data can optionally be encrypted. See Section 1.1.6, “RDP Encryption”. Only the Enhanced RDP Security method (RDP5.2) with TLS protocol provides a secure connection. Standard RDP Security (RDP4 and RDP5.1) is vulnerable to a man-in-the-middle attack.

6.4. Security Recommendations

This section contains security recommendations for specific issues. By default VirtualBox will configure the VMs to run in a secure manner, however this may not always be possible without additional user actions (e.g. host OS / firmware configuration changes).

6.4.1. CVE-2018-3646

This security issue affect a range of Intel CPUs with nested paging. AMD CPUs are expected not to be impacted (pending direct confirmation by AMD). Also the issue does not affect VMs running with hardware virtualization disabled or with nested paging disabled.

For more information about nested paging, see Section 3.7, “Nested Paging and VPIDs”.

The following mitigation options are available.

6.4.1.1. Disable Nested Paging

By disabling nested paging (EPT), the VMM will construct page tables shadowing the ones in the guest. It is no possible for the guest to insert anything fishy into the page tables, since the VMM carefully validates each entry before shadowing it.

As a side effect of disabling nested paging, several CPU features will not be made available to the guest. Among these features are AVX, AVX2, XSAVE, AESNI, and POPCNT. Not all guests may be able to cope with dropping these features after installation. Also, for some guests, especially in SMP configurations, there could be stability issues arising from disabling nested paging. Finally, some workloads may experience a performance degradation.

6.4.1.2. Flushing the Level 1 Data Cache

This aims at removing potentially sensitive data from the level 1 data cache when running guest code. However, it is made difficult by hyper-threading setups sharing the level 1 cache and thereby potentially letting the other thread in a pair refill the cache with data the user does not want the guest to see. In addition, flushing the level 1 data cache is usually not without performance side effects.

Up to date CPU microcode is a prerequisite for the cache flushing mitigations. Some host OSes may install these automatically, though it has traditionally been a task best performed by the system firmware. So, please check with your system / mainboard manufacturer for the latest firmware update.

We recommend disabling hyper threading on the host. This is traditionally done from the firmware setup, but some OSes also offers ways disable HT. In some cases it may be disabled by default, but please verify as the effectiveness of the mitigation depends on it.

The default action taken by VirtualBox is to flush the level 1 data cache when a thread is scheduled to execute guest code, rather than on each VM entry. This reduces the performance impact, while making the assumption that the host OS will not handle security sensitive data from interrupt handlers and similar without taking precautions.

A more aggressive flushing option is provided via the `VBoxManage modifyvm --l1d-flush-on-vm-entry` option. When enabled the level 1 data cache will be flushed on every VM entry. The performance impact is greater than with the default option, though this of course depends on the workload. Workloads producing a lot of VM exits (like networking, VGA access, and similiar) will probably be most impacted.

For users not concerned by this security issue, the default mitigation can be disabled using the `VBoxManage modifyvm name --l1d-flush-on-sched off` command.

6.4.2. CVE-2018-12126, CVE-2018-12127, CVE-2018-12130, CVE-2019-11091

These security issues affect a range of Intel CPUs starting with Nehalem. The CVE-2018-12130 also affects some Atom Silvermont, Atom Airmont, and Knights family CPUs, however the scope is so limited that the host OS should deal with it and Oracle VM VirtualBox is therefore not affected. Leaks only happens when entering and leaving C states.

The following mitigation option is available.

6.4.2.1. Buffer Overwriting and Disabling Hyper-Threading

First, up to date CPU microcode is a prerequisite for the buffer overwriting (clearing) mitigations. Some host OSes may install these automatically, though it has traditionally been a task best performed by the system firmware. Please check with your system or mainboard manufacturer for the latest firmware update.

This mitigation aims at removing potentially sensitive data from the affected buffers before running guest code. Since this means additional work each time the guest is scheduled, there might be some performance side effects.

We recommend disabling hyper-threading (HT) on hosts affected by CVE-2018-12126 and CVE-2018-12127, because the affected sets of buffers are normally shared between thread pairs and therefore cause leaks between the threads. This is traditionally done from the firmware setup, but some OSes also offers ways disable HT. In some cases it may be disabled by default, but please verify as the effectiveness of the mitigation depends on it.

The default action taken by Oracle VM VirtualBox is to clear the affected buffers when a thread is scheduled to execute guest code, rather than on each VM entry. This reduces the performance impact, while making the assumption that the host OS will not handle security sensitive data from interrupt handlers and similar without taking precautions.

The `VBoxManage modifyvm` command provides a more aggressive flushing option is provided by means of the `--mds-clear-on-vm-entry` option. When enabled the affected buffers will be cleared on every VM entry. The performance impact is greater than with the default option, though this of course depends on the workload. Workloads producing a lot of VM exits (like networking, VGA access, and similiar) will probably be most impacted.

For users not concerned by this security issue, the default mitigation can be disabled using the `VBoxManage modifyvm name --mds-clear-on-sched off` command.

Chapter 7 Known Limitations

7.1. Experimental Features

Some Oracle VM VirtualBox features are labeled as experimental. Such features are provided on an "as-is" basis and are not formally supported. However, feedback and suggestions about such features are welcome. A comprehensive list of experimental features is as follows:

- Hardware 3D acceleration support for Windows, Linux, and Oracle Solaris guests
- Hardware 2D video playback acceleration support for Windows guests
- PCI pass-through (Linux hosts only)
- Mac OS X guests (Mac OS X hosts only)
- ICH9 chipset emulation
- EFI firmware
- Host CD/DVD drive pass-through
- Support of iSCSI using internal networking
- Using Oracle VM VirtualBox and Hyper-V on the same host

7.2. Known Issues

The following section describes known problems with this release of Oracle VM VirtualBox. Unless marked otherwise, these issues are planned to be fixed in later releases.

- The following **Guest SMP (multiprocessor) limitations** exist:
  - **Poor performance** with 32-bit guests on AMD CPUs. This affects mainly Windows and Oracle Solaris guests, but possibly also some Linux kernel revisions. Partially solved for 32-bit Windows NT, 2000, XP, and 2003 guests. Requires the Guest Additions to be installed.
  - **Poor performance** with 32-bit guests on certain Intel CPU models that do not include virtual APIC hardware optimization support. This affects mainly Windows and Oracle Solaris guests, but possibly also some Linux kernel revisions. Partially solved for 32-bit Windows NT, 2000, XP, and 2003 guests. Requires the Guest Additions to be installed.
- **NX (no execute, data execution prevention)** only works for guests running on 64-bit hosts and requires that hardware virtualization be enabled.
- **Guest control.** On Windows guests, a process started using the guest control execute support will not be able to display a graphical user interface *unless* the user account under which it is running is currently logged in and has a desktop session.

  Also, to use accounts without or with an empty password, the guest's group policy must be changed. To do so, open the group policy editor on the command line by typing `gpedit.msc`, open the key `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options` and change the value of `Accounts: Limit local account use of blank passwords to console logon only` to Disabled.
- **Compacting virtual disk images is limited to VDI files.** The `VBoxManage modifymedium --compact` command is currently only implemented for VDI files. At the moment the only way to optimize

the size of a virtual disk images in other formats, such as VMDK or VHD, is to clone the image and then use the cloned image in the VM configuration.

- **OVF import/export:**
  - OVF localization, with multiple languages in a single OVF file, is not yet supported.
  - Some OVF sections like StartupSection, DeploymentOptionSection, and InstallSection are ignored.
  - OVF environment documents, including their property sections and appliance configuration with ISO images, are not yet supported.
  - Remote files using HTTP or other mechanisms are not yet supported.
- Neither **scale mode** nor **seamless mode** work correctly with guests using OpenGL 3D features, such as with Compiz-enabled window managers.
- The RDP server in the Oracle VM VirtualBox extension pack supports only audio streams in format 22.05kHz stereo 16-bit. If the RDP client requests any other audio format there will be no audio.
- Preserving the aspect ratio in scale mode works only on Windows hosts and on Mac OS X hosts.
- On **Mac OS X hosts,** the following features are not yet implemented:
  - Numlock emulation
  - CPU frequency metric
  - Memory ballooning
- **Mac OS X guests:**
  - Mac OS X guests can only run on a certain host hardware. For details about license and host hardware limitations. See Mac OS X Guests and check the Apple software license conditions.
  - Oracle VM VirtualBox does not provide Guest Additions for Mac OS X at this time.
  - The graphics resolution currently defaults to 1024x768 as Mac OS X falls back to the built-in EFI display support. See Video Modes in EFI for more information on how to change EFI video modes.
  - Mac OS X guests only work with one CPU assigned to the VM. Support for SMP will be provided in a future release.
  - Depending on your system and version of Mac OS X, you might experience guest hangs after some time. This can be fixed by turning off energy saving. Set the timeout to "Never" in the system preferences.
  - By default, the Oracle VM VirtualBox EFI enables debug output of the Mac OS X kernel to help you diagnose boot problems. Note that there is a lot of output and not all errors are fatal. They would also

show when using a physical Apple Macintosh computer. You can turn off these messages by using the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal2/EfiBootArgs" "  "
```

To revert to the previous behavior, use the following command:

```
$ VBoxManage setextradata VM-name "VBoxInternal2/EfiBootArgs" ""
```

- It is currently not possible to start a Mac OS X guest in safe mode by specifying the `-x` option in `VBoxInternal2/EfiBootArgs` extradata.

- **Oracle Solaris hosts:**
  - USB support on Oracle Solaris hosts requires Oracle Solaris 11 version snv_124 or later. Webcams and other isochronous devices are known to have poor performance.
  - Host Webcam passthrough is restricted to 640x480 frames at 20 frames per second due to limitations in the Oracle Solaris V4L2 API. This may be addressed in a future Oracle Solaris release.
  - No ACPI information, such as battery status or power source, is reported to the guest.
  - No support for using wireless adapters with bridged networking.
  - Crossbow-based bridged networking on Oracle Solaris 11 hosts does not work directly with aggregate links. However, you can use `dladm` to manually create a VNIC over the aggregate link and use that with a VM. This limitation does not exist in Oracle Solaris 11u1 build 17 and later.
- Neither virtio nor Intel PRO/1000 drivers for **Windows XP guests** support segmentation offloading. Therefore Windows XP guests have slower transmission rates comparing to other guest types. Refer to MS Knowledge base article 842264 for additional information.
- **Guest Additions for OS/2.** Seamless windows and automatic guest resizing will probably never be implemented due to inherent limitations of the OS/2 graphics system.
- Some guest operating systems predating ATAPI CD-ROMs may exhibit long delays or entirely fail to boot in certain configurations. This is most likely to happen when an IDE/ATAPI CD-ROM exists alone on a primary or secondary IDE channel.

  Affected operating systems are MS OS/2 1.21: fails to boot with an error message referencing COUNTRY.SYS and MS OS/2 1.3: long boot delays. To avoid such problems, disable the emulated IDE/ATAPI CD-ROM. The guest OS cannot use this device, anyway.

Appendix A Third-Party Materials and Licenses

Oracle VM VirtualBox incorporates materials from several Open Source software projects. Therefore the use of these materials by Oracle VM VirtualBox is governed by different Open Source licenses. This document reproduces these licenses and provides a list of the materials used and their respective licensing conditions. Section 1 contains a list of the materials used. Section 2 reproduces the applicable Open Source licenses. For each material, a reference to its license is provided.

The source code for the materials listed below as well as the rest of the Oracle VM VirtualBox code which is released as open source are available at http://www.virtualbox.org, both as tarballs for particular releases and as a live SVN repository.

A.1. Third-Party Materials

- Oracle VM VirtualBox contains portions of QEMU which is governed by the licenses in A.2.5 and A.2.2 and

  (C) 2003-2005 Fabrice Bellard; Copyright (C) 2004-2005 Vassili Karpov (malc); Copyright (c) 2004 Antony T Curtis; Copyright (C) 2003 Jocelyn Mayer

- Oracle VM VirtualBox contains code which is governed by the license in A.2.5 and

  Copyright 2004 by the Massachusetts Institute of Technology.

- Oracle VM VirtualBox contains code of the BOCHS VGA BIOS which is governed by the license in A.2.2 and

  Copyright (C) 2001, 2002 the LGPL VGABios developers Team.

- Oracle VM VirtualBox contains code of the BOCHS ROM BIOS which is governed by the license in A.2.2 and

  Copyright (C) 2002 MandrakeSoft S.A.; Copyright (C) 2004 Fabrice Bellard; Copyright (C) 2005 Struan Bartlett.

- Oracle VM VirtualBox contains the zlib library which is governed by the license in A.2.6 and

  Copyright (C) 1995-2003 Jean-loup Gailly and Mark Adler.

- Oracle VM VirtualBox may contain OpenSSL which is governed by the license in A.2.7 and

  Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

- Oracle VM VirtualBox may contain NSPR and XPCOM which is governed by the license in A.2.3 and

  Copyright (C) The Authors.

- Oracle VM VirtualBox contains Slirp which is governed by the license in A.2.8 and was written by Danny Gasparovski.

  Copyright (C) 1995, 1996 All Rights Reserved.

- Oracle VM VirtualBox contains liblzf which is governed by the license in A.2.9 and

  Copyright (C) 2000-2005 Marc Alexander Lehmann <schmorp@schmorp.de>

- Oracle VM VirtualBox may ship with a modified copy of rdesktop which is governed by the license in A.2.1 and

  Copyright (C) Matthew Chapman and others.

- Oracle VM VirtualBox may ship with a copy of kchmviewer which is governed by the license in A.2.1 and

  Copyright (C) George Yunaev and others.

- Oracle VM VirtualBox may contain Etherboot which is governed by the license in A.2.1 with the exception that aggregating Etherboot with another work does not require the other work to be released under the same license (see http://etherboot.sourceforge.net/clinks.html). Etherboot is

  Copyright (C) Etherboot team.

- Oracle VM VirtualBox may contain iPXE which is governed by the license in A.2.1 and

  Copyright (C) Michael Brown <mbrown@fensystems.co.uk> and others.

- Oracle VM VirtualBox contains code from Wine which is governed by the license in A.2.2 and

  Copyright 1993 Bob Amstadt, Copyright 1996 Albrecht Kleine, Copyright 1997 David Faure, Copyright 1998 Morten Welinder, Copyright 1998 Ulrich Weigand, Copyright 1999 Ove Koven

- Oracle VM VirtualBox contains code from lwIP which is governed by the license in A.2.11 and

  Copyright (C) 2001, 2002 Swedish Institute of Computer Science.

- Oracle VM VirtualBox contains libxml which is governed by the license in A.2.12 and

  Copyright (C) 1998-2003 Daniel Veillard.

- Oracle VM VirtualBox contains libxslt which is governed by the license in A.2.13 and

  Copyright (C) 2001-2002 Daniel Veillard and Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard.

- Oracle VM VirtualBox contains code from the gSOAP XML web services tools, which are licensed under the license in A.2.14 and

  Copyright (C) 2000-2007, Robert van Engelen, Genivia Inc., and others.

- Oracle VM VirtualBox ships with the application tunctl (shipped as VBoxTunctl) from the User-mode Linux suite which is governed by the license in A.2.1 and

  Copyright (C) 2002 Jeff Dike.

- Oracle VM VirtualBox contains code from Chromium, an OpenGL implementation, which is goverened by the licenses in A.2.15 and

  Copyright (C) Stanford University, The Regents of the University of California, Red Hat, and others.

- Oracle VM VirtualBox contains libcurl which is governed by the license in A.2.16 and

  Copyright (C) 1996-2009, Daniel Stenberg.

- Oracle VM VirtualBox contains dnsproxy which is governed by the license in A.2.4 and

  Copyright (c) 2003, 2004, 2005 Armin Wolfermann.

- Oracle VM VirtualBox may contain iniparser which is governed by the license in A.2.4 and

  Copyright (c) 2000-2008 by Nicolas Devillard.

- Oracle VM VirtualBox contains some code from libgd which is governed by the license in A.2.17 and

  Copyright 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Pierre-Alain Joye (pierre@libgd.org).

- Oracle VM VirtualBox contains code from the EFI Development Kit II which is governed by the license in A.2.18 and

  Copyright (c) 2004-2008, Intel Corporation.

- Oracle VM VirtualBox contains libjpeg which is governed by the license in A.2.19 and

  Copyright (C) 1991-2010, Thomas G. Lane, Guido Vollbeding.

- Oracle VM VirtualBox may contain x86 SIMD extension for IJG JPEG library which is governed by the license in A.2.20 and

  Copyright 2009 Pierre Ossman <ossman@cendio.se> for Cendio AB; Copyright 2010 D. R. Commander; Copyright (C) 1999-2006, MIYASAKA Masaru.

- Oracle VM VirtualBox may ship a copy of Qt which is governed by the license in A.2.2 and

  Copyright (C) 2010, 2011 Nokia Corporation and/or its subsidiary(-ies).

- Oracle VM VirtualBox contains parts of the FreeBSD kernel which is governed by the license in A.2.21.

- Oracle VM VirtualBox contains parts of the NetBSD kernel which is governed by the license in A.2.22.

- Oracle VM VirtualBox contains portions of liblightdm-gobject which is governed by the license in A.2.2 and

  Copyright (C) 2010-2013 Canonical Ltd.; Copyright (C) 2010-2011 Robert Ancell.

- Oracle VM VirtualBox contains portions of glib which is governed by the license in A.2.2 and

  Copyright (C) 1995-2011 The Glib team

- Oracle VM VirtualBox contains portions of PCRE which is governed by the license in A.2.23 and

  Copyright (c) 1997-2012 University of Cambridge; Copyright(c) 2009-2012 Zoltan Herczeg; Copyright (c) 2007-2012, Google Inc.

- Oracle VM VirtualBox contains portions of libffi which is governed by the license in A.2.24 and

  Copyright (c) 1996-2012 Anthony Green, Red Hat, Inc and others. See source files for details.

- Oracle VM VirtualBox contains portions of FLTK which is governed by the licenses in A.2.25 and A.2.2 and

  Copyright (C) 1991-2012 The FLTK team

- Oracle VM VirtualBox contains portions of Expat which is governed by the license in A.2.26 and

  Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper; Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

- Oracle VM VirtualBox contains portions of Fontconfig which is governed by the license in A.2.27 and

Copyright (C) 2001, 2003 Keith Packard

- Oracle VM VirtualBox contains portions of Freetype which is governed by the license in A.2.28 and

  Copyright 2012 The FreeType Project (www.freetype.org). All rights reserved.

- Oracle VM VirtualBox may contain code from the WebM VP8 Codec SDK which is governed by the license in A.2.29 and

  Copyright (c) 2010, The WebM Project authors. All rights reserved.

- Oracle VM VirtualBox may contain code from libopus ("Opus") which is governed by the license in A.2.30 and

  Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic, Jean-Marc Valin, Timothy B. Terriberry, CSIRO, Gregory Maxwell, Mark Borgerding, Erik de Castro Lopo

- Oracle VM VirtualBox may contain portions of FUSE for macOS which is governed by the licenses in A.2.31 and A.2.2 and

  Copyright (c) 2011-2017 Benjamin Fleischer; Copyright (c) 2011-2012 Erik Larsson All rights reserved.

A.2. Third-Party Licenses

A.2.1. GNU General Public License (GPL)

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these

terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

A.2.2. GNU Lesser General Public License (LGPL)

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is

not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or

translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.) Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then

relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

A.2.3. Mozilla Public License (MPL)

MOZILLA PUBLIC LICENSE Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that

Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License. This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions. Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions. Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works. If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY

AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS. The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE. Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____. Portions created by _____ are Copyright (C) _____ _____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

A.2.4. MIT License

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

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

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

A.2.5. X Consortium License (X11)

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

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

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

A.2.6. zlib License

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

```
Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu
```

A.2.7. OpenSSL License

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

A.2.8. Slirp License

Copyright (c) 1995,1996 Danny Gasparovski. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by Danny Gasparovski.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DANNY GASPAROVSKI OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.9. liblzf License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.10. libpng License

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A.2.11. lwIP License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.12. libxml License

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

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

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

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

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

A.2.13. libxslt Licenses

Licence for libxslt except libexslt:

Copyright (C) 2001-2002 Daniel Veillard. All Rights Reserved.

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

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

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

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

Licence for libexslt:

Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard. All Rights Reserved.

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

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

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

Except as contained in this notice, the name of the authors shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

A.2.14. gSOAP Public License Version 1.3a

The gSOAP public license is derived from the Mozilla Public License (MPL1.1). The sections that were deleted from the original MPL1.1 text are 1.0.1, 2.1.(c),(d), 2.2.(c),(d), 8.2.(b), 10, and 11. Section 3.8 was

added. The modified sections are 2.1.(b), 2.2.(b), 3.2 (simplified), 3.5 (deleted the last sentence), and 3.6 (simplified).

1 DEFINITIONS

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code, or Modifications or the combination of the Original Code, and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code, or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2 SOURCE CODE LICENSE.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell ("offer to sell and import") the Original Code, Modifications, or portions thereof, but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Original Code, Modifications, or any combination or portions thereof.

(c)

(d)

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to make, have made, use and sell ("offer to sell and import") the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Contributor Version (or portions thereof).

(c)

(d)

3 DISTRIBUTION OBLIGATIONS.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification created by You will be provided to the Initial Developer in Source Code form and are subject to the terms of the License. 3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent

statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters.

(a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations. Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. If you distribute executable versions containing Covered Code as part of a product, you must reproduce the notice in Exhibit B in the documentation and/or other materials provided with the product.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

3.8. Restrictions. You may not remove any product identification, copyright, proprietary notices or labels from gSOAP.

4 INABILITY TO COMPLY DUE TO STATUTE OR REGULATION.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of

this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5 APPLICATION OF THIS LICENSE.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6 VERSIONS OF THE LICENSE.

6.1. New Versions.

Grantor may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrase "gSOAP" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the gSOAP Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7 DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND ANY WARRANTY THAT MAY ARISE BY REASON OF TRADE USAGE, CUSTOM, OR COURSE OF DEALING. WITHOUT LIMITING THE FOREGOING, YOU ACKNOWLEDGE THAT THE SOFTWARE IS PROVIDED "AS IS" AND THAT THE AUTHORS DO NOT WARRANT THE SOFTWARE WILL RUN UNINTERRUPTED OR ERROR FREE. LIMITED LIABILITY THE ENTIRE RISK AS TO RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. UNDER NO CIRCUMSTANCES WILL THE AUTHORS BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE WHATSOEVER, WHETHER BASED ON CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, ARISING OUT OF OR IN ANY WAY RELATED TO THE SOFTWARE, EVEN IF THE AUTHORS HAVE BEEN ADVISED ON THE POSSIBILITY OF SUCH DAMAGE OR IF SUCH DAMAGE COULD HAVE BEEN REASONABLY FORESEEN, AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY EXCLUSIVE REMEDY PROVIDED. SUCH LIMITATION ON DAMAGES INCLUDES, BUT IS NOT LIMITED TO, DAMAGES FOR LOSS OF GOODWILL, LOST PROFITS, LOSS OF DATA OR SOFTWARE, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION OR IMPAIRMENT OF OTHER GOODS. IN NO EVENT WILL THE AUTHORS BE LIABLE FOR THE COSTS OF PROCUREMENT OF SUBSTITUTE SOFTWARE OR SERVICES. YOU

ACKNOWLEDGE THAT THIS SOFTWARE IS NOT DESIGNED FOR USE IN ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS SUCH AS OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR CONTROL, OR LIFE-CRITICAL APPLICATIONS. THE AUTHORS EXPRESSLY DISCLAIM ANY LIABILITY RESULTING FROM USE OF THE SOFTWARE IN ANY SUCH ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS AND ACCEPTS NO LIABILITY IN RESPECT OF ANY ACTIONS OR CLAIMS BASED ON THE USE OF THE SOFTWARE IN ANY SUCH ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS BY YOU. FOR PURPOSES OF THIS PARAGRAPH, THE TERM "LIFE-CRITICAL APPLICATION" MEANS AN APPLICATION IN WHICH THE FUNCTIONING OR MALFUNCTIONING OF THE SOFTWARE MAY RESULT DIRECTLY OR INDIRECTLY IN PHYSICAL INJURY OR LOSS OF HUMAN LIFE. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8 TERMINATION.

8.1.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2.

8.3.

If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9 LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10 U.S. GOVERNMENT END USERS.

11 MISCELLANEOUS.

12 RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

EXHIBIT A.

"The contents of this file are subject to the gSOAP Public License Version 1.3 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.cs.fsu.edu/~engelen/soaplicense.html. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code of the gSOAP Software is: stdsoap.h, stdsoap2.h, stdsoap.c, stdsoap2.c, stdsoap.cpp, stdsoap2.cpp, soapcpp2.h, soapcpp2.c, soapcpp2_lex.l, soapcpp2_yacc.y, error2.h, error2.c, symbol2.c, init2.c, soapdoc2.html, and soapdoc2.pdf, httpget.h, httpget.c, stl.h, stldeque.h, stllist.h, stlvector.h, stlset.h.

The Initial Developer of the Original Code is Robert A. van Engelen. Portions created by Robert A. van Engelen are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

Contributor(s): "_____." [Note: The text of this Exhibit A may differ slightly form the text of the notices in the Source Code files of the Original code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

EXHIBIT B.

"Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved. THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

A.2.15. Chromium Licenses

A.2.15.1. Main License

Copyright (c) 2002, Stanford University All rights reserved.

Some portions of Chromium are copyrighted by individiual organizations. Please see the files COPYRIGHT.LLNL and COPYRIGHT.REDHAT for more information.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Stanford University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.15.2. COPYRIGHT.LLNL File

This Chromium distribution contains information and code which is covered under the following notice:

Copyright (c) 2002, The Regents of the University of California. Produced at the Lawrence Livermore National Laboratory For details, contact: Randall Frank (rjfrank@llnl.gov). UCRL-CODE-2002-058 All rights reserved.

This file is part of Chromium. For details, see accompanying documentation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer (as noted below) in the documentation and/or other materials provided with the distribution.

Neither the name of the UC/LLNL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, THE U.S. DEPARTMENT OF ENERGY OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional BSD Notice

1. This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

2. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness,

or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

3. Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

A.2.15.3. COPYRIGHT.REDHAT File

This Chromium distribution contains information and code which is covered under the following notice:

Copyright 2001,2002 Red Hat Inc., Durham, North Carolina.

All Rights Reserved.

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

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

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

A.2.16. curl License

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2009, Daniel Stenberg, daniel@haxx.se.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

A.2.17. libgd License

Portions copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.

Portions copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.

Portions relating to GD2 format copyright 1999, 2000, 2001, 2002 Philip Warner.

Portions relating to PNG copyright 1999, 2000, 2001, 2002 Greg Roelofs.

Portions relating to gdttf.c copyright 1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).

Portions relating to gdft.c copyright 2001, 2002 John Ellson (ellson@lucent.com).

Portions copyright 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Pierre-Alain Joye (pierre@libgd.org).

Portions relating to JPEG and to color quantization copyright 2000, 2001, 2002, Doug Becker and copyright (C) 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group. See the file README-JPEG.TXT for more information.

Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande.

Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation.

This software is provided "AS IS." The copyright holders disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this code and accompanying documentation.

Although their code does not appear in gd, the authors wish to thank David Koblas, David Rowley, and Hutchison Avenue Software Corporation for their prior contributions.

A.2.18. BSD License from Intel

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.19. libjpeg License

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2010, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

A.2.20. x86 SIMD Extension for IJG JPEG Library License

Copyright 2009 Pierre Ossman <ossman@cendio.se> for Cendio AB

Copyright 2010 D. R. Commander

Based on

x86 SIMD extension for IJG JPEG library - version 1.02

Copyright (C) 1999-2006, MIYASAKA Masaru.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

A.2.21. FreeBSD License

The compilation of software known as FreeBSD is distributed under the following terms:

Copyright (c) 1992-2009 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.22. NetBSD License

Copyright (c) 1992, 1993 The Regents of the University of California. All rights reserved.

This software was developed by the Computer Systems Engineering group at Lawrence Berkeley Laboratory under DARPA contract BG 91-66 and contributed to Berkeley.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.23. PCRE License

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions, and a just-in-time compiler that can be used to optimize pattern matching. These are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS. Written by: Philip Hazel; Email local part: ph10; Email domain: cam.ac.uk University of Cambridge Computing Service, Cambridge, England. Copyright (c) 1997-2012 University of Cambridge All rights reserved.

PCRE JUST-IN-TIME COMPILATION SUPPORT. Written by: Zoltan Herczeg; Email local part: hzmester; Emain domain: freemail.hu Copyright(c) 2010-2012 Zoltan Herczeg All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER. Written by: Zoltan Herczeg; Email local part: hzmester; Emain domain: freemail.hu Copyright(c) 2009-2012 Zoltan Herczeg All rights reserved.

THE C++ WRAPPER FUNCTIONS. Contributed by: Google Inc. Copyright (c) 2007-2012, Google Inc. All rights reserved.

THE "BSD" LICENCE. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.24. libffi License

Copyright (c) 1996-2012 Anthony Green, Red Hat, Inc and others. See source files for details.

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

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

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

A.2.25. FLTK License

December 11, 2001

The FLTK library and included programs are provided under the terms of the GNU Library General Public License (LGPL) with the following exceptions:

1. Modifications to the FLTK configure script, config header file, and makefiles by themselves to support a specific platform do not constitute a modified or derivative work.

   The authors do request that such modifications be contributed to the FLTK project - send all contributions through the "Software Trouble Report" on the following page:

http://www.fltk.org/str.php

2. Widgets that are subclassed from FLTK widgets do not constitute a derivative work.

3. Static linking of applications and widgets to the FLTK library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared FLTK libraries, or link their applications or widgets against a user-supplied version of FLTK.

   If you link the application or widget to a modified version of FLTK, then the changes to FLTK must be provided under the terms of the LGPL in sections 1, 2, and 4.

4. You do not have to provide a copy of the FLTK license with programs that are linked to the FLTK library, nor do you have to identify the FLTK license in your program or documentation as required by section 6 of the LGPL.

   However, programs must still identify their use of FLTK. The following example statement can be included in user documentation to satisfy this requirement:

   [program/widget] is based in part on the work of the FLTK project (http://www.fltk.org).

A.2.26. Expat License

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

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

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

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

A.2.27. Fontconfig License

Copyright (C) 2001, 2003 Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the author(s) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHOR(S) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE

AUTHOR(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

A.2.28. Freetype License

2006-Jan-27

A.2.28.1. Introduction

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don't promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution)
- You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage)
- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (`credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

Portions of this software are copyright (C) <year> The FreeType Project (www.freetype.org). All rights reserved.

Please replace <year> with the value from the FreeType version you actually use.

A.2.28.2. Legal Terms

0. Definitions

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

There are two mailing lists related to FreeType:

- freetype@nongnu.org

  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@nongnu.org

  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

http://www.freetype.org

A.2.29. VPX License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google, nor the WebM Project, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

A.2.30. Opus License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Opus is subject to the royalty-free patent licenses which are specified at: Xiph.Org Foundation: https://datatracker.ietf.org/ipr/1524/ Microsoft Corporation: https://datatracker.ietf.org/ipr/1914/ Broadcom Corporation: https://datatracker.ietf.org/ipr/1526/

A.2.31. FUSE for macOS License

Copyright (c) 2011-2017 Benjamin Fleischer; Copyright (c) 2011-2012 Erik Larsson; All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B Oracle VM VirtualBox Privacy Information

Version 5, Dec 13, 2012

The Oracle Privacy Policies posted on https://www.oracle.com/legal/privacy/privacy-policy.html apply to your personal data collected and used by Oracle. The following privacy information describes in more detail which information is exchanged between the Oracle VM VirtualBox application and Oracle, and which information is collected by the virtualbox.org website.

§ 1 virtualbox.org. The "virtualbox.org" website logs anonymous usage information such as your IP address, geographical location, browser type, referral source, length of visit and number of page views while you visit (collectively, "anonymous data"). In addition, but only if you choose to register, the website's bug tracking and forum services store the data you choose to reveal upon registration, such as your user name and contact information.

§ 2 Cookies. The virtualbox.org website, the bug tracker and the forum services use cookies to identify and track the visiting web browser and, if you have registered, to facilitate login. Most browsers allow you to refuse to accept cookies. While you can still visit the website with cookies disabled, logging into the bug tracker and forum services will most likely not work without them.

§ 3 Oracle VM VirtualBox registration process. The Oracle VM VirtualBox application may ask that the user optionally register with Oracle. If you choose to register, your name, e-mail address, country and company will be submitted to Oracle and stored together with the IP address of the submitter as well as product version and platform being used.

§ 4 Update notifications. The Oracle VM VirtualBox application may contact Oracle to find out whether a new version of Oracle VM VirtualBox has been released and notify the user if that is the case. In the process, anonymous data such as your IP address and a non-identifying counter, together with the product version and the platform being used, is sent so that the server can find out whether an update is available. By default, this check is performed once a day. You change this interval or disable these checks altogether in the Oracle VM VirtualBox preferences.

§ 5 Usage of personal information. Oracle may use anonymous and personal data collected by the means above for statistical purposes as well as to automatically inform you about new notices related to your posts on the bug tracker and forum services, to administer the website and to contact you due to technical issues. Oracle may also inform you about new product releases related to Oracle VM VirtualBox.

In no event will personal data without your express consent be provided to any third parties, unless Oracle may be required to do so by law or in connection with legal proceedings.

§ 6 Updates. Oracle may update the privacy policy at any time by posting a new version at https://www.oracle.com/legal/privacy/privacy-policy.html and the privacy information will be kept up to date in the documentation which comes with the Oracle VM VirtualBox application. You should check these places occasionally to ensure you are happy with any changes.