Chapter 7. Remote Virtual Machines

Table of Contents

7.1. Remote Display (VRDP Support)
- 7.1.1. Common Third-Party RDP Viewers
- 7.1.2. VBoxHeadless, the Remote Desktop Server
- 7.1.3. Step by Step: Creating a Virtual Machine on a Headless Server
- 7.1.4. Remote USB
- 7.1.5. RDP Authentication
- 7.1.6. RDP Encryption
- 7.1.7. Multiple Connections to the VRDP Server
- 7.1.8. Multiple Remote Monitors
- 7.1.9. VRDP Video Redirection
- 7.1.10. VRDP Customization

7.2. Teleporting
7.3. VBoxHeadless

7.1. 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 Section 1.5, "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 from VirtualBox Manager in the **Display** settings, see Section 3.6, "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 `--vrde-port` 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* **--vrde-port 5000,5010-5012** configures the server to bind to one of the ports 5000, 5010, 5011, or 5012. See Section 8.10, "VBoxManage modifyvm".

The actual port used by a running VM can be either queried with the **VBoxManage showvminfo** command or

seen in VirtualBox Manager 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.

7.1.1. 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.

  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.

7.1.2. VBoxHeadless, the Remote Desktop Server

While any VM started from 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 optionally 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**. The full documentation of the command is in Section 7.3, “VBoxHeadless”.

- Start **VBoxHeadless** from VirtualBox Manager, 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.

7.1.3. 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 7.1, "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 7.1.1, "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 7.1.1, "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.

7.1.4. 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 Section 3.11.1, "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. Some versions of **uttsc**, a client tailored for the use with Sun Ray thin clients, support accessing remote USB devices. RDP clients for other platforms will be provided in future Oracle VM VirtualBox versions.

7.1.5. 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 `--vrde-auth-type` option. See Section 8.10, "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 macOS 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 --vrde-auth-type 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 11, *Oracle VM VirtualBox Programming Interfaces*.

7.1.6. 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 --vrde-property "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 \
  --vrde-property "Security/CACertificate=path/ca_cert.pem"

vboxmanage modifyvm VM-name \
  --vrde-property "Security/ServerCertificate=path/server_cert.pem"

vboxmanage modifyvm VM-name \
  --vrde-property "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.

7.1.7. 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 --vrde-multi-con on
```

7.1.8. Multiple Remote Monitors

To access two or more remote VM displays you have to enable the VRDP multiconnection mode. See Section 7.1.7, “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.

7.1.9. 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 --vrde-video-channel 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 --vrde-video-channel-quality 75
```

7.1.10. 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 --vrde-property Client/DisableDisplay=1
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableInput=1
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableUSB=1
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableAudio=1
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableClipboard=1
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableUpstreamAudio=1
```

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

```
$ VBoxManage modifyvm VM-name --vrde-property Client/DisableDisplay=
```

7.2. 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 macOS 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 --teleporter-port 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 Section 8.10, “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 Section 8.20, “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-portability-level** command. See Section 8.10, "VBoxManage modifyvm".

7.3. VBoxHeadless

Oracle VM VirtualBox remote desktop server.

Synopsis

```
VBoxHeadless [--startvm= [ uuid | vmname ]] [--vrde= on | off | config ] [--vrdeproperty=prop-name=[prop-value]]
    [--settingspw=[password]] [--settingspwfile=password-file] [--start-paused=vmname] [--capture] [--width=width]
    [--height=height] [--bitrate=bit-rate] [--filename=filename]
```

Description

The **VBoxHeadless** command is an alternate front end that enables you to remotely manage virtual machines (VMs). The front end is a CLI rather than the VirtualBox Manager graphical user interface (GUI).

For information about using this command, see Section 7.1.2, "VBoxHeadless, the Remote Desktop Server".

Command Options

`--startvm=uuid | vmname`

Specifies the Universally Unique Identifier (UUID) or name of the VM to start.

Use the **VBoxManage list vms** command to obtain VM information.

The short versions of this option are `-s` and `-startvm`.

`--vrde=on | off | config`

Specifies how to use the VRDP server. The default value is `config`. Valid values are as follows:

- `on` enables the VRDP server.

  ```
  VBoxHeadless --startvm=vmname --vrde=on
  ```

- `off` disables the VRDP server.

  ```
  VBoxHeadless --startvm=vmname --vrde=off
  ```

- `config` enables the VRDP server depending on the VM configuration.

  ```
  VBoxHeadless --startvm=vmname --vrde=config
  ```

The short version of this option is `-v`.

`--vrdeproperty=`*`prop-name`*`=`*`prop-value`*

Specifies a value for one of the following properties:

- The `TCP/Ports` property value is a comma-separated list of ports to which the VRDE server can bind. Use a hyphen (`-`) between two port numbers to specify a range of ports.
- The `TCP/Address` property value is the interface IP address to which to bind the VRDE server.

`--settingspw=[`*`password`*`]`

Specifies a settings password to access encrypted settings. If you do not specify the password on the command line, **VBoxHeadless** prompts you for the password.

`--settingspwfile=`*`password-file`*

Specifies the file that contains the settings password.

`--start-paused=`*`vmname`*

Starts the specified VM in the paused state.

`--capture`

Records the VM screen output to a file. In addition to this option, you must use the `--filename` option to specify the name of the file.

`--width=`*`width`*

Specifies the frame width of the recording in pixels. This option is associated with the `--capture` option.

`--height=`*`height`*

Specifies the frame height of the recording in pixels. This option is associated with the `--capture` option.

`--bitrate=`*`bit-rate`*

Specifies the bit rate of the recording in kilobits per second. This option is associated with the `--capture` option.

`--filename=`*`filename`*

Specifies the name of the file in which to store the recording. The codec used is based on the file extension that you choose. You must specify this option if you use the `--capture` option.

Examples

The following command starts the `ol7u4` VM:

```
$ VBoxHeadless --startvm "ol7u4"
```

The following command starts the `ol7u6` VM in the Paused state.

```
$ VBoxHeadless --startvm "ol7u6" --start-paused
```

The following command starts the `ol7u6` VM and records the session. The recording is saved to the `ol7u6-recording` WebM file.

```
$ VBoxHeadless --startvm "ol7u6" --capture --filename ol7u6-recording.webm
```

See Also

Section 8.5, “VBoxManage list”, Section 8.19, “VBoxManage startvm”