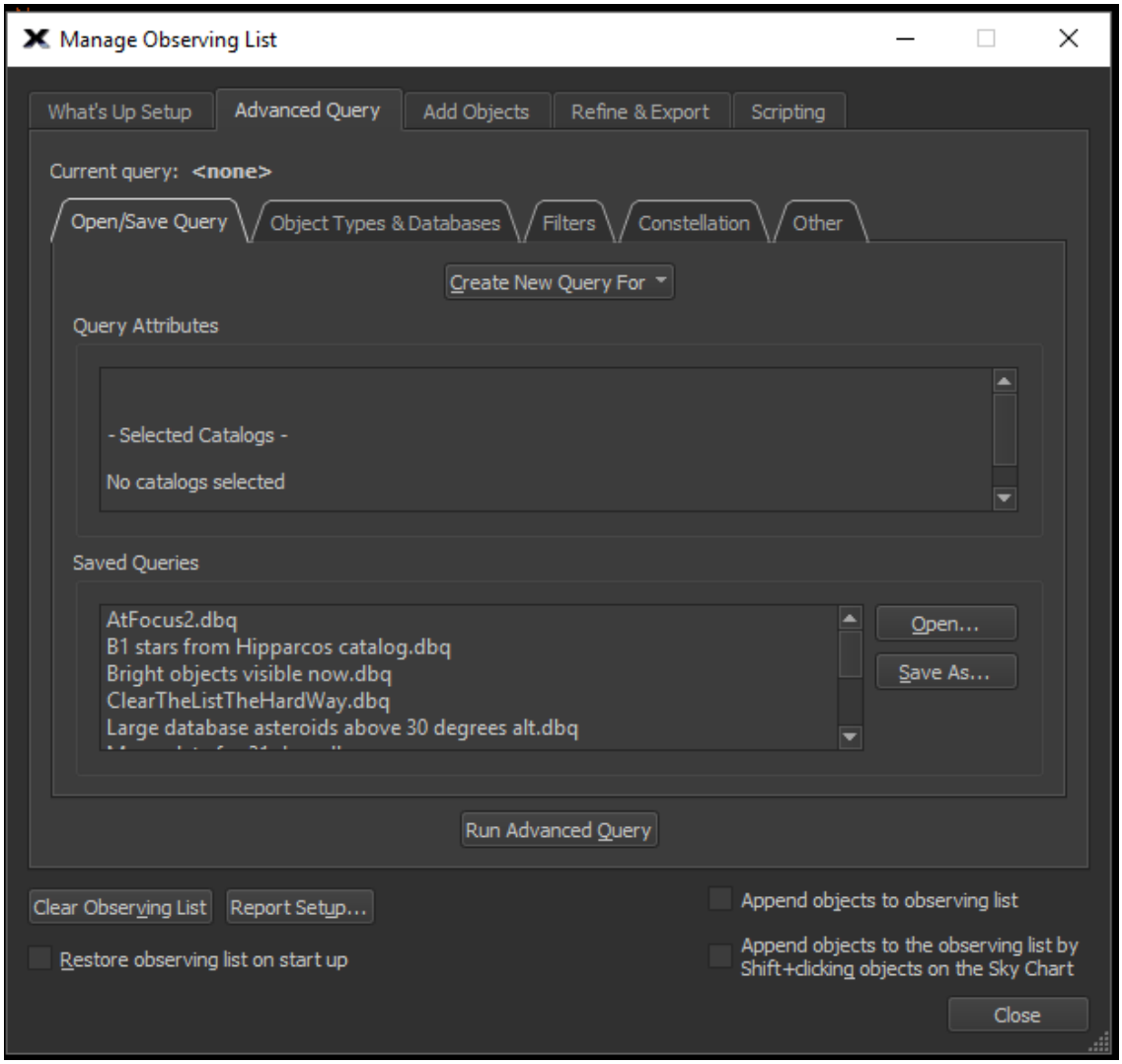
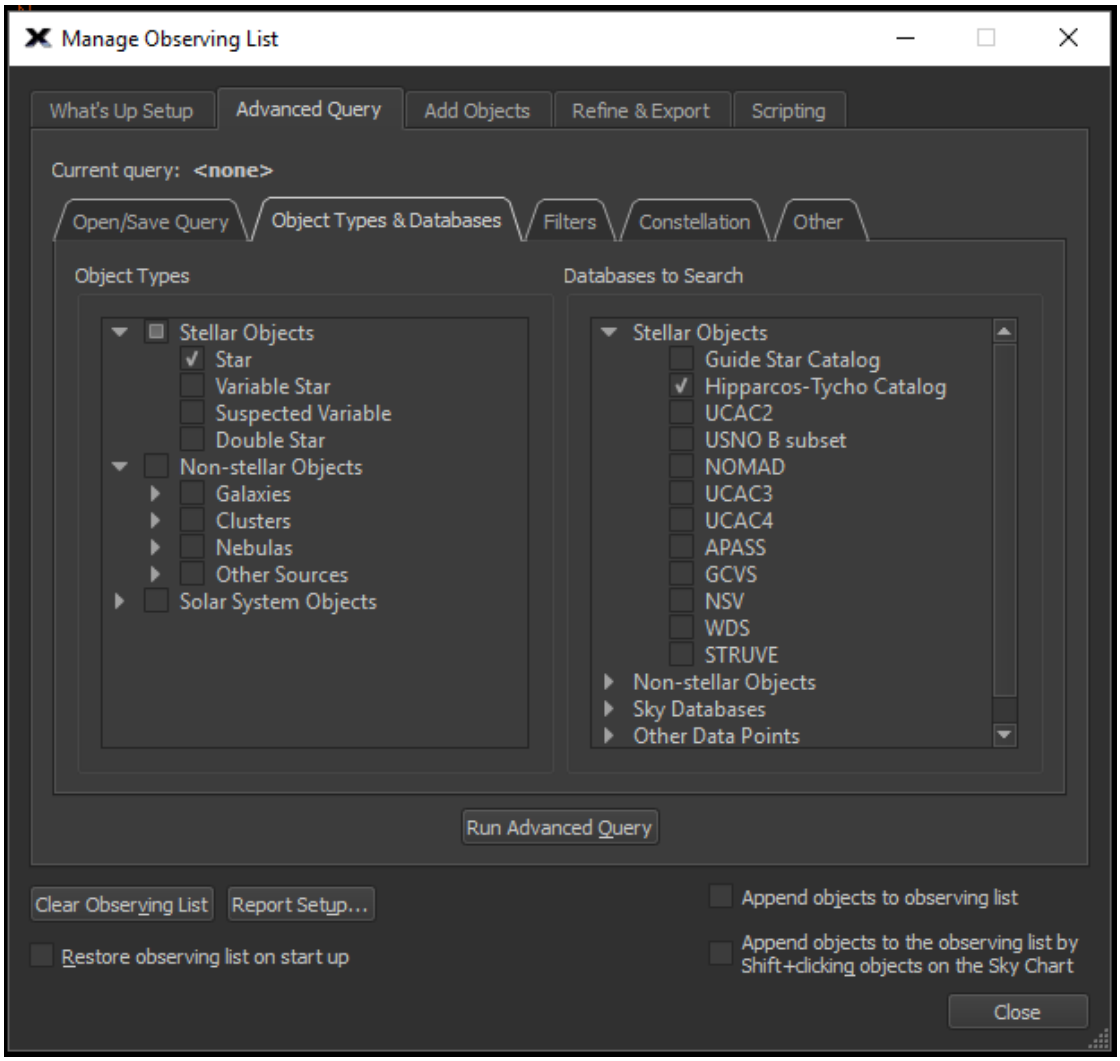
Scripting with observing list queries is kind of a blast once you get the basics down. There are really just three steps. Let's say our goal is to identify all stars between 4.5 and 8.5 magnitude that are in between a minimum and a maximum distance (in arcsec) from a given RA/Dec position. The first step is manual in that you need to construct a database query that meets the needs of this application.

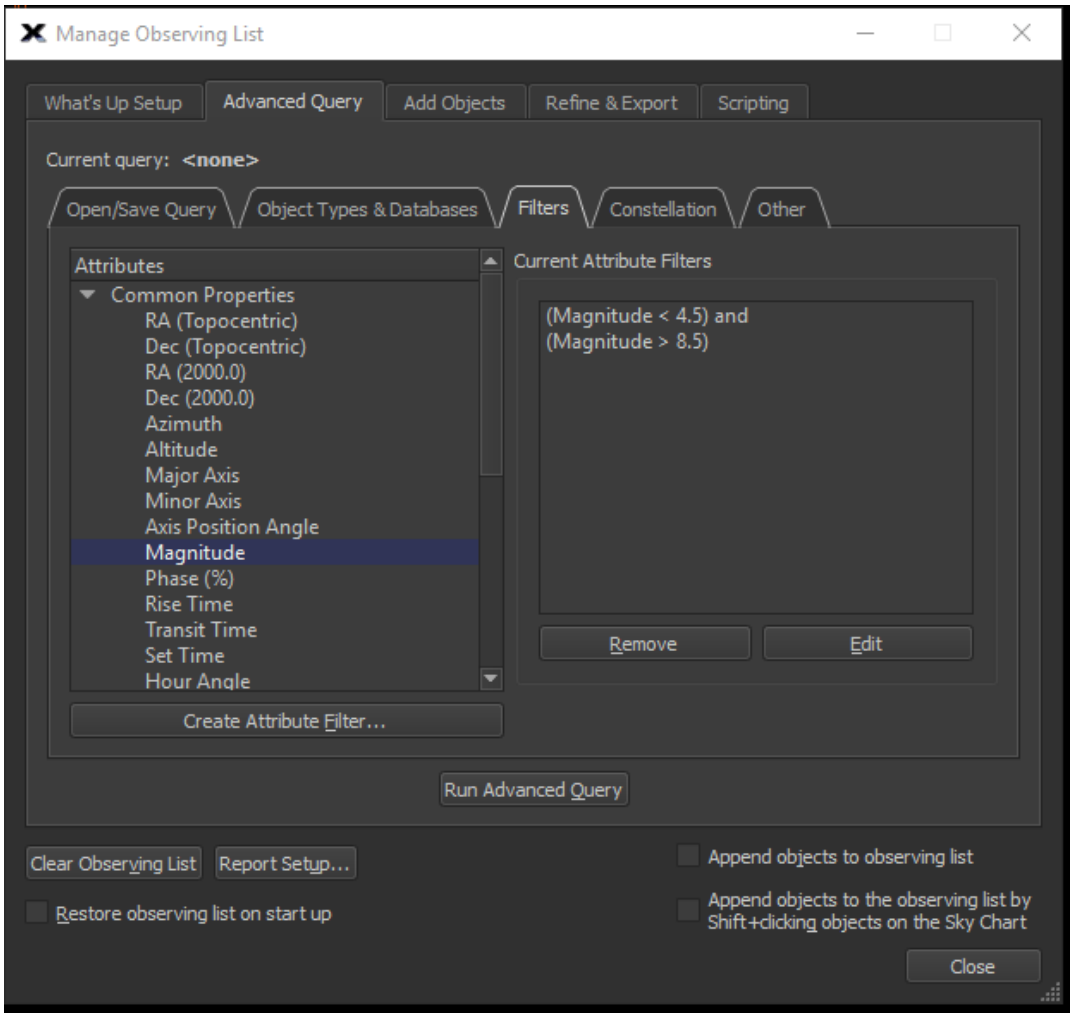
Open the Manage Observing List window. Select the Advanced Query tab. Then, under Current query, select the Open/Save Query tab. Click on Create New Query For and select "Stars" in the drop down menu.



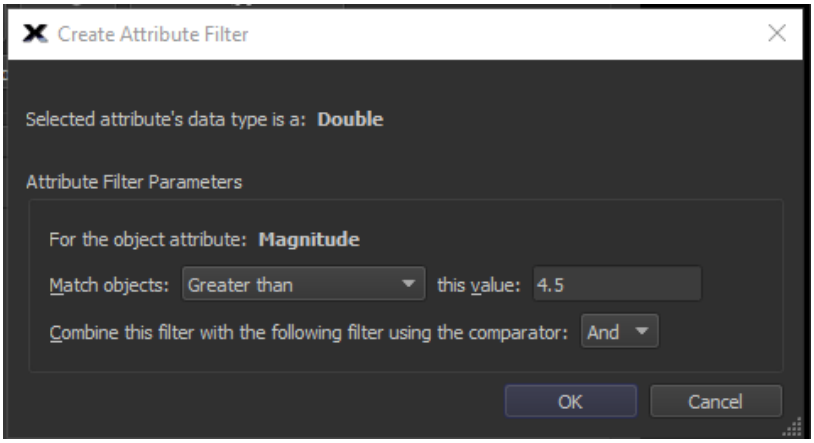
Next, select the Object Types & Databases tab. Under Object Types, pick star and under Databases to Search, pick Hipparcos-Tycho Catalog.



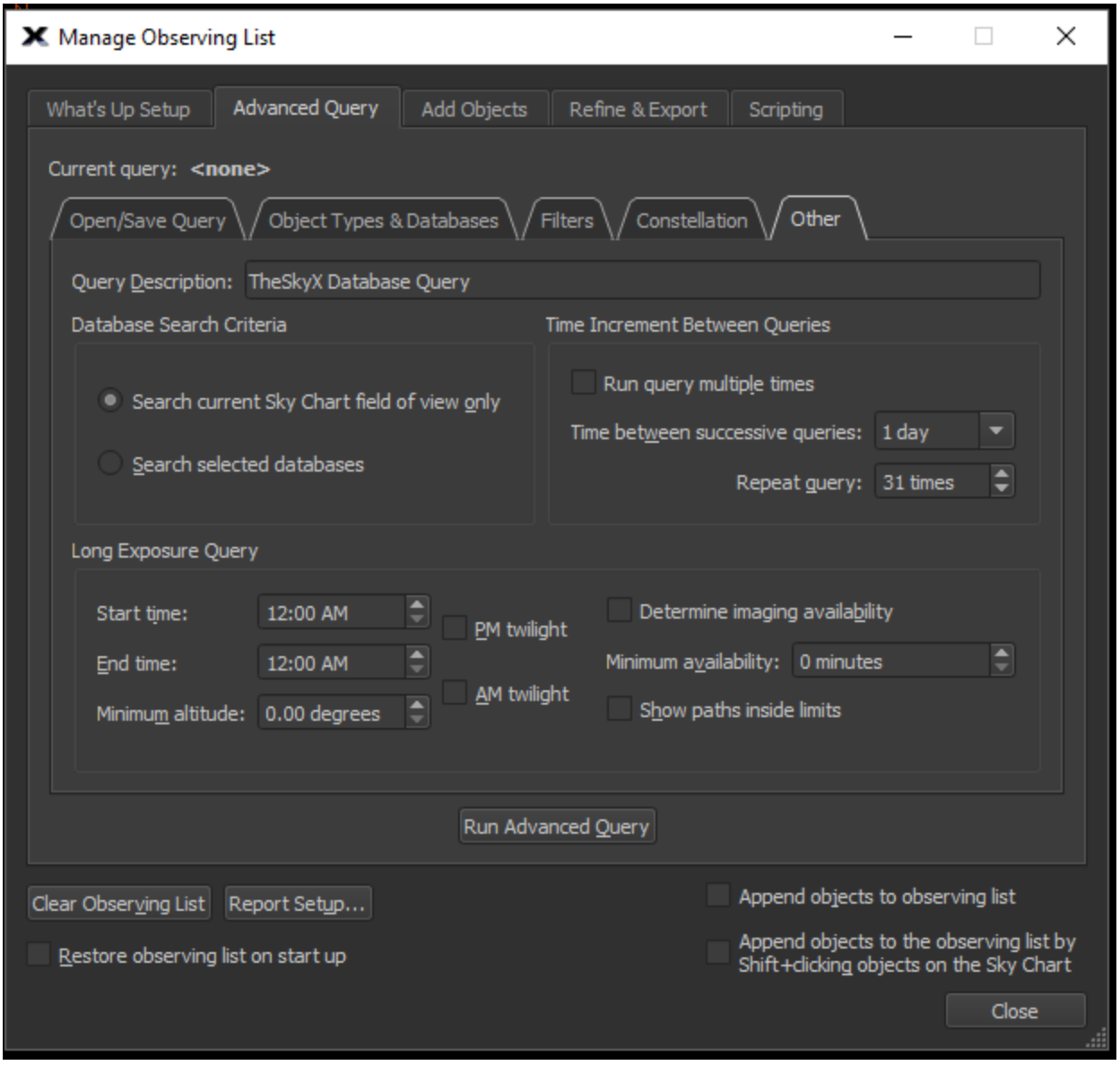
Next, select the Filters tab. This is the place where the catalog is culled for objects that meet the criteria that you're looking for in a star, in this case, a magnitude range. To set criteria, pick an attribute of the star, in this case, Magnitude. Then click on Create Attribute Filter and a new window will pop up.



We'll have to do this twice to set the range. The first time set the Match field to Greater Than and the value to 4.5. Click on the And to set this up for the other end of the range. Then OK. The second time, set the Match to Less than and the value to 8.5. Then just click OK. Your filter will show up in the urrent Attributes Filter box.



Next, select the Other tab. The only criteria we want to set here is Search Current Star Chart Field f View Only. This will allow us to limit the list of stars to a usable range for refining the search via scripts.



Lastly return to the Open/Save query tab and select Save As. You can pick any name for this query, but lets call it LocalStars. Note the location of the query in the Software Bisque Database Queries folder. You'll be needing this information later. In my case, the path would be:

C:\Users\Rick\Documents\Software Bisque\TheSkyX Professional Edition\Database Queries\LocalStars.dbq

And, we're done with the first step. You can run this query for fun to see what it produces, but the next step is to do so with a script.

Scripting

There are three classes that we'll use for running the query and processing the results: Sky6DataWizard, Sky6ObjectInformation, and Sky6StarChart.

The first part of the script locates and runs the script:

```
Sky6DataWizard.Path = "Your full path goes here"
Sky6DataWizard.RunQuery()
```

When the script is run, the results are made available in the sky6ObjectInformation object array, one entry for each object found. As we process the array, both the property sky6ObjectInformation.Count and sky6ObjectInformation.Index are used. The first property lets us know the number of objects (stars) that the query found. The second property is set such that we can index into the array. That is, if we set the Index to 5, we will be accessing the properties of the 6th (remember zero-based) object (star) in the array of objects.

With these properties we can build a loop to go through all the objects that the query produced:

```
For i = 0 to sk6objectInformation.Count
        Sky6ObjectInformation.Index = i
        //the rest of the processing, see below//
Next
```

Now let's explore how we access the properties of our stars. The ObjectInformation class uses a two-step approach to access the data of any object. First a Property is declared, then the value associated with that property is extracted:

```
Sky6ObjectInformation.Propery(sky6objectinformationProperty.ObjInfoProp_Name1)
Starname  = sky6objectinformation.ObjInfoPropOut
```

What we're interested in here is RA and Dec, because we'll use those to locate stars on the chart.

```
Sky6ObjectInformation.Propery(sky6objectinformationProperty.ObjInfoProp_RA2000)
StarRA  = sky6objectinformation.ObjInfoPropOut
Sky6ObjectInformation.Propery(sky6objectinformationProperty.ObjInfoProp_DEC2000)
StarDec  = sky6objectinformation.ObjInfoPropOut
```

So this will give us the means to get the data on each star. Now to the sky6StarChart.

Sky6StarChart methods allow us to get and set characteristics of the star chart is the TSX window. There is a lot of them. However, for this exercise we only need to know location of

the center of star chart, and how it's X/Y coordinates map into RA/Dec coordinates. Our script can use that information to determine what stars are at the minimum and maximum distance from the center.

First, however, we'll have to backtrack a bit. Remember that we designed the query to choose only stars which would show on the star chart. So, if we made the star chart size the same as the maximum distance from the center, then we could reduce the number of stars to cull through significantly. To do so, before the script command RunQuery, we need to set the Width of the star chart to twice our maximum distance (in arcsec). The width of the current display is set (in degrees) by

```
Sky6starchart.FieldOfView = MaxArcSec/1000
```

Back to here again, we can now get the scale in pixels/arcsec by getting the width of the chart in pixels:

```
Pixperarcsec = Sky6starchart.WidthInPixels/(sky6starchart.FieldOfView*1000)
```

The last little piece we need is to convert the location (in RA/Dec) of a star to a location (in pixels) on the chart. The method;

```
Sky6StarChart.EquatorialToXY(StarRA,StarDec)
StarX = sky6StarChart.dout0
starY = sky6StarChart.dout1
```

can give us the location of each star from the object list. The location of the center of the star chart is:

```
centerX = sky6StarChart.Width/2
centerY = sky6StarChart.Height/2
```

The distance in X and Y of the star from the center is:

```
distanceX = StarX-CenterX
distance = StarY-CenterY
```

The absolute distance of any star from the center is just root of sum of squares.

```
distancePixels = SquareRoot((StarX^2)+(starY^2))   //in pixels
```

and

```
distanceArcSec = distancePixels/pixelsperarcsec
```

if the distance is less than the maxdistance and greater than the minimum distance, then the star is in the right place.

That’s all the pieces. I don’t want to spoil all the fun, so try putting them together into the script that does the job.