IRAF$_V M_t utorial$

Jan 13, 2023

Contents:

Instructions for running Astroconda (32-bit) IRAF on MacOS 10.15 or later, under an x86 Linux virtual machine.

A link to the previous instructions for VirtualBox can be found below, but they do **not** work on newer M1/M2 (ARM) Apple machines, nor can VMWare or Parallels run Gemini IRAF on those machines.

This method uses QEMU to emulate x86_64 hardware on ARM64, which works well, producing the same results as on a Linux machine, but with a large performance penalty. A test case using GMOS multi-slit data took **14 times** as long to run on an M1 machine as when running natively on a recent i7 processor, with the overhead varying considerably between different processing steps. This is, however, the *only* way of running the necessary IRAF packages on new Apple CPUs (at least prior to MacOS 13).

Compared with the original 2020 release for VirtualBox, the new Gemini IRAF VM image has the following features:

- A new control interface, GemVM.
- A simpler installation method, using Anaconda / Miniconda.
- The latest Gemini IRAF 1.15.
- The latest DRAGONS 3.0.3.[1]
- A new version of PyRAF (2.2.1) that is fully compatible with Python 3.
- More recent Anaconda (2021.11) package versions.
- The (unofficial) PyFU scripts for mosaicking IFU datacubes.
- Support for compiling IRAF packages.
- Miscellaneous OS updates etc.

[1] You should run DRAGONS directly on your host machine instead, unless you really need to use it in conjunction with IRAF.

CHAPTER 1

Installation & setup

1.1 Get the installation files

- Download the IRAF VM disk image file (`gemini-IRAF-CO7-2022.07.zip`) from . This is 5GB in size, so will take a while to transfer.
  - If you're using Apple's Safari Web browser, it will probably unzip the file for you automatically. Otherwise, you can open a terminal window, go to the directory where the file was downloaded (usually `cd Downloads`) and type `unzip gemini-IRAF-CO7-2022.07.zip` (or use another archive manager) to extract it.
  - To ensure the integrity of the download, type `shasum gemini-IRAF-CO7-2022.07.qcow2` and verify that the resulting checksum is `932d9db4224429cb326bfe969253ec9e75253dcc`; if not, you should try downloading again.
- If you don't already have Anaconda installed on your host machine, download it from (noting the non-commercial for their repository). You may use Miniconda if you prefer.

> **Warning:** You should normally download Anaconda's "64-Bit Command Line Installer" for Intel, **even on Apple M1/M2 (ARM64) machines**, *not* the "64-Bit (M1)" installer.
>
> Anaconda's new M1 installer *can* be used, but there is no DRAGONS build for it yet (as of mid 2022), which is needed for reducing Gemini imaging (and eventually spectroscopic) data in Python. In the meantime, a small M1-native enviroment can be created within the Intel/MacOS version of Anaconda, allowing the IRAF VM to coexist easily with DRAGONS.

1.2 Set up Anaconda on your host machine

If you haven't already done so, you should install Anaconda as for (or following Anaconda's instructions). Make sure that the necessary conda channels are defined, as in the section "Set up Anaconda Channels" of that page – in summary, you should have done the following:

```
conda config --add channels conda-forge
conda config --add channels http://astroconda.gemini.edu/public
```

It is not obligatory to install DRAGONS itself. Don't install Gemini IRAF, since that's provided by the VM (on MacOS 10.15+).

1.3 Install GemVM

With the Anaconda base environment activated (type `conda activate`) and Gemini's public conda channel defined (see the DRAGONS link above), issue the following command to install GemVM and its dependencies:

Apple M1/M2 (ARM64) host machines:

```
CONDA_SUBDIR=osx-arm64 conda create -n gemvm gemvm
```

Intel host machines:

```
conda create -n gemvm gemvm
```

1.4 Set up the VM

- Place the VM disk image that you downloaded earlier in a safe, permanent location (such as `~/GemVM/`). If anything happens to this file, you will lose the entire contents of the VM. The file will initially occupy 14GB of disk space, growing to a maximum of 50GB as you create data files on the VM.
- Run `gemvm-config` to assign a name/label (and any other parameters you want to tweak) to the downloaded VM image, eg.:

  ```
  conda activate gemvm
  gemvm-config add IRAF-2022.07 ~/GemVM/gemini-IRAF-CO7-2022.07.qcow2
  ```

  where `IRAF-2022.07` is the name you wish to use. Referring to the VM image by an assigned name is both more convenient and safer than typing the file path every time, since it's easy to work in another directory, with less risk of inadvertently deleting the image.

CHAPTER 2

Starting the VM & logging in

- Make sure the appropriate conda environment is activated in your terminal window:

```
conda activate gemvm
```

- Start the VM as follows:

```
gemvm IRAF-2022.07
```

  (where `IRAF-2022.07` is whatever name you assigned to the VM in *Set up the VM*).

  For additional options, refer to the help output from `gemvm -h`.

  The status shown in the terminal will change from "booting" to "running" once the VM is ready to log into (which should take less than a minute). See *Troubleshooting* if that doesn't happen.

  If there is not a large enough block of memory available (3GB by default), GemVM will ask you to try re-starting any large programs, such as your Web browser; see *Failure to start*.

- Log in "remotely" to the VM with X11 forwarding, by typing the following command in at least one other terminal window:

```
ssh -Y -p 2222 irafuser@localhost
```

  The port number (2222) might vary if you override it. The password should appear above the login prompt and is unchanged from the previous (VirtualBox) version of the VM.

Note: Use `-Y`, rather than `-X`, to help avoid time-outs that can cause graphics display to stop working.

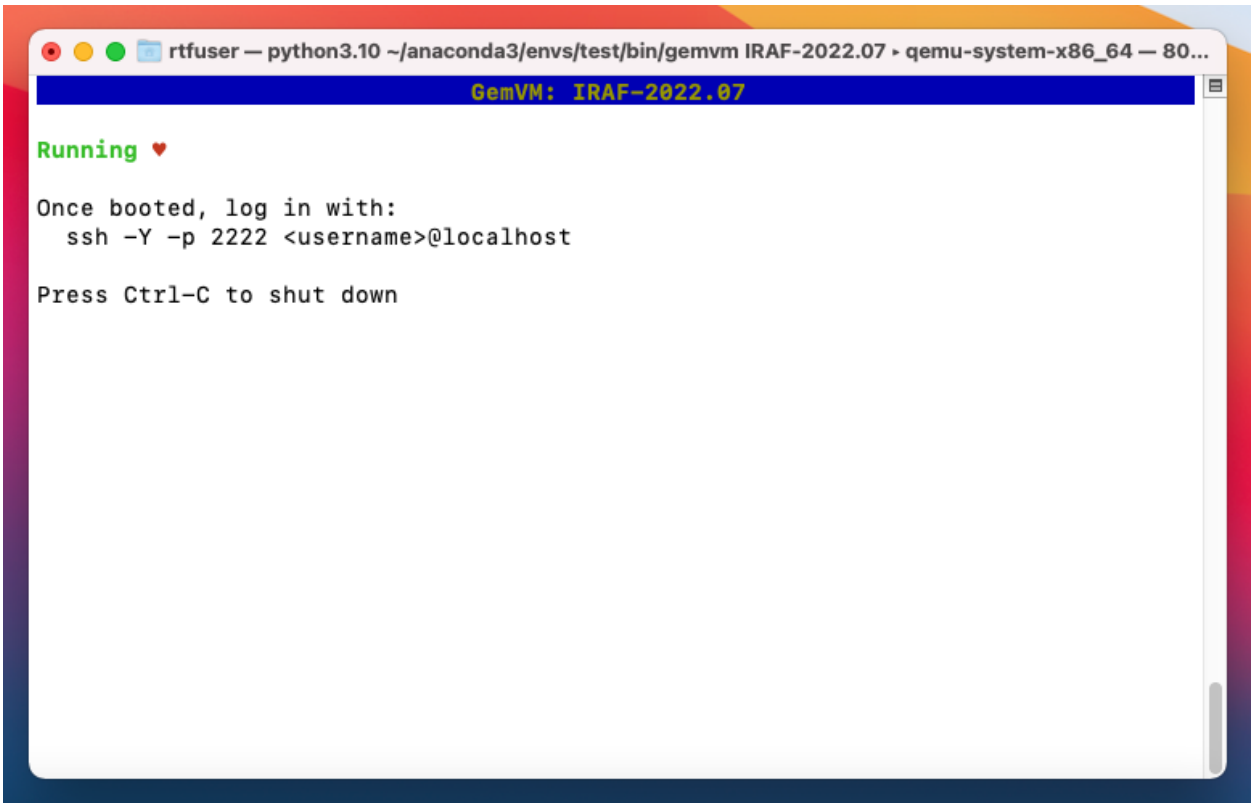

Fig. 1: The GemVM control screen, running in a terminal window.

CHAPTER 3

Sharing data with the host

In order to share files between the VM and your host machine, you can mount a subdirectory from the host on `/home/irafuser/vm_transfer`, using `sshfs`.

For example, if your username is `gumby` and you wish to copy files to/from `/home/gumby/data`, you can log onto the VM as `irafuser` and issue the following command:

```
sshfs gumby@vmhost:data/ ~/vm_transfer/
```

where `vmhost` is a literal alias for your host machine. Any path after the colon that doesn't begin with a `/` is relative to your home directory. Remember you're entering the password for `gumby`, not `irafuser`.

To umount the shared directory again (so you can mount a different one), type `fusermount -u ~/vm_transfer`.

Note: The `vm_transfer` directory is intended for sharing input & output files, not for processing data in. It's suggested that you keep raw input files there and set the `rawpath` parameter of the relevant Gemini IRAF tasks to point to `vm_transfer` from your working directory. Then when your data reduction is complete, you can copy the final results back there.

When using GemVM rather than VirtualBox, you *can* work directly in `vm_transfer` if you really want to, but data processing will take roughly 3-4x as long (and is *already* slow when emulating Intel Linux on M1). You can fit about 40GB of data on the VM itself, as long as your host machine has that much space available for the expanded disk image.

Note: If you wish to mount shared directories without having to enter a password, you may create an ssh key pair on the VM and install the public key (eg.`~/.ssh/id_rsa.pub`) on your host machine. We cannot do that for you without compromising your security, but a Web search (eg. for "linux ssh key pair login") will turn up many tutorials explaining what to do. **Make sure that the disk image (qcow2 file) on your host machine is not readable by other users, since it can now be used to gain access to your host machine!**

CHAPTER 4

Using the VM

- The VM comes with an `~/iraf` directory, containing a default `login.cl` & `uparm/`.
- The `geminiconda` environment is activated automatically on login.
- Start DS9 (or XImtool) and PyRAF in your VM session:

```
ds9 &
cd iraf
pyraf
gemini
```

- You may then continue processing data as usual. All the IRAF packages from Gemini's Astroconda channel are available, along with `emacs` & `vi`.
- You can also use DRAGONS (and other conda packages), but it's better to run those directly on the host machine wherever possible, since processing data on non-x86 machines is an order of magnitude slower when using the VM and you are more likely to run into resource limitations there. Only the IRAF packages *require* a VM on MacOS.
- You can administer the conda packages in `/home/irafuser/miniconda3` as usual, though conda commands might take some minutes to run on the VM. The operating system (CentOS 7) installation is **not** intended to be user-maintainable.
- It should be possible to leave the VM running while your laptop is suspended and resume processing later – but you may find that the clock time is wrong on the VM afterwards. There is currently no way to suspend the VM itself.

CHAPTER 5

Shutting down the VM

- You can shut down the VM by pressing control-c at any time in the terminal where `gemvm` is running. This does a clean ACPI shutdown (similar to pressing the soft power button), but it's best to stop any programs that are running first, especially data reduction scripts. If the OS is still booting when you press ctrl-c, you will have to wait for that to complete before the shutdown starts, after which it should only take a few (~5) seconds.
- Occasionally, if something goes wrong with the boot or shut-down sequence, GemVM might time out and return you to the command prompt, with QEMU still running as a background process. See *Time-Outs*.

CHAPTER 6

Other things to know

- While the VM is running idle in the background, it will probably use a few percent of a CPU core, which may drain your laptop battery slightly faster than usual.
- If you delete or overwrite the qcow2 file, you will lose all the files that were on the VM (except those mounted under `vm_transfer`). To reduce the risk of this happening, it is recommended that you keep that disk image in a safe location, refer to it using a name defined in your configuration file (not by its path) and run gemvm in another directory (such as ~).
- If your disk image (qcow2 file) is readable by other users on the same host machine or network filesystem, they will be able to boot a copy and see all of the files you have placed on the VM. For that reason, the image is distributed in an archive file with restricted read permissions by default. Do not let anyone obtain a copy of your disk image if you have installed an ssh key that allows you to mount directories from your host machine without a password! They will be able to log into your host account.

CHAPTER 7

Troubleshooting

7.1 Failure to start

If `gemvm` returns you to the command prompt with an error status almost immediately, the most likely reasons are:

- The VM image is already running. Either you have `gemvm` active in another terminal or QEMU has become detached (eg. after a time-out) and is still running in the background. The log file will say `Failed to get "write" lock` and `Is another process using the image [<filename>]?` at the top. If you can't find an active `gemvm` control screen, try logging into the VM with SSH (*Starting the VM & logging in*) and issue `sudo shutdown now` when finished. Failing that, see *Time-Outs* below.
- A large enough block of memory (by default 3GB) could not be allocated. GemVM will tell you this and advise you to try re-starting any large programs, such as your Web browser, to reduce memory fragmentation and usage. On low-memory (eg. 4GB) systems, you might have to keep such programs closed while using the VM, but it's more likely that re-starting them will suffice. If such errors persist, try reducing the VM's memory allocation (in GB), by specifying the `-m` argument to `gemvm` (or `gemvm-config add`, to make it permanent). While not optimal, PyRAF can run on CentOS 7 in as little as 0.5GB, or even 0.25GB with a modest amount of swapping. It is inadvisable to try booting with <0.25GB of RAM, which could lead to pathologically slow and/or problematic behaviour.
- The default ssh port (2222) is unavailable. The log file will say `Could not set up host forwarding rule 'tcp:127.0.0.1:2222-:22'` at the top. This can happen because another VM is already running using a *different* disk image. To run another VM, you'll need to increment the port number to 2223 using the `-p` option (and so on).
- You have installed the Intel version of the GemVM stack on an M1/M2 machine. This is an unstable and extremely slow combination, which won't start at all with the QEMU options passed by GemVM. The log file will say `Error: HV_ERROR` near the top. Please remove the problematic environment using `conda remove -n <bad_env_name> --all` (after doing `conda deactivate`) and see *Install GemVM* for the correct installation command.

7.2 Time-Outs

If something goes wrong with the boot or shut-down sequence (eg. due to filesystem corruption), GemVM might time out and return you to the command prompt. In this case, QEMU will probably still be running as a background process. GemVM will try to print an informative message, including the process number (if applicable). You can still try logging into the VM (if it has just taken longer than normal to boot) and issuing `sudo shutdown now` as `irafuser` once you have finished. If you can't log in, try killing `qemu-system-x86_64` with no flags (ie. `kill <pid>`), which should cause QEMU to flush its disk buffers before terminating without an error – that is not a clean OS shutdown but is the next best thing. If that doesn't stop the process, you can clean up with `kill -9 <pid>` as a last resort.

If the VM persistently fails to boot, a `--console` option can be used to see a boot screen that may provide additional information (please summarize any errors as accurately as feasible in any helpdesk communications). How this looks depends on the back-end QEMU capabilities and it might not be visible at all if you run gemvm remotely.

Another reason for a time-out could be booting from a file that is not a valid disk image (normally it will be a QCOW2 file). The log file will probably say `No bootable device` near the end. You will have to kill QEMU manually in this case. The stray background process should not consume a lot of resources, but will tie up the default ssh port (see last bullet) and is best cleaned up.

7.3 Known problems

- The VM clock can lose time when a host laptop goes to sleep, even if the lid is open. We don't have an immediate solution for this; for the time being, you can either disable sleep mode or live with incorrect (but functionally harmless) timestamps on your files. Sorry for any inconvenience.

7.4 Helpdesk

When requesting helpdesk support for problems related to starting, stopping or logging into the VM, please send `gemvm_<name>.log`, from the directory where you run `gemvm` (**without** the `--console` option). Wait until the process has stopped before copying the log file.

CHAPTER 8

Old VirtualBox instructions

These are the "old" instructions from 2020, for Apple machines with an Intel CPU; they do not work on machines with an M1/M2 (ARM) processor, but you might find it convenient to continue using this method on older Apple machines.

These instructions are for use with Virtual Box. You may prefer to import the provided OVA file into some other virtualization software, in which case you will have to determine how to adapt the following installation and start up steps accordingly.

An alternative to using Gemini's VM image as described here is to set up your own virtual machine (including 32-bit OS libraries) and install the Astroconda packages for Python 2 on it. This is reportedly easy to do using Parallels (for a fee).

8.1 Installation & setup

- Get the installation files
  - Download the OVA file containing the IRAF VM image (`gemini-IRAF-CO7.ova`) from . This is 6GB in size, so will take a while to transfer.
  - To ensure the integrity of the download, you can open a terminal window (see *Starting the VM & logging in*), type `shasum Downloads/gemini-IRAF-CO7.ova` (substituting whatever path you downloaded it to) and verify that the resulting checksum is `1881ae0afa3e9699ccec861b508a630787cf566d`; if not, you should try downloading again.
  - Download the DMG file for the free version of Virtual Box from , under `OS X hosts`. A checksum is also available on that page, which you can verify using `shasum -a 256 filename.dmg`.
- Install Virtual Box
  - Once the downloads are complete, go to `Downloads` and select the VirtualBox DMG file to start the installer. Follow the instructions, entering your personal password when prompted to do so. Your administrator may have to perform this step, if you do not have software installation privileges.
- Import the VM image

- Ensure that you have more than 16GB of available disk space for the VM image in your home directory (if necessary, you may reclaim 6GB afterwards by deleting the OVA file).
- Start the "VirtualBox" application.

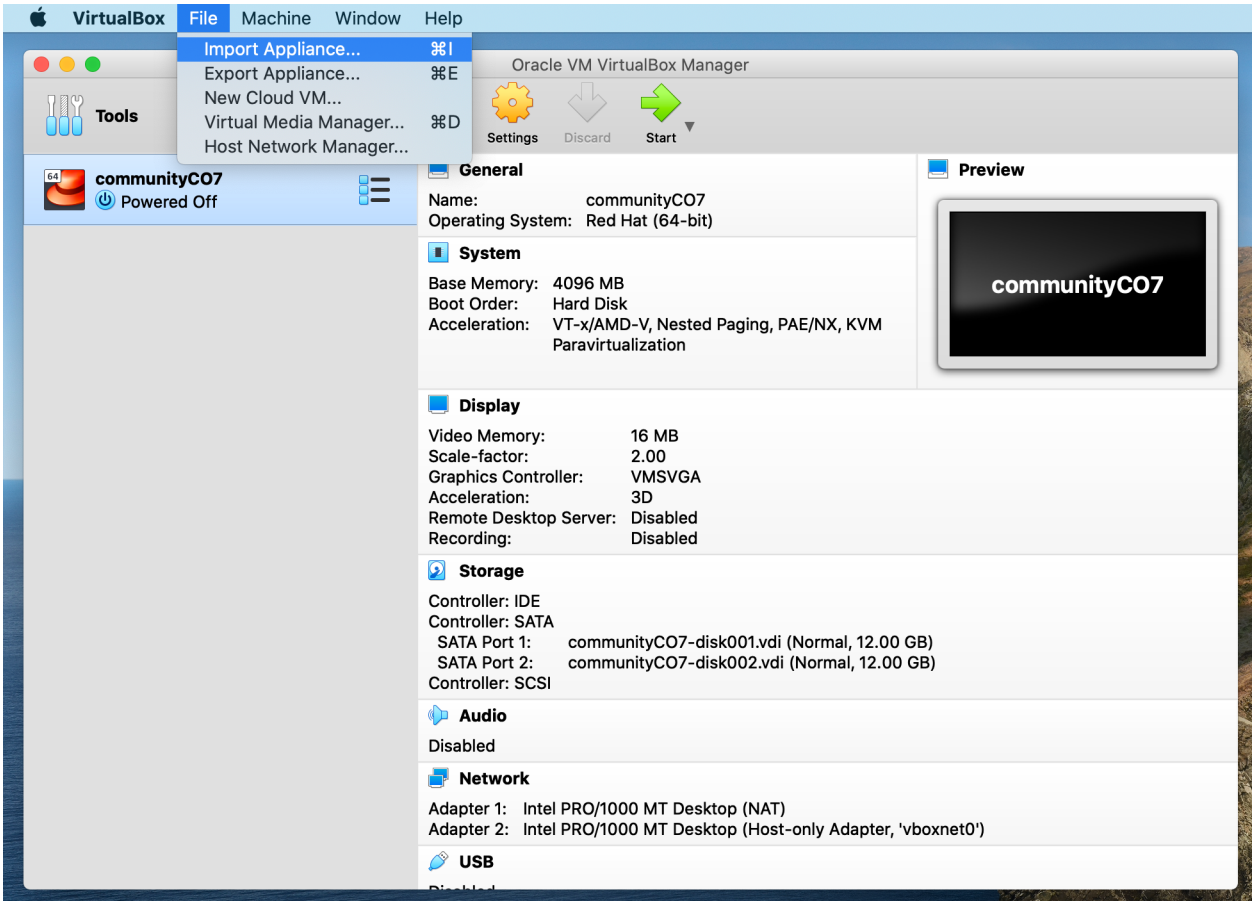

Fig. 1: The main VirtualBox Manager window and File menu.

- Go to `File` in the menu bar at the top of the screen, then `Import appliance`. Press the icon to the right of the `File` box, go to `Downloads` (or wherever you put the OVA file) and select `gemini-IRAF-CO7.ova`. Press `Continue` in the main window.

  Alternatively, double clicking on `gemini-IRAF-CO7.ova` may open it in Virtual Box automatically, depending on your settings.

  * Accept the default settings.
  * Press `Import` and wait for the process to complete.

- Configure networking
  - With the `Oracle VM VirtualBox Manager` window selected, go to `File` in the menu bar at the top of the screen and select `Host Network Manager`.
  - Press `Create` at the top left of the `Host Network Manager` window. This should automatically add an entry with network address `192.168.56.1/24` in the table beneath. Don't enable `DHCP Server` (unless you're already using it for another purpose). Close the window.
- Make sure `gemini-IRAF-CO7` is selected on the left-hand side of the `Oracle VM VirtualBox Manager` window.

- Configure a shared data directory.
  - Under your home directory (or another writeable location) on your host machine, create a subdirectory for exchanging data files between the host and the VM, eg. `vm_transfer/`.
  - In the `Oracle VM VirtualBox Manager` window, press `Settings`, then `Shared Folders` in the top row of icons, then the + icon to the right of the main table. In the sub-window that pops up, set the `Folder Path` to the directory you created on the host machine (eg. `/Users/<username>/vm_transfer`) and the `Mount Point` to `/home/irafuser/vm_transfer` (or similar). Select the `Auto-mount` option (and `Make Permanent`, if you have it). Press `OK` and then `OK` again in the parent window.

8.2 Starting the VM & logging in

- Press the `Start` arrow at the top of the `Oracle VM VirtualBox Manager` window to turn on the VM. A console window will open and you will see the machine booting in it. You don't need to press anything when the boot menu briefly appears. Once the machine has finished booting, you should see a "Welcome to the Gemini IRAF VM!" banner with a login prompt underneath.

> **Warning:** If your mouse pointer stops working, press the left *command* key to toggle off mouse/keyboard capture by the VM window.

- Note the password shown in the banner in the console window.
- Minimize the console window, which is only used for controlling the VM and not for processing data. The VM has no graphical desktop enviroment and graphics are instead displayed to the host desktop using ssh and XQuartz.
- Log in remotely to the VM from one or more terminal window(s).
  - Open a Terminal window (eg. by typing *term* in the `Launchpad` search box and pressing the Terminal icon).
  - Log in to the following fixed IP address with X11 forwarding, using the password noted above:

```
ssh -Y irafuser@192.168.56.15
```

> **Note:** Use `-Y`, rather than `-X`, to help avoid time-outs that cause graphics display to stop working.
>
> There is already an `~/iraf` directory with a default `login.cl` & `uparm/`.
>
> The necessary conda environment is activated automatically on login.

8.3 Using the VM

- Start DS9 (or ximtool) & PyRAF in your VM session:

```
ds9 &
cd iraf
pyraf
gemini
```

- You may then continue processing data as usual. All the IRAF packages from Astroconda are available, along with `emacs` & `vi`.

> **Warning:** Don't work in `vm_transfer`; use it for transferring input/output files.
>
> When working in the directory shared with the host machine (eg. `vm_transfer`), a problem has been observed where certain IRAF tasks (including `gdisplay`) appear to run normally but only process one image extension per file, or one of several files, producing incomplete output. You should therefore **work in** another directory under `/home/irafuser` (so that temporary files get written there), but you can still keep input & output files in `vm_transfer` and point the `rawpath` parameter of the appropriate IRAF tasks there, in order to find them.

- You can also use DRAGONS (and other conda packages), but it is recommended that you instead do that directly on the host machine where possible, as you are more likely to run into resource limitations on the VM. Only the IRAF packages in Astroconda require a 32-bit VM on MacOS >=10.15.
- You should be able to administer the conda packages installed in `/home/irafuser/anaconda2` as usual. The operating system (CentOS 7) installation is **not** intended to be user-maintainable.

8.4 Shutting down the VM

- You can shut down the VM by clicking the close button (⊗) on the upper left of the console window, or from `Machine` in the menu bar. To shut it down completely, you can choose `Send the shutdown signal` / `ACPI shutdown`, or you can `Save the machine state` to resume later where you left off.

8.5 Other important notes

- While the VM is running idle in the background, it will probably use a few percent of a CPU core, which may drain your laptop battery faster than usual.
- If you delete the VM, you will likely lose all the files that were on it (except those in the shared directory).
- Make sure you have read the warning about not working directly in the `vm_transfer` directory under *Using the VM*.