

News from the Society for Astronomical Sciences

Vol. 14 No.3 (November, 2016)

Getting Ready for SAS-2017

The 2017 SAS Symposium will be held on June 15-16-17, 2017, at the Ontario Airport Hotel. There will be Workshops on topics of interest to the small-telescope research community, Technical Papers on research results and project plans, and our Sponsors will have new products on display.

The SAS Program Committee is preparing the details of the Symposium and looking into some new features.

Here's what you can start doing now:

- Block the dates (June 15-16-17, 2017) on your calendar.
- Make travel reservations.
- Keep working on your projects, and decide which one you would like to present.
- Start preparing your Abstract
- Invite your colleagues who are interested in small-telescope research activities.

Kudos or Criticisms?

We are looking forward to seeing results on a wide diversity of projects and objects at SAS-2017! If you have any questions or ideas for the Symposium, or comments related to the 2016 Symposium, please share them with the Program Committee at program@SocAstroSci.org.

We will appreciate your input about the technical sessions, the workshops, the hotel, the banquet, the lunchtime discussions, and anything else that might help us improve the future SAS Symposia.



Reminders ...

Membership Renewal: Even if you can't attend the annual Symposium, we value your support of the Society for Astronomical Sciences, and your interest in small-telescope science. You can renew your membership on the SAS website (SocAstroSci.org), by going to the MEMBERSHIP/REGISTRATION tab.

Symposium Proceedings: Published proceedings from all recent Symposia are freely available in PDF format at the PUBLICATIONS tab of the SAS website (www.SocAstroSci.org).

Symposium Videos: If you missed a recent Symposium, you can still watch the presentations. Videos of most of the technical presentations have been posted on the SAS website at the PUBLICATIONS tab.

Workshop Videos: Video recordings of most of the Workshops from recent years are available from SAS. If you were registered for the Workshop, then the recording is free. If you were not a registered attendee, then the price is \$50 per workshop. Contact Bob Buchheim (Bob@RKBuchheim.org) for the details.

Contact Information: If you haven't been receiving e-mail messages about the Newsletter or the SAS Symposium, perhaps it's because you've changed your e-mail address. You can update your contact information on the MEMBERSHIP/REGISTRATION tab of the SAS website. SAS never sells or shares your name or contact information without your explicit permission.

Links to Videos of 2016 SAS Symposium Presentations		
Title	Author	Link to Video
The Role of Amateur Astronomers in Exoplanet Research	Dennis M Conti	https://youtu.be/PlpzUZFv598
An Automated System for Citizen Searches for Exoplanets	Stephen J Edberg	https://youtu.be/vRpQmKbsXQg
A Bespoke Spectropolarimetrist	John L Menke	https://youtu.be/5Tukpb6GbVY
Measuring Stellar Radial Velocities with a LISA Spectrograph	David Boyd	https://youtu.be/yxXK_w2-o_E
Crowd-Sourced Spectroscopy of Long Period Mira-Type Variables	John C Martin	https://youtu.be/GZo0pWufUyg
Small telescope Spectropolarimetry: Instrumentation and Observations	Gary M Cole	https://youtu.be/DylEipi9a9o
Repeating the Experiment that made Einstein Famous	Donald G Bruns	https://youtu.be/ADON2mlBysY
Small Telescope Research Communities of Practice	Russell M Genet	https://youtu.be/SXQzHzdY32w
Empirical Measurements of Filtered Light Emitting Diode (FLED) Replacements	Eric R Craine	https://youtu.be/m8P24VXvZVw
Clues to the Evolution of W Ursae Majoris Contact Binary Star Systems	Robert M Gill	https://youtu.be/rYb98C2V2c8
Follow Up Photometry on MOTESS-GNAT Variable Star Candidates	John E Hoot	https://youtu.be/Z-SsxPijPrM
New Observations of the Variable Star NGC 6779 V6	Jerry D Horne	https://youtu.be/ZNiymaTwVz0
Mixed-model Regression for Variable-star Photometry	Eric V Dose	https://youtu.be/mc0ym-NBYVU
Predicting a Luminous Red Novae	Daniel M Van Noord	https://youtu.be/YKR3Ev4w5jA
White Dwarf WD-1145	Thomas G Kaye	https://youtu.be/l38sdKCUd7g
Astronomical Instrumentation System Markup Language (AISML)	Jesse M Goldbaum	https://youtu.be/FRs9Bqwt8
M-Dwarf discovery and variability from serendipitous field star observations.	Wayne L Green	https://youtu.be/XJ6uQmmVK3A
Early Images of Sodium in the Tail of Comet Hale-Bopp	Tom Buchanan	https://youtu.be/g8F5AOKahSA
What color is PG1302-102?	Robert K Buchheim	https://youtu.be/VDFv9aYDXQw

Symposium Sponsors

The Society for Astronomical Sciences thanks the following companies for their participation and financial support. Without them, our Symposium would not be possible. We encourage you to consider their fine products for your astronomical needs.



Sky & Telescope Magazine

The Essential Magazine of Astronomy

<http://www.skyandtelescope.com/>



DC3 Dreams Software

Developers of ACP Observatory Control Software

<http://www.dc3.com/>



PlaneWave Instruments

Makers of the CDK line of telescopes

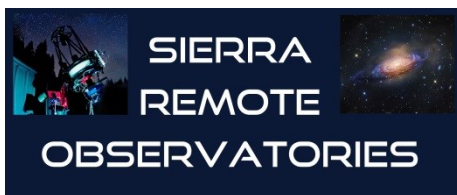
<http://www.planewaveinstruments.com/>



Woodland Hills Camera & Telescopes

Providing the best prices in astronomical products for more than 50 years

<http://www.telescopes.net/>



Sierra Remote Observatories

Hosting telescopes for remote imaging and data acquisition

<http://www.sierra-remote.com/>

Targets that need your attention

Quite a few objects and projects were identified at SAS-2016 that are in need of monitoring and/or follow-up.

To refresh your memory – and your observing list – here are the ones that are well-placed for this observing season:

BRITE target: The following northern-hemisphere stars have been identified as BRITE targets for which ground-based observations are requested:

eps Per (ϵ Per) Vmag \approx 2.9 displays beta Cephei-type pulsations. Time-resolved spectroscopy, with time resolution better than 15 minutes is requested. The BRITE contacts are: Elzbieta Zocłonska (ela@camk.edu.pl) and Gerald Handler (gerald@camk.edu.pl). The star's coordinates are RA= 03:57:51.2, Dec= 40:00:36.77.

Targets mentioned at SAS-2016:

VV Cep: This target is still observable, promptly after dark. A long run of pre-eclipse photometry and spectroscopy is needed to characterize the system before the eclipse begins. Based on prior eclipses, first contact is expected in August 2017, so now is the time to start if you haven't already. The emission features seem to change from night-to-night (at $R \approx 5000$ or higher), out-of-eclipse. Refer to the ARAS discussion forum for more about this system. See

www.spectro-aras.com/forum/.

WD-1145+17: As Tom Kaye described in his presentation at SAS-2016, something odd is going on at this "zombie star". The truth is out there, but further time-series photometry is needed, to find it. The star is in Virgo (RA 11h 48m 33.63s, Dec +01° 28' 59.4"), so the observing season for mid-northern latitudes begins in December (really early in the morning). Beware, it's about mag 17.

Pi-03 Ori (π -3 Ori) (F6V, V 3.2) is one of the chromospheric activity stars that Dr. Kafka mentioned. This should be an interesting target for long-term monitoring with medium-resolution spectroscopy.

HD 2421 is a challenging spectroscopic binary (single-lined) with ΔV of about 35 km/sec. It is nicely placed for evening observation in Andromeda.

AAVSO Alert:

CI Cam is the B[e] optical counterpart of a HMXB system. and was reported in outburst in early October. AAVSO Alert Notice 559 requests BVRI photometry in support of high-res spectroscopy being done at La Palma. This appears to also be a fruitful target for amateur-scale spectroscopy (see the ARAS website for recent amateur spectra). It is nicely placed for evening observations, at R.A. 04 19 42.14 Dec. +55 59 57.7. Photometric and spectroscopic observations should be continued through January, 2017.

Double Stars: Alan Whitman's article in Sky & Tel (November 2015, p. 53) mentioned two visual double stars that are not in the WDS, and for which no accurate relative astrometry (ρ , θ) has been reported. These are both in Aries, well-placed for this season.

GSC 1212-0409 is reported to be a pair of nearly-equal mag 12.5 stars, with separation of about $\rho \approx 6$ arc-sec.

GSC 1212-0301 is reported to be a pair consisting of a mag 10.3 and mag 10.6 stars, with separation of about $\rho \approx 7$ arc-seconds.

Could one or two of you measure (ρ , θ) for these, and report them via JDSO?

Transients: Obviously, these are targets of opportunity. Use your slitless spectrograph (e.g. SA-100 grating) to classify faint transients reported on the ATel, ASASSN, and CBAT transients web pages. Several projects should be triggered when a bright-enough nova or SN appears: confirmation of the transient, photometric monitoring, and spectroscopic examination.

The best way to start is to try your hand when the next reasonably-bright transient appears.

Refer to Dr. Martin's presentation at the SAS-2016 "Small Telescope Spectroscopic Research" workshop for practical tips about observations and data analysis.

ACP Expert at the Great Basin Observatory

by Bob & Stephanie Denny, DC3 Dreams

Culminating a 4 year process of securing funding, operating budgets, National Park Service permissions, and only 5 weeks' construction (!!) the Great Basin Observatory was dedicated and opened for first light in late August.

In late August we traveled to the Great Basin National Park (eastern Nevada) for the dedication of the Great Basin Observatory on Thursday Aug 25. The dedication was part of the celebration of the 100th Birthday of the National Park Service.

The Great Basin Observatory, a research-class astronomical observatory, is the first of its kind in any national park. The Great Basin Observatory opens up the darkest night skies in the United States to scientists, and future scientists, by conducting world-class research and education, thanks to the Great Basin National Park Foundation and their donors.

The Great Basin Observatory uses ACP Expert as its way of providing these services to users all over the nation. The observatory will not host visitors, all operation is remote.

A video of the ribbon cutting ceremony can be viewed at <https://www.youtube.com/watch?v=p5qLnHqqvM>

Lead Engineer Paul Gardner in the control room. This building will be unattended and locked except for servicing. All operations will be remote.



In the screenshot above, Lead Engineer Paul Gardner of Observatory Systems (on the very right) opens the dome via his mobile phone using ACP Expert's mobile/web System Status applet.

A great crowd was on hand for this event. Here's a little idea of the venue and some of the people.



Left to right below are: Bob Denny of DC-3 Dreams, Jason Simison of Sea West Enterprises (architects and contractors), Becky Mills, former Superintendent of Great Basin NP and lead organizer of the Foundation, and Lead Engineer Paul Gardner of Observatory Systems (design, assembly, and testing).



VirtualBox™ and Pipelines for Data Reduction

by Wayne Green (July 2016)

SAS 2016 prompted a serious discussion about pipeline architecture back at home. We heard Benjamin Shappee from ASAS-SN comment on their pipeline experiences, and I have three pipelines under development right now. My assessment is that you cannot write a general pipeline. One may generalize parts of it, but not generalize the whole thing in one step..

For a single machine it makes no real sense to break a pipeline down into a series of small programs. Multiple cores (more common now than in the past) and distributed machines still carry merit for small programs. It does make sense to make a source-code development toolkit (SDK) to accelerate pipelines.

OK... presume we've made a pipeline... based on IRAF... how do we support it? There are two new ways: Virtual-Box and "Containers" like Docker. Here we will examine a comprehensive way to get IRAF into a VirtualBox and retain some development capability.

IRAF is essentially a series of small programs. While this may slow down the calculations, the slow, methodical and careful steps to dig the subtle nuances of the science out of the data are important. There is a lot to be said for a careful approach. On the other hand, if you have an archive of 43,000 images to reduce (for example), then something may have to give.

So let's build a VirtualBox using CentOS 6.8, add in some useful tools from the Linux distribution, install Ureka 1.5.2 release (not Astroconda), add in scripts and source code for processing. The article is designed to act as a guide to refer back to after reading the whole article. Read further down into the article to pick up a little deeper detail of each step.

Appendix B below goes into an expanded discussion of the history and players surrounding this topic.

1. Overview

This section contains a checklist/outline of things to download and the order we will be working with them.

Before things get too confusing, I will call the main operating system of the host computer "**OS**". I will call the second "virtual" operating system you load and run within Virtual-Box "**VOS**". VirtualBox is controlled by the VBox Control Panel (**VBoxCP**). I am using Ubuntu 14.04 as my "**OS**" and installing CentosOS 6.8 as my "**VOS**". They both run on the single base hardware at the same time. RHEL stands for Red Hat Enterprise Linux owned and maintained by Oracle Corp.

The main players are the main OS, VBox, VBox's VOS, and VOS environment, Ureka, and Ureka's 32-bit environment, IRAF and the IRAF uparm files. All are compartmentalized and therefore easier to control. The 32-bit compatibility environment has to be added to the VOS to make IRAF work.

1. Create a folder on your machine to hold all these

downloaded files. You should only have to download them once. Make this a shared folder.

2. Download CentOS 6.8 for your system. <http://ssb.stsci.edu/ureka/1.5.2/docs/downloads.html#ureka-core-binaries>

3. Download VirtualBox and the matching extension pack from Oracle.com for your operating system. This is the wrapper that will interface the guest OS (VOS=CentOS 6.8) to your OS.

4. Download the Ureka installer for 1.5.2 and the matching installer install_ureka_1.5.2.

5. Download the Ureka tarball to match 1.5.2 installer.

6. Install VirtualBox and start it up.

- (a) Tie in the matching extension pack. (Known to all VBoxes) under this install. VBoxCP:

File->Preferences->Extensions and choose the extension pack from the shared folder.

On the VBoxCP, with the "New" button, create a new VBox. **Section 1.3** will go into details, but here are the steps:

1. Enter the name for this VOS, something like CUreka for CentosUreka.

2. Choose "Linux" and "Red Hat64-bit"

3. Next

4. Memory Size: be generous. I used 1/4 of 32G of main memory.

5. Next

6. Hard Disk

- (a) Create Virtual Disk Now

7. Create Button

- (a) VDI (VirtualBox Disk Image)

8. Next

- (a) "Storage on physical hard disk"

- (b) "Dynamically Allocated"

9. Next

10. Choose where VBox images will reside by default. I use "/home/wayne/VirtualBox VM" (not a shared folder)

11. Set this disk size to be pretty big. Temporary files take up room.

12. Done

At this point there is an empty VBox with all the parameters needed to work the host OS and no VOS.

1.1 The Base OS

I chose to test this article under Ubuntu 14.04 LTE. I chose this package solely in order to mirror a client's work. For other (new) applications I have been recommending the CentOS 6.8 release of late. It fixed a few of the bugs in 6.7 version, and it is stable. (The move to the 7.x releases by RHEL and CentOS breaks quite a few things, so I don't yet recommend it).

You want a .iso image of the Centos OS you will use. You can download one or more from a mirror-site recommended at <https://wiki.centos.org/Download>. The end-of-life for the Centos 6.x suite is 2020, so we are good for quite a while.

You want VirtualBox from <https://www.virtualbox.org/wiki/Downloads>. Get the extension pack Oracle_VM_VirtualBox_Extension_Pack-5.1.2-108956.vbox-extpack to match your VBox download. (I'm obviously using version 5.1.2).

You may also want the User Manual, which is available at <http://download.virtualbox.org/virtualbox/5.1.2/UserManual.pdf>

I created a directory /home/wayne/centos to keep all the parts I need to (re)make a vbox. This includes the .iso files for the OS and the tarball for Ureka.

The Ureka tarballs are downloaded from:

<http://ssb.stsci.edu/ureka/1.5.2/docs/downloads.html#ureka-core-binaries>

or

<http://ssb.stsci.edu/ureka/1.5.1/docs/downloads.html#ureka-core-binaries>

They install with the installer scripts `install_ureka_1.5.x -t <tarball>` where <tarball> is the long-winded tarball file name.

Because I've done a bit of customization for IRAF and for ds9, I put those files into the /home/wayne/centos directory. I have Python scripts, so I save them here too. Anything that wants to get on the bus to VBox land needs to be put in the /home/wayne/centos directory.

1.2 Loading the Guest OS into VBox on Your Base OS

Download and install VirtualBox for your Host OS.

Then, do two short routines, to (a) set up an "escape hatch" from your VM, and (b) provide support for modern USB devices.

(a) "Escape Hatch": Start the VBox control panel, then under

File->Preferences->Input->Virtual Machine Host Key Combination

find the most under-utilized key on your keyboard that this program will see and use that key. The default is the right shift key, but it doesn't work on all keyboards. The best approach is to not rely on the default, but instead make your own selection of the "escape hatch" key. Just type a candidate key. If VBox sees the key, its name will appear in the entry box.

The point of creating this setting is that with it, I can "release" my keyboard and mouse from a VBox VOS while it is running, so I can do other things in the main OS.

Why doesn't the default escape-hatch key always work? The Apple Keyboard uses its own special codes instead of PC-104 codes in lots of irritating places. For example, my full-sized Apple aluminum keyboard does not return a recognizable key-code for the right-shift key entry, so I use the "end" key in the middle tier of keys on my keyboard). Dell keyboards have a similar oddity. In a few steps, this will not really matter. But for now, it's a good idea to create an escape hatch.

(b) USB support: Before you do anything else, download and install the Oracle VM VirtualBox Extension Pack. Under the main **VBoxCP**, File->Preferences->Extensions, use that real small pull-down to go into your host file system to find the extension pack – chose that and hit OK. This enables USB and lots of other basic things. It kinda punches a small and very useful hole between the VBox part and your main system. We will use that hole to tie into the underlying "OS" file system, where our data files reside.

1.3 New VBox guest OS

Now to load the CentOS 6.x .iso media image. Use the **VBoxCP** "New" button. A dialog pops up, where you enter a name for this guest OS, something like "CentosUreka". Choose "Linux" as the type, and "Red Hat (64-bit)" as the Version. Use "Next" button to proceed.

Set the "Memory size". I use 1/4 of my machine's main memory (8192 MB). IRAF likes its memory! Click Next to proceed.

The next panel is "Hard disk" and I used the default "Create a virtual hard disk now" then "Create Button" to proceed. I chose the default "VDI (VirtualBox Disk Image)" and Next to proceed. Under the "Storage on physical hard disk" I choose a "Dynamically Allocated" and Next to proceed.

On the next panel, I left the location in its default "/home/wayne/VirtualBox VM" but set the size to be real big – 200GB. Then "Create" to proceed.

I now have a new VBox under the control panel. It is called CentosUreka1 and it is empty waiting for the OS .iso image.

The next steps are to work on that VBox directly – to associate it with the .iso disk image etc.

Double-click CentosUreka, and a VBox pops up, saying it is "Powered off" together with a "Select start-up disk". There is a little icon of a folder with a little green-arrow off to the right side of this window. Choose the .iso and continue. Now the VBox starts, loads that .iso into the VBox and things start looking like you are building the new machine. The window says to "load in xxx seconds, hit return". An old fashion load-window pops up to "test the media"; here, you really want to use the "Skip" feature and load the media directly. A GUI for the VOS takes over the VBox to start your download.

At this point you have to click the mouse twice: once to select the window of VBox, and the second time for VBox to capture your mouse and keyboard. This is where that Host Key escape plan we made earlier comes in handy. I follow

the prompts and pick English, U.S. English, choose "Basic Storage Devices", choose (scary) "Yes delete data", then give the VBox a "hostname" like "cbureka". In the next step I choose America/Denver as my timezone, give myself a decent root password, tell this VBox to "Use All Space", and tell it to write to disk.

Things now get serious. I choose basic desktop, and down at the bottom, I check the "Customize Now" button. I make my selections:

1. Applications:
 - (a) check Emacs (if you want to)
 - (b) uncheck Office Suite Productivity (to save space)
2. Base System (take the defaults)
3. Databases
 - (a) check PostgreSQL client (I rely on this in pipelines)
4. Desktops
 - (a) Uncheck Fonts (to save space)
 - (b) Make sure the X-Window options are ON (default)
5. Development
 - (a) Check Additional Development, Desktop Platform Development, Development tools.

Most users will leave the rest as defaults. You may read them to become familiar with things, maybe choose a few.

Next to proceed. This will start loading the VOS and all its support code, packages, programs etc. that are available via the default. This will take a while – it is making a new (virtual!) machine, after all. There are progress bars, etc., to amuse you as it goes along.

It finishes (about 5 minutes on my machine) and asks for a reboot. This will reboot the VBox. If this were a non-virtual case you would have reached the point where the temporary OS loader has finished and is turning control over to the new machine. So Reboot.

The new boot sequence starts, the "screen" looks just like a real CentOS boot.

You are at the point where you are creating a user, and working down to the Kdump question. I take the defaults for now, make "wayne" as the user and fill in the rest in a basic way. I disable the Kdump mechanism, as this is a virtual machine. The reboot finishes and asks me to login as a user!

I want to feather the VOS nest.

System->Preferences->Screensaver: set to a long time.

System->Preferences->Windows: I like to select windows when the mouse moves over them but not raise the selected window after an interval. Also I set "Title Bar Actions" to roll-up.

System->Preferences->Keyboard->Layouts->Layout Options (Button)...Ctrl key position and check the Caps Lock as Ctrl (relieve carpal tunnel syndrome).

System->Preferences->Screensaver and set the time way-out. Your main system screen-saver preferences will apply. This VBox goes to sleep after a few minutes. Annoying.

Right-click on the "Panel" (the stripe across the VBox screen that has the Applications Places etc. on it) and choose "Add to Panel". On the popup, near the bottom you find a "Shut Down" button. Drag this up to the Panel. Add a few others if you want them.

2. Get It All Together

As the hand-grenade manual sez right after "step 1 – pull the pin, step two – release safety spoon" – you come to "But First"!

2.1 Feather the Nest

I want to hook up to a directory somewhere. I have several physical disk drives on my system. Under *inx, for the most part all drives are located under one master point for my computer called the "root" directory at the location /. Every thing "mounts" to that point. So I have a tonne of images at /usr2/Astrodata. I have a sub-directory there called VBox. This is the place I will share all the common things I want on all my VBoxes, like my special ds9 ini environment, a tarball for my Emacs development environment, etc. These are copies, and can not be hurt. I'll copy new ones as I need to.

Just outside of the VOS, along the top of the main window, there is a menu-pull down called "Devices". At the very bottom there is an item "Insert Guest Additions CD Image". Use that. An .iso file will appear on the desktop. Double-click this disk image, and a dialog will pop up. Answer the questions, let it run, then stop and restart the VBox image. This is the safest way.

Under your unix host, you must allow a directory within your file system to be shared.

For NFS follow the OS directions. I have an entry in my /etc/exports that lets a machine on my local subnet connect:

```
/usr2/Astrodata
192.168.0.0/24(rw,nohide,insecure,no_subtree_check,
async)
```

```
mount -t vboxsf astrodata /media/astrodata
```

In the VBoxCP, under File->Preferences->Extensions you can load the extension pack we mentioned before.

```
VBoxManage setextradata "VM name" VBoxInternal2/SharedFoldersEnableSymlinksCreate/sharename 1
```

With the share working, I installed my /home/emacs major directory, and edited/added vboxaliases to the main directory. I then shutdown the VBox. I right-click on the CentosUreka image, choose Clone, do a full clone. This takes a while, the reserved disk area is pretty big. Choose a name for the clone, something like CentosUreka2. Keep cloning as you make major alterations.

2.2 Clone the VBox

Let's clone this VBox. It is clean and mostly to our tastes. I want to drag over my aliases, ds9 ini files, Emacs support files (decades in the making) and my scripts to make pipelines. All this and more lives sprinkled around between machines and the cloud. I include this here to emphasize I don't want to do any of that in this clean pretty VBox.

In this section we'll bounce around a little bit. First we don't want to load our data into the VBox – let the data reside on the main OS system. So we need to tell Linux to link down to that file system. To do this we want to create a mount

I use Emacs and development tools to make sure I get the compilers etc. I have special utilities in Python and C/C++ I use. I have to port those into the VBox later.

The next step asks to permit kdump. This is not a bad idea on a real machine, but I say no and suffer a reboot step to remove that from the VBox. It makes things easier on everyone.

Before I do anything, I want a "Snapshot" of this clean install. If I mess up, I can return to the Snapshot and start over. When it comes to backups, there are two kinds of people: those who do and those that will.

The whole reason I started this mess was to get IRAF via Ureka. So I will load Ureka and then take a "snapshot" of my new system for next time.

OK, I download the deprecated installer (I don't like Astro-conda just yet – it's real new and still getting better). I run the installer with the `.install_ureka_1.5.2` (the bash command `<dot>` means source so I use the current shell to do the trick). and discover that there isn't enough room. So we power down the VBox and resize it on my main host machine:

```
VBoxManage modifyhd "/home/wayne/VirtualBox
VMs/Centos67/Centos67.vdi" --resize 16384
```

2.3 Shared Folders

Virtual Machines are designed to isolate the underlying host from Bad Things™. For some purposes, you must relax the restrictions, and one very important purpose is "Shared Folders".

I maintain my data on two separate disk drives. The RawData and derived products on one, and developing analysis products on another. This gives me a mirror on the local machine. So I want to share the disk space. This keeps the virtual image size down.

3. Loading Ureka

You can grab copies of the two tarballs 1.5.1 and 1.5.2. The installer will work with `install_ureka_1.5.x -t <tarball>` to save the repeated downloads of the tarball. You can hack the install script to prevent deleting the tarball at the end.

4. Feathering IRAF's Nest

There are a few files that make IRAF easier to use. They are in the `~/iraf` directory, and are called `login.cl` and `loginuser.cl`. The most important thing to remember about these files is that the last line should read "keep". Stare at these and the documentation and you will learn a lot about IRAF.

There are a few small files that you may find useful to edit and replace into the Ureka/ subdirectories. For example:

I added these three sections to `/home/wayne/Ureka/iraf/noao/lib/obsdb.dat` for the University nearby, my home observatory, and some Subaru data I've been working with:

```
observatory = "sbo"
name = "Sommers Bausch Observatory" longitude =
-105.55926
latitude = +40.00372
altitude = 1653
timezone = 7
```

```
observatory = "gao"
name = "Greens Acre Observatory" longitude = -
105.20416666666667
latitude = 40.0905555
altitude = 1555
timezone = 7
```

```
observatory = "subaru"
name = "SUBARU Telescope, NAOJ" longitude =
155.4706
latitude = 19.8255
altitude = 4163
timzone = 10
```

I add these lines to `iraf/loginuser.cl`

```
task $fvl = ("$(/home/wayne/bin/fitsverify -I $* |
less)")
task $r = "$ ( ls -lt $* | grep -v '[ ][-.]' |
head -10 )"
task mll = "$ ( ls -l *fits > l.l)"
task $fitsls="$foreign" task $fitsgrep="$foreign"
task $ds9="$foreign"
task $iraffind="$foreign"
```

load some packages at startup time

```
noao
imred
ccdred
```

set up for spectra analysis

```
twospec
apextract
onedspec
```

set up to do interactive images

```
images
tv
```

This lets me type "fvl" at the pyraf command prompt and have a sub-program run that is the `fitsverify` program from `Hear-sarc`. The "r" command is short hand to do a `ls` for recent files in a directory; and the "mll" means `make l.l` to put all the fits files into one file so I can work on them all at once with the IRAF '@' facility.

The “fitsls” and “fitsgrep” are python scripts I wrote, and “iraffind” is a bash script to locate files in the IRAF release that match a pattern.

The rest loads packages for me before the first command prompt so I can hit the deck running.

5. Summary

In this brave new world of computing, with cloud data and data-appliances, making use of VirtualBoxes and Containers will be the new norm. On one workstation, you can run three different operating systems – each tailored for its own task. This little exercise in bringing up a VBox with CentOS 6.8 for IRAF gives you the most opaque tool for astronomical data analysis out there!

Appendix A: Escape Plans

The Preferences->Input->Virtual Machine submenu allows you to select the “Host Key Combination” shortcut where you can hit odd keys until one that you seldom want to use will register. The “right shift” is the default. For my Apple Aluminum keyboard, I found the “end” key in the center cluster of keys would be recognized by the VM. That is the key I decided to use.

It is a good idea to be tight with your OS. VBox will capture the mouse and getting it back can be a problem. I use an Apple White large USB keyboard, and the keymap is not consistent. This can be an issue. Under Ubuntu the Option+Control+F1 gets me to a plain console window. I log in as root and use the kill command to get rid of all the tasks associated with the VBox. Then Alt-Control-F7 to get back to the window. This rudely kills the VBox session.

The next trick is to right-click on the top panel and add the shut-down button. This will allow you to get control of your system back, when all the key/mouse capture events are consumed by VBox.

Whilst using VBox, hitting the control key on the right side of the keyboard (the one that we righty’s never quite use) will release the mouse/keyboard back for use with the support system (Windows or my other Linux distro). Gotta handle this some way, and this is it.

Under File->Preferences->Input there is a little checkbox “Auto Capture Keyboard”. Check that box to get the Right-Ctrl thing to work.

A.1. Shared Data

Under the Settings->Shared Folders, add the name of the disk/directory you want to use. With my Linux box under /usr2/Astrodata I mount as /astrodata on the CentOS67 image. So I first make the /astrodata mount point, then use the command:

```
mount -t vboxsf -o uid=$UID,gid=$(id -g) share /astrodata
```

to achieve the mount.

Appendix B: Real Obscure Advice

The Ureka 1.5.1 installer, asks a few questions, goes to the net and grabs a tarball, installs all the stuff, then DELETES that tarball. When building a VM it is good to just have – and keep – the tarball. So, hack the install script to keep the tarball around.

System->Preferences->Screensaver - set to long time.

System->Preferences->Windows - I like to have it select a window when the mouse moves over it, but not raise selected window after an interval. Also I set “Title Bar Actions” to roll-up.

System->Preferences->Keyboard->Layouts->Layout Options (Button)...Ctrl key position and check the Caps Lock as Ctrl (relieve carpal tunnel syndrome).

B.1 Enter VBox for Linux and Ureka

VirtualBox™¹ is used in remote telescope operations, i.e. to allow DC3 Dream’s ACP to use two ‘separate’ machines to support more than one camera, etc. It allows seamless use of Linux from within Microsoft Windows OS based computers. While Microsoft licensing is a nightmare, requiring you to port your licenses to each VBox, installing VirtualBox using Linux is a breeze. The small technical downside is the introduction of a little bit of latency between the decision-making actions of the software and the firmware in the mounts and cameras you want to control. A careful compromise is usually worth it.

Ureka² is a complete roll-up of Python 2.7 and IRAF, together with useful tools like ds9 and the more-or-less complete StSci suite of tools. It even includes Ximtool. Ureka is now migrating to live under AstroConda – it is not perfectly settled down yet, but will be great when it gets there. The good news is that IRAF is being better maintained. AstroConda is a package that lives under Anaconda (aka conda), a platform-agnostic release of big-data analytic tools from Continuum.io. Conda contains the R statistics package for example.

The ‘nix³ environment is very mature and robust. It lacks MS Office™, so it relies on LibreOffice. There are essentially three *nix flavors remaining: The first, and hardest to use, are the Berkeley System Division (BSD) based releases. The BSD OS has to be rebuilt to extend, and drivers are not as plentiful, but BSD is robust in a lot of ways. The second flavor that remains close to the original Unix System Laboratory’s genetics, includes Solaris – via Sun via Oracle – and has resulted in the OpenSolaris project. Oracle also owns and operates Red Hat Linux (aka RHEL or Red Hat Enterprise Linux). CentOS is a RHEL based release.

Linux is the third (somewhat radical) departure from the original Unix genetics. It was created by Linus Torvalds back in the 1990s as one of the truly great experiments in community based software collaboration. You find Unix genetics under Microsoft Window’s hood – commercialized to make them unique while distancing themselves for marketing/commercial reasons. Today, BSD Unix via Steve Job’s Next Inc’s ideas still underpin Apple OS/X while they are migrating to iOS Linux releases to distinguish themselves by the update mechanism they use. The RHEL releases use

“yum”⁴ and rpm files while Debian releases use “dpkg” with front-ends like aptitude with its shorter apt-get mechanism.

One should buy a “computer” based on the decisions about the software you want, then choose the operating system, then choose your hardware. After all, the software is what makes things happen. In the general market, the hands-down overall choice of software is Microsoft Office, hence Windows as the OS and “WinTel” for hardware. It is a huge market with the usual savings that scale brings along.

Currently, the market is driving hard into cloud based storage and computing, and to use a hand-held “data appliance” as the means to control it all. The PMT (Processor-Memory-Transducer) idea from Peter Freeman’s early work in “Software Engineering” (... use the way-back machine, Mr. Peabody ...) was incredibly prescient in this department. My phone can control my telescope located across the world. It is as fascinating as it is frightening!

footnotes

¹Oracle Corp.

²Google as ureka STSci

³The history of Unix/Linux and other specially named unix-like environments have been nicknamed <wildcard>nix by the community. The most unholy of Holy Wars™ can and do take place on a regular basis.

⁴Yellowdog update manager

SAS Leadership

Corporate Officers:

Bob Buchheim – President
Robert Stephens – Treasurer
Robert Gill – Secretary

Newsletter Editors:

Dale Mais
Robert Buchheim

Sponsor Liaison:

Jerry Foote
Cindy Foote
SASLiaisons@gmail.com

Registration:

Margaret Miller

All SAS Leaders are volunteers, serving without compensation.

Advisors:

Dr. Arne Henden
Dr. Alan W. Harris
Dr. Dirk Terrell

Membership Information

The Society for Astronomical Sciences welcomes everyone interested in small telescope astronomical research. Our mission is to provide education, foster amateurs' participation in research projects as an aspect of their astronomical hobby, facilitate professional-amateur collaborations, and disseminate new results and methods. The Membership fee is \$25.00 per year.

As a member, you receive:

- Discounted registration fee for the annual Symposium.
- A copy of the published proceedings each year, even if you do not attend the Symposium.

Membership application is available at the REGISTRATION page of the SAS web site: <http://www.SocAstroSci.org>.

The SAS is a 501(c)(3) non-profit educational organization. Your Membership dues and donations may be tax deductible.

SAS Contact Information

9302 Pittsburgh Avenue, Suite 200,
Rancho Cucamonga, CA 91730

On the web:

www.SocAstroSci.org

Program Committee:

program@SocAstroSci.org

Membership:

Robert Stephens:
rstephens@socastrosci.com

Newsletter:

dmais@socastrosci.com
Bob@RKBuchheim.org

