Observation Planning

Questions to ask when planning a photometric observing session ("run"):

Where is my target?

Index

Let's pick a few exciting targets to look at. These are gigantic evolved stars, Luminous Blue Variables and Yellow Hypergiants and Supergiants:

1. P Cygni
2. Rho Cassiopeia
3. HD 168607
4. Polaris

To find where these sources are on the sky, we can query a few databases, including the SESAME name resolver service (http://cds.u-strasbg.fr/cgi-bin/Sesame (http://cds.u-strasbg.fr/cgi-bin/Sesame)) and the SIMBAD service. We can do this using `astropy` , the main library for astronomical analysis.

In [1]:
```
from astropy import coordinates, units
```

In [2]:
```
# this uses SESAME
pcyg_coord = coordinates.SkyCoord.from_name('P Cygni')
pcyg_coord
```

Out[2]:
```
<SkyCoord (ICRS): (ra, dec) in deg
    (304.44667535, 38.03293131)>
```

We can display this in other ways too:

In [3]:
```
pcyg_coord.to_string('hmsdms')
```

Out[3]:
```
'20h17m47.202084s +38d01m58.552716s'
```

The SIMBAD service gives us a little additional information:

In [4]:
```
from astroquery.simbad import Simbad
```

In [5]:
```
pcyg_simbad = Simbad.query_object('P Cygni')
pcyg_simbad
```

Out[5]: *Table length=1*

MAIN_ID	RA	DEC	RA_PREC	DEC_PREC	COO_ERR_MAJA	COO_ERR_MINA	COO_ERR_A
	"h:m:s"	"d:m:s"			mas	mas	
object	str13	str13	int16	int16	float32	float32	
* P Cyg	20 17 47.2020	+38 01 58.552	11	11	0.120	0.130	

We'll just use SESAME for the others, though. Fill in the SESAME approach for `rhocas` (Rho Cassiopeia), `hd168607` (HD 168607), and `polaris` ("Polaris"):

(`coordinates.SkyCoord.from_name` uses SESAME to obtain the object's coordinates)

In [6]:
```
# replace this cell with your answer
# the variable names should be those used in the next cell

rhocas_coord = coordinates.SkyCoord.from_name('rho Cas')
print(rhocas_coord)

hd168607_coord = coordinates.SkyCoord.from_name('HD 168607')
print(hd168607_coord)

polaris_coord = coordinates.SkyCoord.from_name('Polaris')
print(polaris_coord)
```

```
<SkyCoord (ICRS): (ra, dec) in deg
    (358.59597498, 57.4993797)>
<SkyCoord (ICRS): (ra, dec) in deg
    (275.31202306, -16.37548903)>
<SkyCoord (ICRS): (ra, dec) in deg
    (37.95456067, 89.26410897)>
```

In [7]:
```
print(pcyg_coord.to_string('hmsdms'))
print(rhocas_coord.to_string('hmsdms'))
print(hd168607_coord.to_string('hmsdms'))
print(polaris_coord.to_string('hmsdms'))
```

```
20h17m47.202084s +38d01m58.552716s
23h54m23.0339952s +57d29m57.76692s
18h21m14.8855344s -16d22m31.760508s
02h31m49.0945608s +89d15m50.792292s
```

When can I observe my target?

Index

Next, knowing where these are on the sky, we can ask what times of year they are up at night.

We first have to define where *we* are by setting the observatory location.

In [8]:
```
from astroplan import Observer
from astropy import units as u # shortcut
```

In [9]:
```
# sometimes we have to update the database; if that's necessary, a message will p
from astropy.utils import iers
iers.conf.IERS_A_URL = 'ftp://cddis.gsfc.nasa.gov/pub/products/iers/finals2000A.a
iers.conf.IERS_A_URL_MIRROR = 'https://datacenter.iers.org/data/9/finals2000A.all
from astroplan import download_IERS_A
download_IERS_A()
```

In [10]:
```
CTO = Observer(location=coordinates.EarthLocation(lat=29.643018, lon=-82.349004*u
               timezone='US/Eastern',
               name='University of Florida Campus Teaching Observatory',
              )
CTO
```

Out[10]:
```
<Observer: name='University of Florida Campus Teaching Observatory',
    location (lon, lat, el)=(-82.34900399999998 deg, 29.643018 deg, 30.99999999
950558 m),
    timezone=<DstTzInfo 'US/Eastern' LMT-1 day, 19:04:00 STD>>
```

Before we go on, we can use some information we already have in hand to determine approximately when we can observe these objects. It is important to do this sort of qualitative 'sanity check' to ensure that the numbers coming from the computer are reasonable.

The March equinox defines the zero-point of the Right Ascension coordinate, which means that 12h RA goes overhead at midnight (0h RA goes overhead at noon).

Questions to Answer

Index

There are 24 hours in a day and 12 months in a year. By how many hours does the sky rotate each month?

A: Stars rise 4 min earlier each day. 4min(30 days) = 120 min = 2 h

How many degrees?

A: Over the course of the month stars shift ~ 30 degrees

Given the above information, answer the following questions:

1. What is the Right Ascension of the Sun in the March (vernal) equinox?

A: 0 h

2. What is the RA of the Sun at the September (autumnal) equinox?

A: 12 h

3. During what season or month do our targets reach zenith at midnight? (one answer per target)

   a star is at zenith when dec = latitude Sky moves 2 h per month stars shift 30 degrees per month

A:

- P Cyg:

current: 304.44667535, 38.03293131

38 - 29.6 = 8.4 degrees late August early Sept

- Rho Cas:

current: 358.59597498, 57.4993797

57.5 - 29.6 = 27.9 so almost 30, Early August

- HD 168607:

current:(275.31202306, -16.37548903)

29.6 + 16.37 = 46 late october

- Polaris:

current: (37.95456067, 89.26410897)

Polaris never rises or sets so it will never be at our zenith

Having done these checks, we can proceed to get more precise answers.

We create `target` objects to observe at different times.

Answer questions with astroplan

Index

In [11]:
```
from astroplan import FixedTarget
```

In [12]:
```
pcyg_target = FixedTarget(pcyg_coord, name="P Cygni")
rhocas_target = FixedTarget(rhocas_coord, name="Rho Cas")
hd168607_target = FixedTarget(hd168607_coord, name="HD 168607")
polaris_target = FixedTarget(polaris_coord, name="Polaris")
pcyg_target, rhocas_target, hd168607_target, polaris_target
```

Out[12]:
```
(<FixedTarget "P Cygni" at SkyCoord (ICRS): (ra, dec) in deg (304.44667535, 38.
03293131)>,
 <FixedTarget "Rho Cas" at SkyCoord (ICRS): (ra, dec) in deg (358.59597498, 57.
4993797)>,
 <FixedTarget "HD 168607" at SkyCoord (ICRS): (ra, dec) in deg (275.31202306, -
16.37548903)>,
 <FixedTarget "Polaris" at SkyCoord (ICRS): (ra, dec) in deg (37.95456067, 89.2
6410897)>)
```

To determine the observability of an object, we ask whether the object is "up" (in the sky, as opposed to underfoot), and whether it is night time.

What time are you working on this notebook? Is it night? Are the target sources "up"? (answer these before filling out the code)

It is 4:30 pm, the stars are above the horizon

Since the answers to all of our questions depend on both the time of day and the time of year, we need to define our time.

In [13]:
```
from astropy.time import Time
```

In [14]:
```
now = Time.now()
now
```

Out[14]:
```
<Time object: scale='utc' format='datetime' value=2021-09-15 15:51:37.251635>
```

Are any of these sources "up"? Astroplan will tell us:

In [15]:
```
CTO.target_is_up(now, [pcyg_target, rhocas_target, hd168607_target, polaris_targe
```

Out[15]:
```
array([False,  True, False,  True])
```

At least some are "up" at the time of writing.

OK, but is it night?

In [16]:
```
CTO.is_night(now)
```

Out[16]:
```
False
```

For planning purposes, we care more about what the future holds and how we're going to schedule our time, so we want to know when sunrise and sunset are. We use `now` to tell the software that we care about sunrise and sunset *today*; we can put in future days to find out how sunrise/sunset times change.

In [17]:
```
CTO.sun_set_time(now), CTO.sun_rise_time(now)
```

Out[17]:
```
(<Time object: scale='utc' format='jd' value=2459473.4794238056>,
 <Time object: scale='utc' format='jd' value=2459472.9708817797>)
```

These aren't formatted in a very familiar way, so we can use a standard format - e.g., ISO, the International Standards Organization format - to show them more nicely:

In [18]:
```
CTO.sun_set_time(now).iso, CTO.sun_rise_time(now).iso
```

Out[18]:
```
('2021-09-15 23:30:22.217', '2021-09-15 11:18:04.186')
```

Those times look a bit funny because they're in UTC, Universal Coordinated Time.

Time Zones

Index

Time zones are tricky, so we rely on libraries to help us with them.

In [19]:
```
now.to_datetime(timezone=CTO.timezone).isoformat()
```

Out[19]:
```
'2021-09-15T11:51:37.251635-04:00'
```

That's a pretty verbose way to ask for and display this, so we can define a shortcut to get the time of day in the Eastern time zone:

In [20]:
```
def eastern(time):
    est = time.to_datetime(timezone=CTO.timezone)

    return est.strftime('%H:%M:%S')
```

In [21]:
```
(eastern(CTO.sun_set_time(now)), eastern(CTO.sun_rise_time(now)))
```

Out[21]:
```
('19:30:22', '07:18:04')
```

Now you can see that sunset is around 7:30 PM and sunrise is around 7:00 AM. Note that we have sunset first, sunrise second, because we care about the interval between the sun going down and coming back up again, not the daytime.

We need to know the sun's rise and set time to avoid it, but we also can ask when our sources rise and set.

In [22]:
```
(eastern(CTO.target_rise_time(time=now, target=pcyg_target)),
 eastern(CTO.target_set_time(time=now, target=pcyg_target)))
```

Out[22]:
```
('14:22:13', '05:55:41')
```

In [23]:
```
(eastern(CTO.target_rise_time(time=now, target=rhocas_target)),
 eastern(CTO.target_set_time(time=now, target=rhocas_target)))
```

Out[23]:
```
('15:29:37', '12:00:50')
```

In [24]:
```
(eastern(CTO.target_rise_time(time=now, target=hd168607_target)),
 eastern(CTO.target_set_time(time=now, target=hd168607_target)))
```

Out[24]:
```
('14:50:34', '01:35:46')
```

In [25]:
```
### Note: this will fail!  "Comment it out" and move on
((CTO.target_rise_time(time=now, target=polaris_target)),
 (CTO.target_set_time(time=now, target=polaris_target)))
```

```
WARNING: TargetAlwaysUpWarning: Target with index 0 does not cross horizon=0.0
deg within 24 hours [astroplan.observer]
C:\ProgramData\Anaconda3\lib\site-packages\erfa\core.py:154: ErfaWarning: ERFA
function "utctai" yielded 1 of "dubious year (Note 3)"
  warnings.warn('ERFA function "{}" yielded {}'.format(func_name, wmsg),
```

Out[25]:
```
(<Time object: scale='utc' format='jd' value=nan>,
 <Time object: scale='utc' format='jd' value=nan>)
```

What's going on with that last one? Since Polaris is the North Star, it is always observable from any northern observatory. It has no rise and set time!

Airmass Plots

Index

We can visualize the "up time" for the targets using an *airmass plot*, which plots the airmass versus time of night.

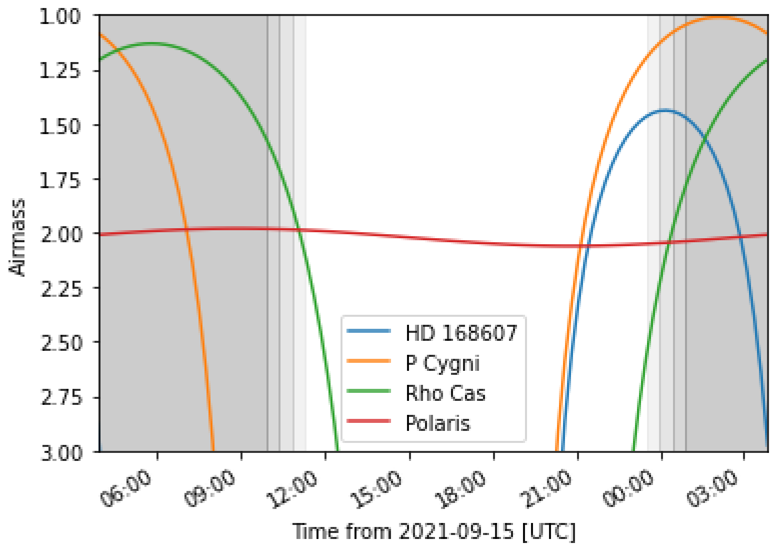
Airmass (https://en.wikipedia.org/wiki/Air_mass_(astronomy)) is the path length of air through which we're observing, and it is usually approximated as secant(zenith angle) and measured relative to zenith.

1 airmass is straight up, and is the minimum amount of atmosphere through which you can observe.

Generally, it is best to observe at airmass < 2 when possible. Airmass up to ~3 can be useful.

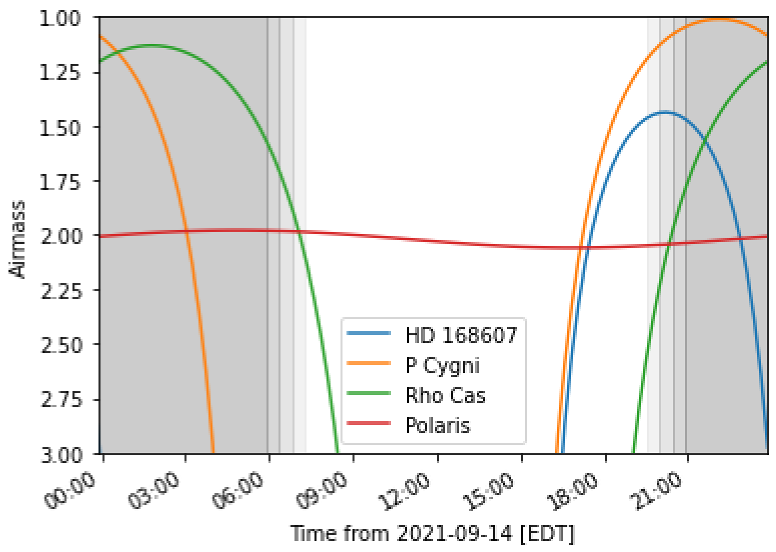
Beyond that, serious complications tend to arise.

In [26]:
```
from astroplan.plots import plot_airmass
import matplotlib.pyplot as plt

plot_airmass([hd168607_target, pcyg_target, rhocas_target, polaris_target],
             observer=CTO,
             time=now,
             brightness_shading=True)

plt.legend(loc='best')
plt.show()
```



Note that the X-axis is in Universal Coordinated Time (UTC). If you want the plots in local time, you can request that the local timezone be used (but only for certain versions of astroplan):

In [27]:
```
# if this fails, comment it out and continue
plot_airmass([hd168607_target, pcyg_target, rhocas_target, polaris_target],
             observer=CTO,
             time=now.to_datetime(timezone=CTO.timezone),
             use_local_tz=True,
             brightness_shading=True)

plt.legend(loc='best')
plt.show()
```



Answer some questions:

Which sources are up at night? For how long is each source observable at airmass $\lesssim 2$? (Polaris is close enough to 2 to consider it observable)

All stars are up for a period of time. HD is observavle <~ 2 airmass from a little after 17:00 to 00:00, so around 7 hours P Cygni is observable from a around 17:00 to about 5:00, so around 12 hours Rho Cas is observable from around 19:30 to 9:00, so around 14 hours and 30 minutes

Whether we can actually see these objets during their observable periods depends on if it is night or not.

How do I know when I've found my target? Make a finder chart

Index

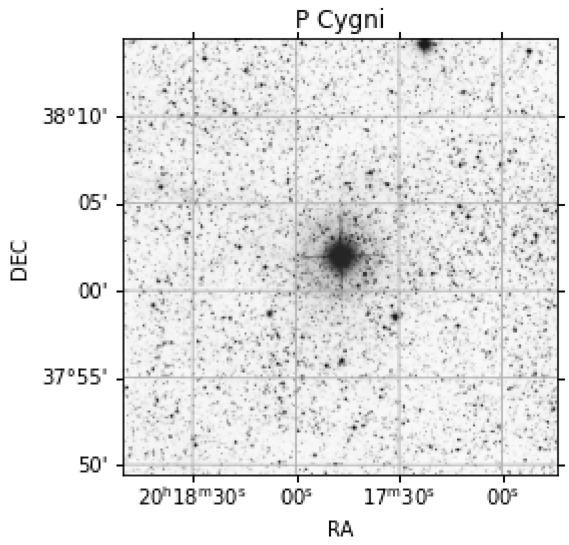
We need to know how to tell we're 'on target' when we've pointed our telescope at the right part of the sky. With binoculars and other very wide field-of-view instruments, we can use known asterisms or constellations to navigate by, but with narrow field-of-view professional telescopes, we have to use fainter stars to aid our pointing.

We can obtain finder charts from the many public astronomical archives, like the Digitized Sky Survey, which carefully logged the whole sky in visible light images.

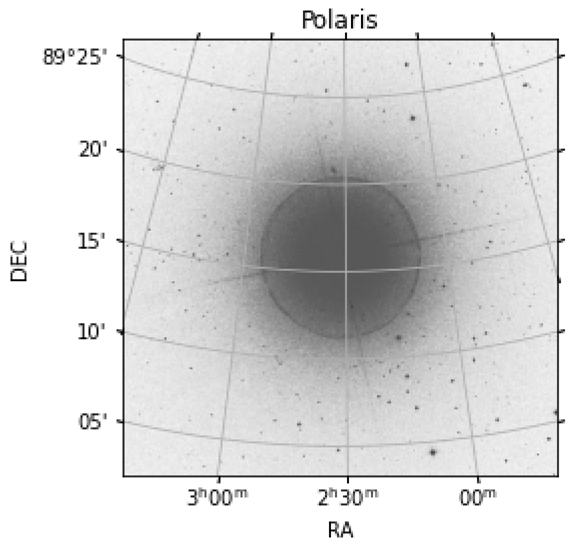
`astroplan` provides a means to obtain these images and plot them. It uses `astroquery`, another library, to query the `SkyView` image cutout service.

In [28]:
```
from astroplan.plots import plot_finder_image
```

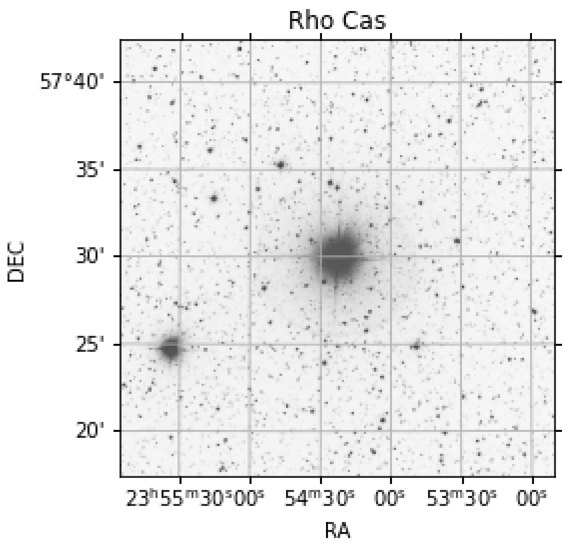
In [29]:
```
ax, hdu = plot_finder_image(pcyg_target, survey='DSS', fov_radius=25*u.arcmin)
```



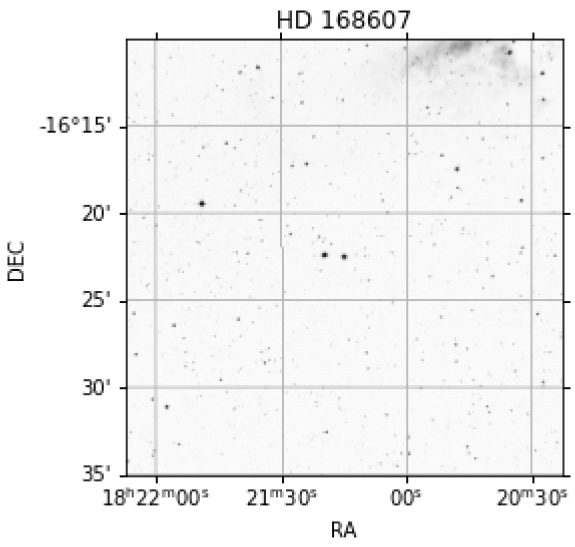
In [30]:
```
ax, hdu = plot_finder_image(polaris_target, survey='DSS', fov_radius=25*u.arcmin)
```



```
In [31]: ax, hdu = plot_finder_image(rhocas_target, survey='DSS', fov_radius=25*u.arcmin)
```



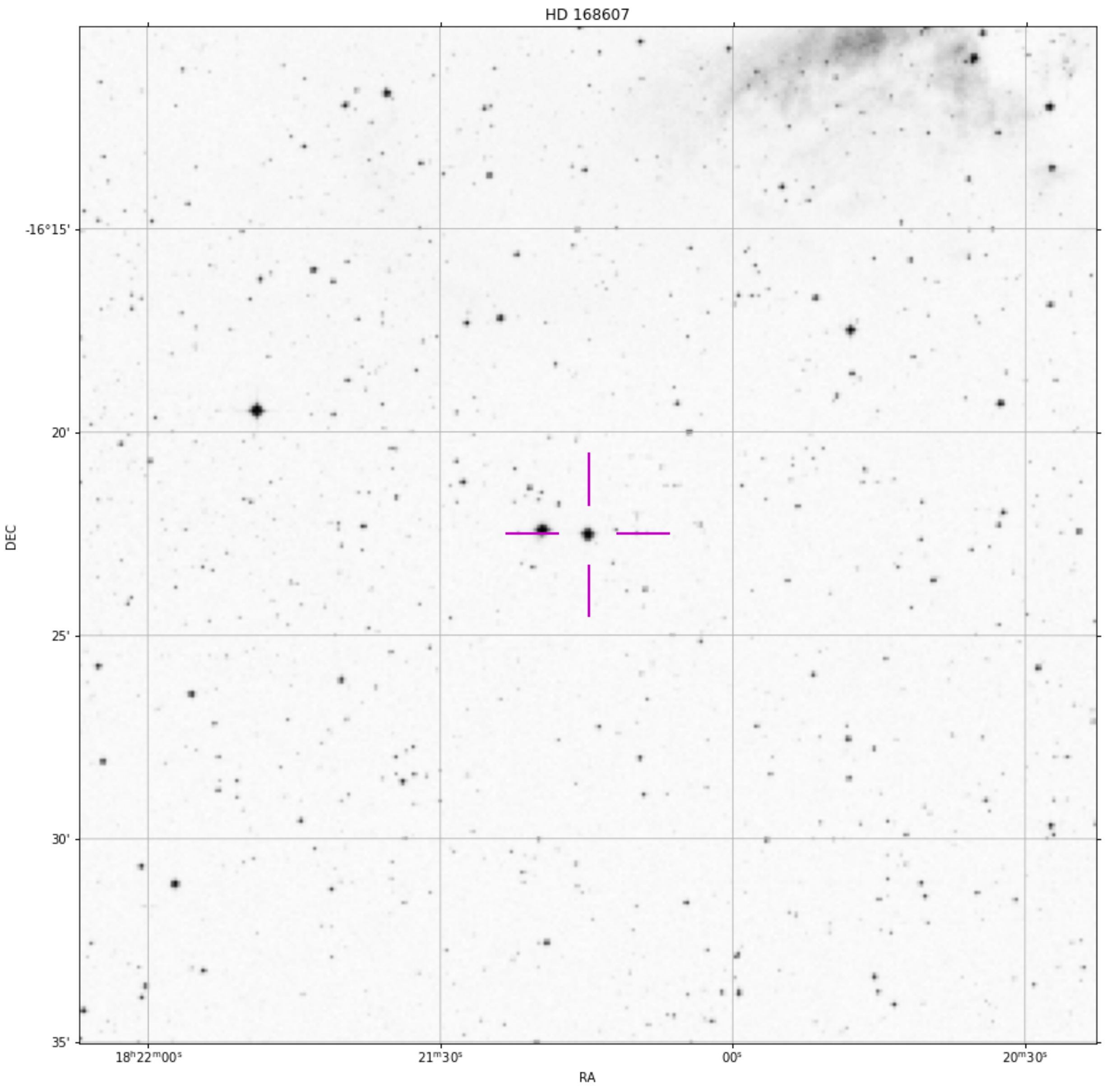
```
In [32]: ax, hdu = plot_finder_image(hd168607_target, survey='DSS', fov_radius=25*u.arcmir
```



You might notice these images come out a little smaller than you would like. You can control the figure size using matplotlib commands:

```
In [33]: import matplotlib.pyplot as pl
```

In [34]:
```
# specify the figure size in inches
pl.figure(figsize=(15, 15))
# a "reticle" is a set of crosshairs highlighting the target source
ax, hdu = plot_finder_image(hd168607_target, survey='DSS', fov_radius=25*u.arcmin
```



Planning for Clouds

Index

What if the night is cloudy? For professional observations, we will often write off cloudy nights as a lost cause, but for this class, sometimes we need to find the holes in the clouds and observe whatever's there!

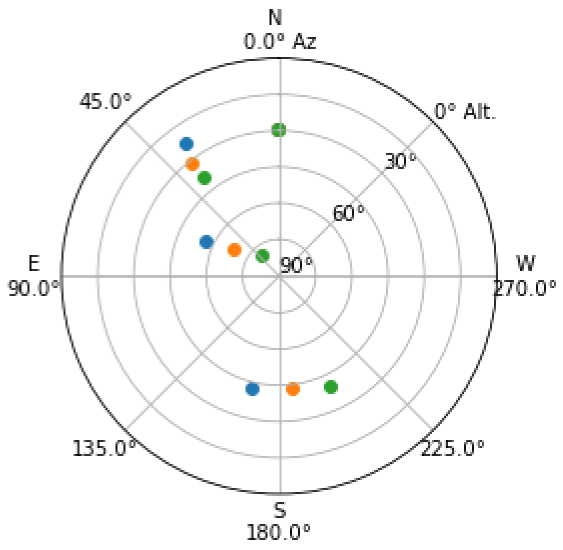
We can enable this by making sky charts.

In [35]:
```
from astroplan.plots import plot_sky
```

In [36]:
```
# first, let's find out where the objects are at sunset
sunset = CTO.sun_set_time(now)
# then, let's look 1 hour after sunset
onehour_after_sunset = sunset + 1*u.hour
twohours_after_sunset = sunset + 2*u.hour
```

In [37]:
```
targets = [hd168607_target, pcyg_target, rhocas_target, polaris_target]
plot_sky(target=targets, observer=CTO, time=sunset)
plot_sky(target=targets, observer=CTO, time=onehour_after_sunset)
plot_sky(target=targets, observer=CTO, time=twohours_after_sunset)
```

Out[37]:
```
<PolarAxesSubplot:>
```



In the star chart, the colors represent different times. One of the points only shows the final color (maybe you can see some others behind it). Why isn't it moving?

A: This star isn't moving because it generlly doesn't move trhoughout the night and is fixed on the night sky on the North Celestial Pole. (Polaris)

In [ ]:

EXERCISE

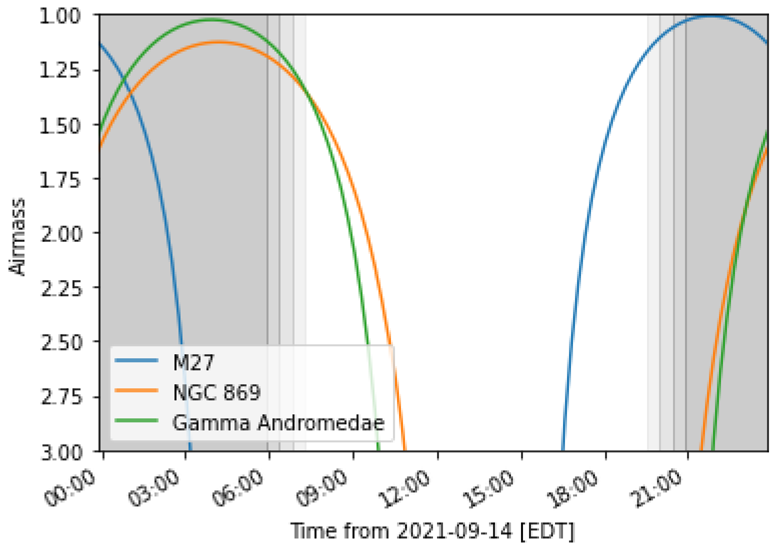
Index

Create an altitude plot (showing the 'up time'), a sky chart, and a finder chart for each of the following sources:

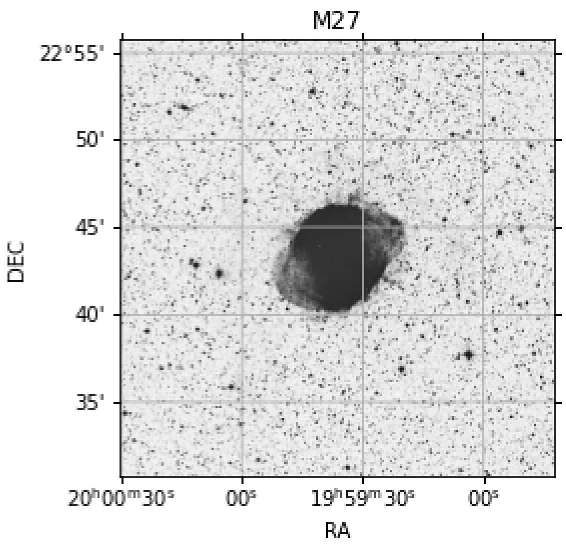
1. M27
2. NGC 869
3. Gamma Andromedae

In [38]:
```
## creating coordinates and targets ##

m27_coord = coordinates.SkyCoord.from_name('M27')
ngc_coord = coordinates.SkyCoord.from_name('NGC 869')
andro_coord = coordinates.SkyCoord.from_name('Gamma Andromedae')

m27_target = FixedTarget(m27_coord, name="M27")
ngc_target = FixedTarget(ngc_coord, name="NGC 869")
andro_target = FixedTarget(andro_coord, name="Gamma Andromedae")
```

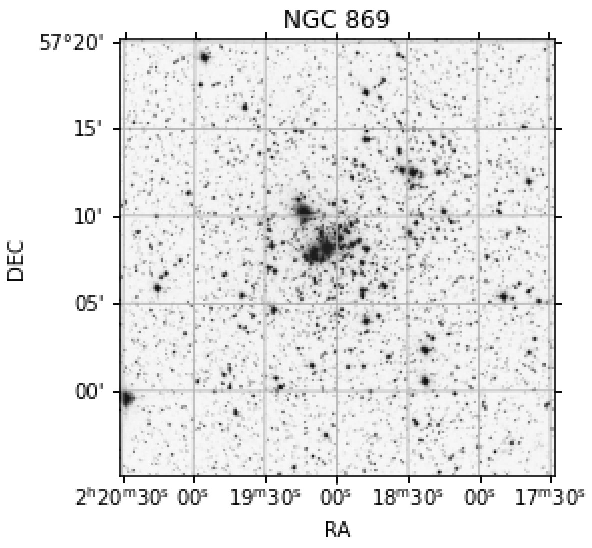
In [39]:
```
# airmass/altitude plot
plot_airmass([m27_target, ngc_target, andro_target],
             observer=CTO,
             time=now.to_datetime(timezone=CTO.timezone),
             use_local_tz=True,
             brightness_shading=True)

plt.legend(loc='best')
plt.show()
```



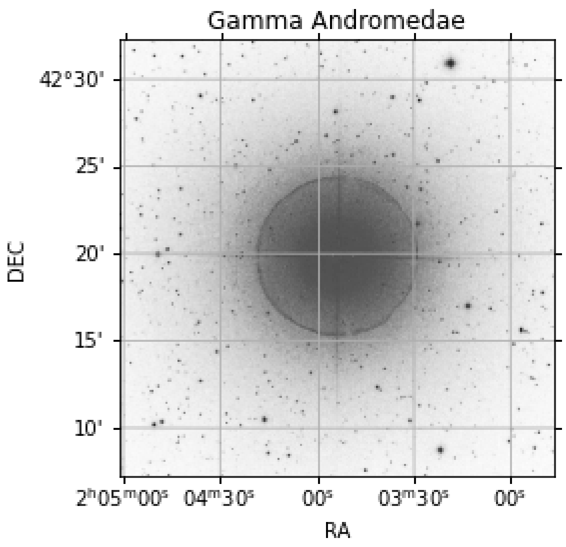
In [40]:
```python
## sky chart M27 ##
ax, hdu = plot_finder_image(m27_target, survey='DSS', fov_radius=25*u.arcmin)
```



In [41]:
```python
## sky chart NGC 869 ##
ax, hdu = plot_finder_image(ngc_target, survey='DSS', fov_radius=25*u.arcmin)
```

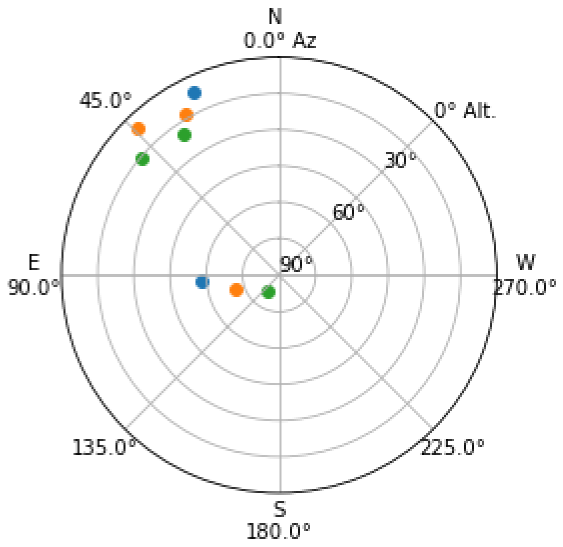


In [42]:
```python
## sky chart Gamma Andromedae ##
ax, hdu = plot_finder_image(andro_target, survey='DSS', fov_radius=25*u.arcmin)
```



In [43]:
```
## Finder chart ##

# where the objects are at sunset
sunset = CTO.sun_set_time(now)

# hour after sunset
onehour_after_sunset = sunset + 1*u.hour
twohours_after_sunset = sunset + 2*u.hour

targets = [m27_target, ngc_target, andro_target]
plot_sky(target=targets, observer=CTO, time=sunset)
plot_sky(target=targets, observer=CTO, time=onehour_after_sunset)
plot_sky(target=targets, observer=CTO, time=twohours_after_sunset)
```

Out[43]:
```
<PolarAxesSubplot:>
```



In [ ]: