

VIVO v1.2 VirtualBox Appliance

note|Some of the content on this page refers to VIVO 1.2 and earlier releases. Please see [VIVO v1.3 VirtualBox Appliance](#) for additional information.

The VirtualBox appliance for v1.2 uses Debian 6.0.1

Login Information

The root user is *vitro*. The password is *vitro123*.

The default administrator of the VIVO site is *defaultAdmin*. The password is *defaultAdmin*. There will be a password change prompt on first login.

note: as of 1.3 admin is: *vivo_root@localhost* with password of *vitro123*

Browse from host machine

Tomcat runs on port 8080. To browse from the host machine, after installing the machine, you will have to run the following commands in a console.

```
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/HostPort" 8080
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/GuestPort" 8080
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/Protocol" TCP
```

Note: <guestname> is the name of the virtual machine within Virtualbox. (ie. "vivo-rel-1.2-vm")

In the VirtualBox Manager, click "settings", then click "Network", then click the arrow next to "Advanced" and change the MAC address to 080027C19477.

Then you can browse VIVO via _

```
http://localhost:8080/vivo
```

_. You can now login to the VIVO web application <http://localhost:8080/vivo> Username: defaultAdmin Password: defaultAdmin

ssh into Appliance

The 1.2 appliance has an ssh server installed. In order to use it, you will have to configure VirtualBox to map a port on the host system to the ssh port (port 22) on the guest.

To map host port 2222 (for example) to guest port 22, run the following commands in a console, replacing <guestname> with the name of your appliance (by default *vivo-rel-1.2-vm*):

```
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/HostPort" 2222
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/GuestPort" 22
VBoxManage setextradata <guestname> "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/Protocol" TCP
```

Then you can ssh into the appliance using the following command:

```
ssh -l vitro -p 2222 localhost
```

Enter *vitro123* at the password prompt and you're in.

Secure file copy

From the host computer, you can copy files to the virtual appliance as follows:

```
scp -P 2222 <filename> vitro@localhost:<Destination>
```

Bash Script

A convenient bash script for those who find themselves reinstalling the virtual appliance repeatedly is this:

```
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/HostPort" 8080
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/GuestPort" 8080
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/apache/Protocol" TCP
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/HostPort" 2222
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/GuestPort" 22
VBoxManage setextradata $1 "VBoxInternal/Devices/pcnet/0/LUN#0/Config/ssh/Protocol" TCP
```

The \$1 refers to the first command-line argument, so calling this script would look something like this:

```
bash vmports.sh demovm
```

if the script and virtual machine are named vmports.sh and demovm respectively.

Conflicts with a local VIVO Installation

If you have a local VIVO running on port 8080, just change the HostPort for apache in the above commands to something else, like 8081. Keep GuestPort 8080. Then you can run something like

```
http://localhost:8081/vivo
```

, to connect to your virtual appliance's VIVO installation running on (virtual) port 8080, while

```
http://localhost:8080/vivo
```

will connect to your local VIVO installation.

Log files

The log files catalina.out and vivo.all.log are not in the same folder (as is the case with some VIVO installations). Their locations are:

```
/var/log/tomcat6/catalina.out
/usr/share/tomcat6/logs/vivo.all.log
```