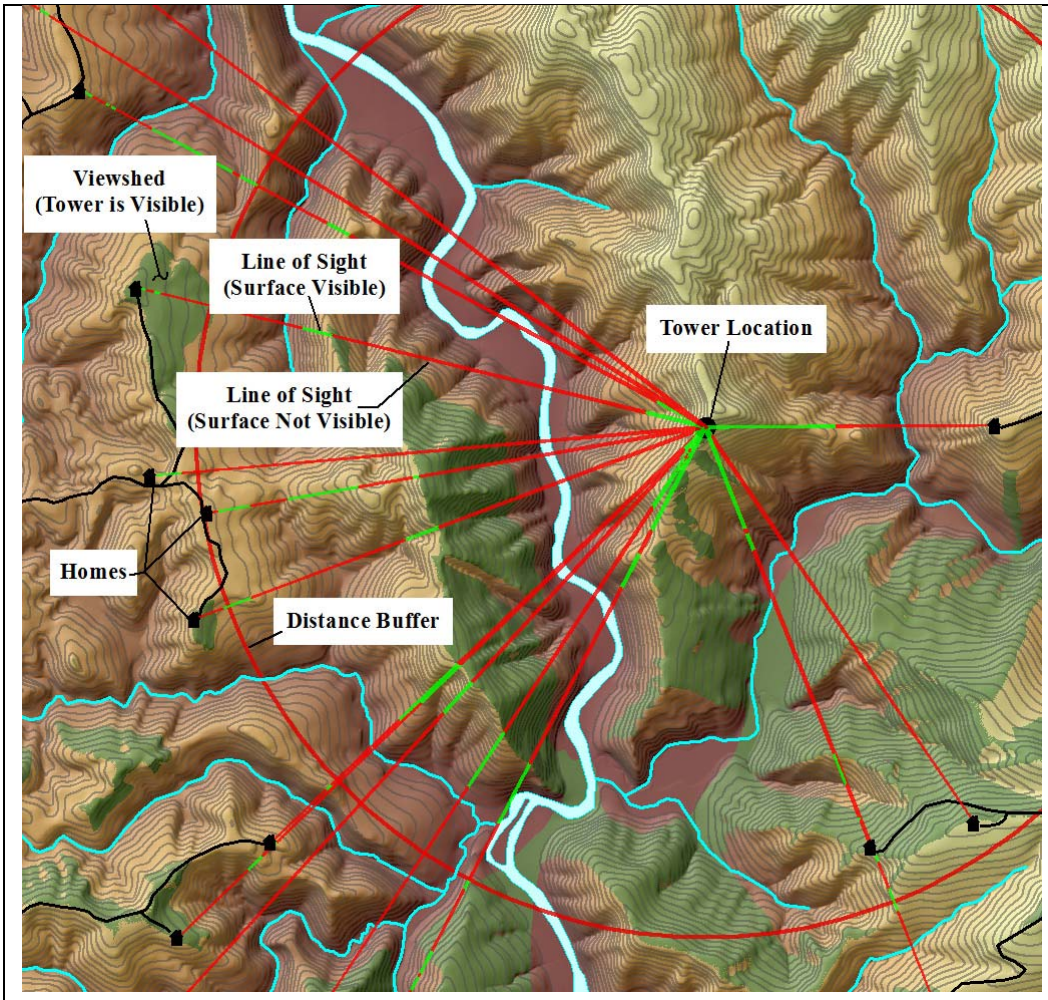
A telecommunications company often needs to build telecommunications towers near scenic, low density residential areas and needs to find suitable sites for its towers while minimizing costly land surveying expenditures and opposition from local residents. The sites need to be located at a high enough elevation to allow for good signal transmission but in such a way that they will not be visual eyesores to current residents in the area.

A telecommunications company has hired a GIS consultant to develop web applications that will assist in determining possible locations for proposed telecommunications towers. The GIS consultant is tasked with creating a web application that will allow the end-userto choose a point/location for the tower and input a visibility buffer distance. The web application will then return a viewshedand a distance buffer, indicating the locationswithin the study area, from which,the tower will be visible,and a circle with a user defined radius will be created, depicting the designated distance from the chosen tower location.

The GIS consultant is also tasked with producing a line of sight web application that will help determine which surface areas between the proposed tower location and the individual homes in the area are visible. Together, these applications will be used by the telecommunications company to determine the best preliminary locations and sizes of their telecommunications towers while reducing the amount of ground survey work that would normally be required.



A tower location is chosen by "clicking" a location on the map with the mouse.

The application creates a point on the map and then uses the DTM to calculate a viewshed that shades all areas on the map from which the tower can be seen (depicted with dark green shading in this example).

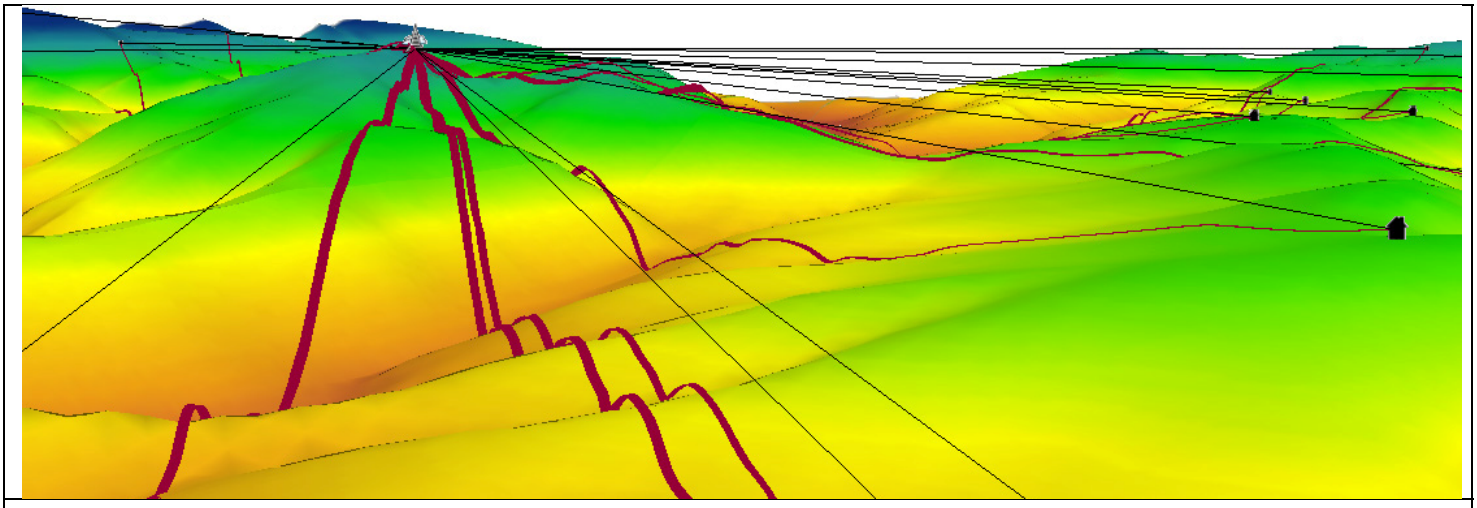
A buffer is created around the selected tower location (depicted as a large red circle in this example). The buffer radius is a parameter that is chosen by the end user.

Lines of sight that go from each home to the chosen location are created. The portions of those lines in this image that are red, are areas of the DTM surface that aren't visible from their respective home and the green portions of the lines are areas of the DTM surface that can be seen from their respective homes.

The distance buffer helps the telecommunications company determine whether the proposed tower will appear large enough to be a nuisance to the residents whose homes will have an unobstructed view of the tower. The larger and taller a tower is, the further it will need to be located from a home.

Theviewshedis a collection of all of the elevation grid cells that are visible from an observation point; in this case, the observation point is the elevation grid cell that is selected as the location of the proposed tower. In the above image, someone would be able to see the proposed tower if they were standing on any of the green viewshed patches, unless there were other obstructions on or above the surface, such as trees or shrubs that were blocking the view. This is where the line of sight application comes in.

The line of sight application is similar to the viewshed application in the way is uses the elevation values of the elevation grid cells to determine visibility but as the name suggests, it calculates a view between two points, an observation point and a target point. These were given additional elevation value parameters to replicate viewing from a window in a two story house (observation point) and the top of a 100 foot tall tower (target point). The home/towerheight parameters can be adjusted to reflect the number of stories a house might have and/or to allow for a taller or shorter tower.

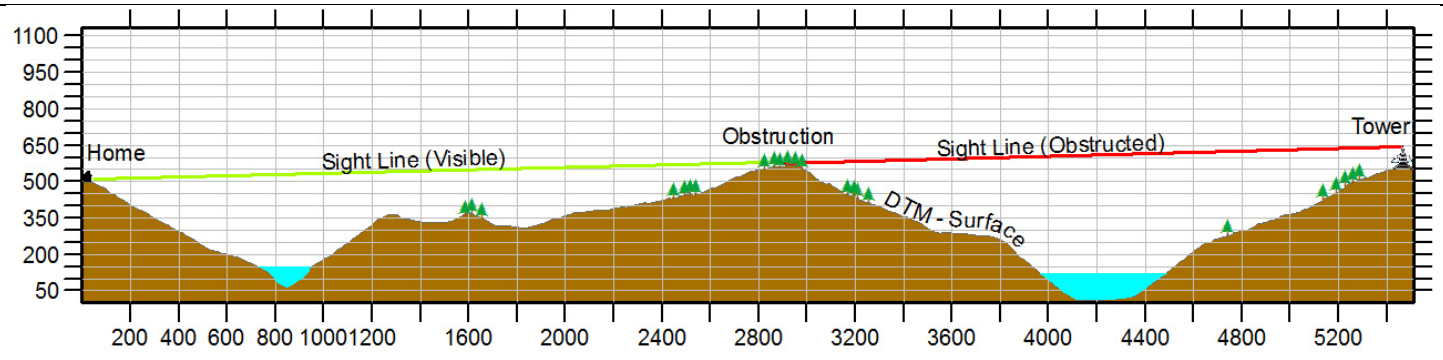


In the aboveArcScene image, the black lines are “sight lines” and the red lines are “lines of sight”. As you can see, the red “lines of sight” have many vertices and follow the contours of the surface while the black “sight lines” are straight lines between two points. The perspective view of this Arc Scene image belies the fact that each “sight line” is paired up with one “line of sight” and that both have identical x,y coordinate geometry values but different z values. This could facilitate further calculations to determine where these z values might be close enough together to allow for surface objects to obstruct the view. If a tree outcropping exists or can be planted in locations where the z values are near merging, a view of the proposed tower may be obstructed.

If too many homes fall within the visibility buffer radius and are located within the visible viewshed area, the end user can simply choose another point/location for the tower or try a smaller tower until an acceptable balance between the distance from the tower, the height of the tower and the number of homes that can see the tower is achieved. At that point, it needs to be determined whether there are any existing surface features that will obstruct the chosen location for the tower or whether it is possible to mitigate by planting some surface features that will obstruct the view of the proposed tower.

The GIS consultant’s application will use publicly available, downloadable surface data such as Digital Terrain Models (DTM) or Digital Elevation Models (DEM)in raster elevation grid form. These DTMs don’t always include surface features that might serve as visibility obstructions, so pre-determining the locations where the DTM surface nearly merges with the line of sight will save the telecommunications company money because it

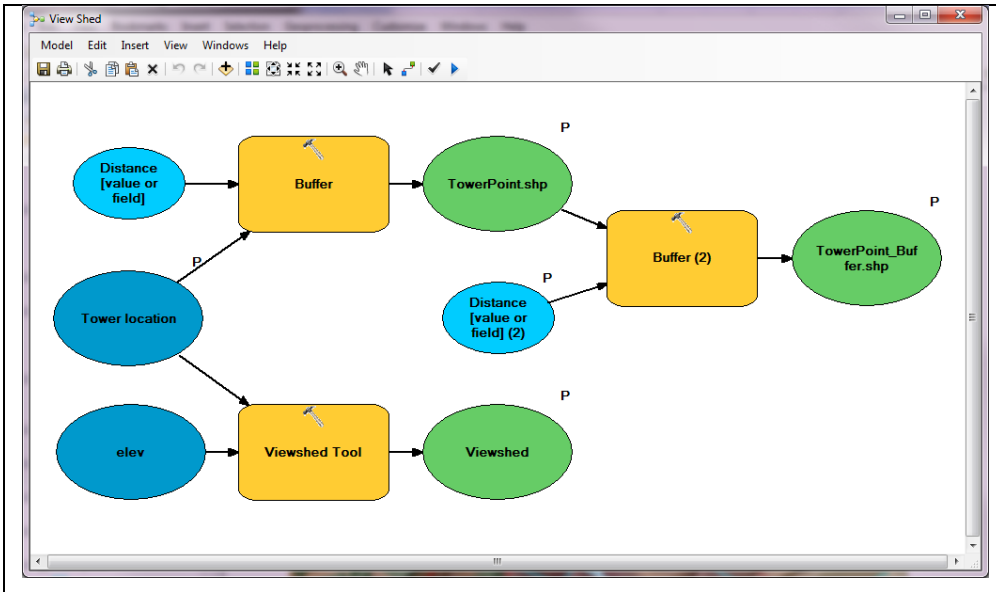
can send a field survey team out to these specific coordinates to gather data on existing surface conditions rather than paying for massive surveys of the entire area.



This scaled cross section, courtesy of the [A-Prime Software "CrossView" plugin], shows an example of a home that has a view of the proposed tower and falls within the 2 mile visibility buffer. In this project scenario, a survey may reveal that an existing grove of trees serves as a visibility obstruction.If no trees exist where the surface and line of sight come close to merging, it may be possible for the telecommunications company to plant a grove of trees in this area to create a visibility obstruction.

I began with some data that was obtained from another GIS class I had recently completed. This data included a raster DTM as well as some roads, rivers, houses and tower point shape files. Creating my base map was fairly simple and straightforward. I created a hillshade raster and some contours to give the base map more of a 3D look and to facilitate depth and slope perception. Unfortunately, I spent many hours trying to get my raster based applications to work with a web client before discovering why things weren't working.

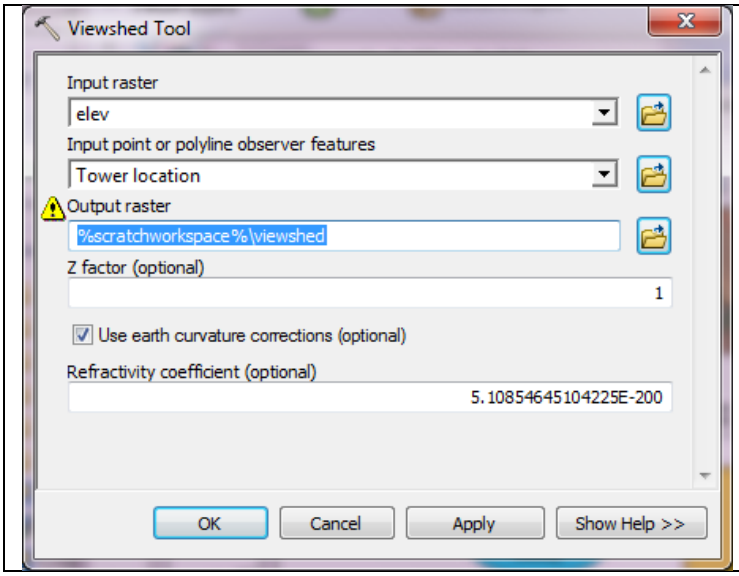
Understanding which input and output data types will work with which end user software client is extreemly important. In my case, the raster DTM, hillshade and contoursdisplayed just fine in the ArcGIS Desktop Client, the ArcGIS Explorer client and the Web application client but my geoprocessing applications would only work in the ArcGIS Desktop client because the raster data type is not supported as an input parameter in the other two clients.



The viewshed and buffer model uses a "mouse click" as input for the proposed tower location and builds a viewshed with shapefile, rather than raster, as an output type.

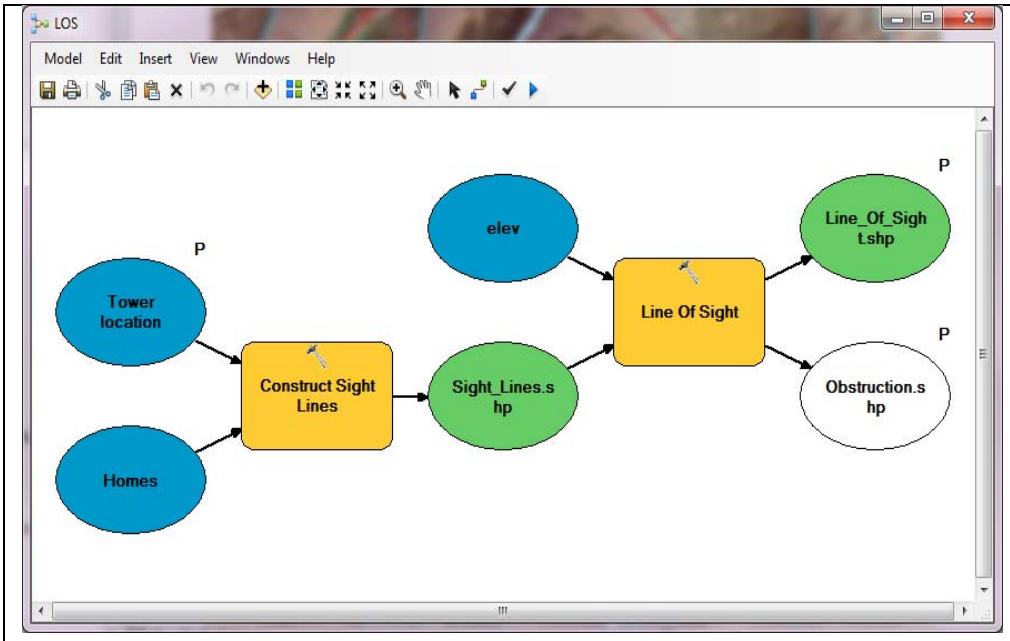
It also creates two buffer shapefiles, a small one for the tower location point and one for the visibility distance parameter that is input before running the application.

The Tower location input had to be set up as a feature set rather than a feature class due to supported input data types.



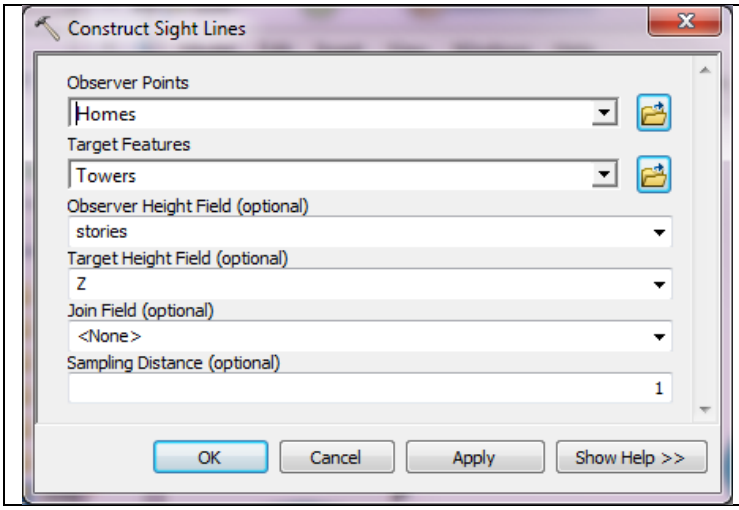
The viewshed parameters require an input elevation raster, an observation point and an output location for the viewshed raster that would be created.

It was important to make sure my geoprocessing/environments/workspace file path locations were properly setup and that the output location for my file was set to "%scratchworkspace%\XXXXX", where X equals the name of the file being created.



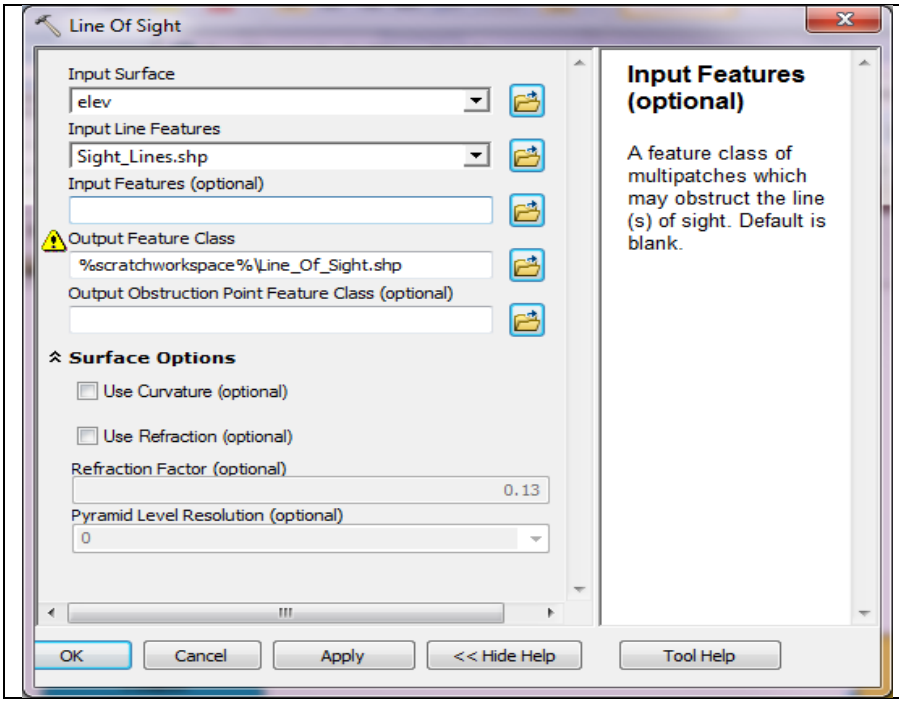
My Line of Sight model was initially setup to use the tower point shape file that was created in my viewshed model but I kept getting file sharing errors until I revised it to use an input variable just like the other model used. The user simply clicks the mouse on top of the point created with the viewshed application.

This was similar to what happens when you remove a layer from ArcMap and then try to delete it with ArcCatalog or through windows. It tells you that the file can't be deleted because it is in use by another program.



As you can see from the Line of Sight model above, the model first creates 3D sight lines between the observer points and the target features. These are the straight black lines shown in the ArcScene image above.

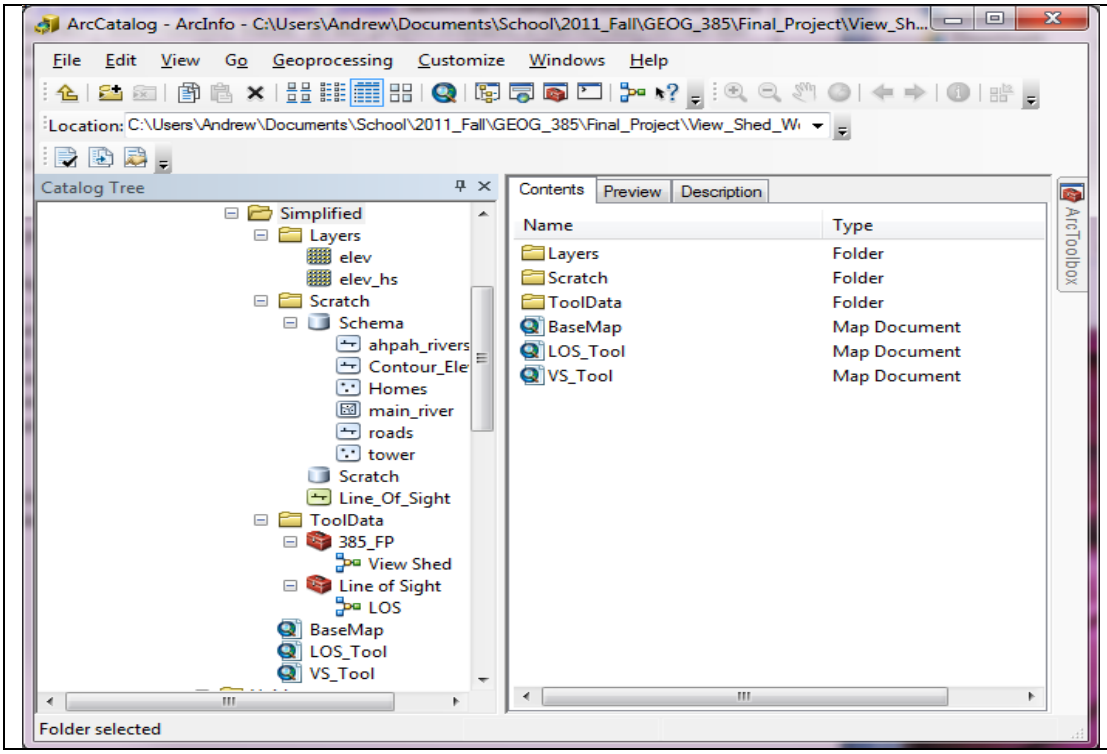
This is where the observer height fields and target height fields are set. The homes point file has a "stories" attribute field that reflects the number of stories a home has, or the height from which someone looking out a window on the first or second floors might be.



The Line of Sight uses the surface raster and the sight lines created above to create the surface hugging red lines shown in the ArcScene image above.

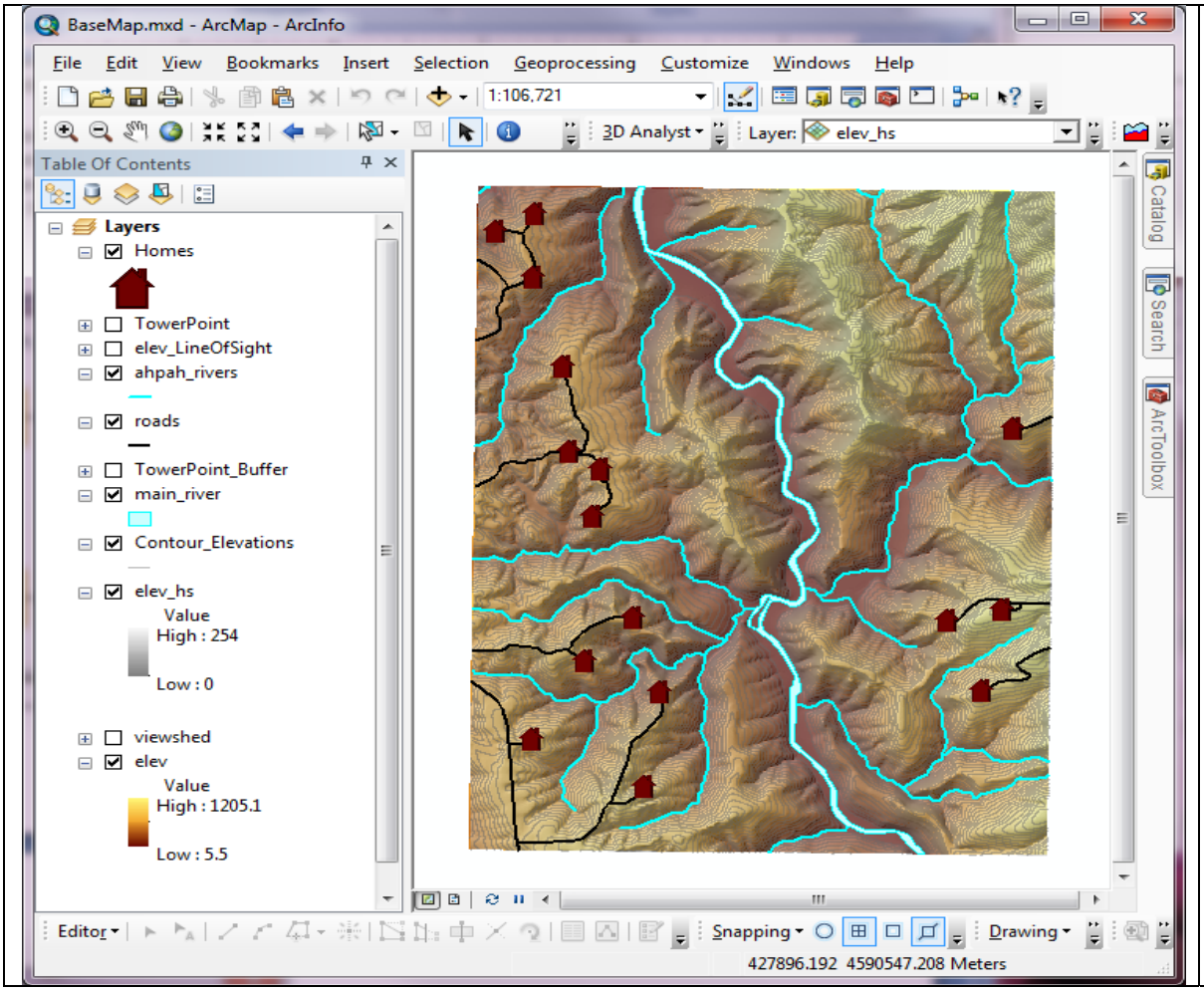
These lines are red where portions of the surface can't be seen from the starting endpoint (homes) of the lines and green where the surface is visible. If the ends of the lines are green, the tower is visible.

This is also where I had really hoped to incorporate the input feature class option to simulate surface feature data like a grove of trees that had been obtained via a ground survey or similar means.



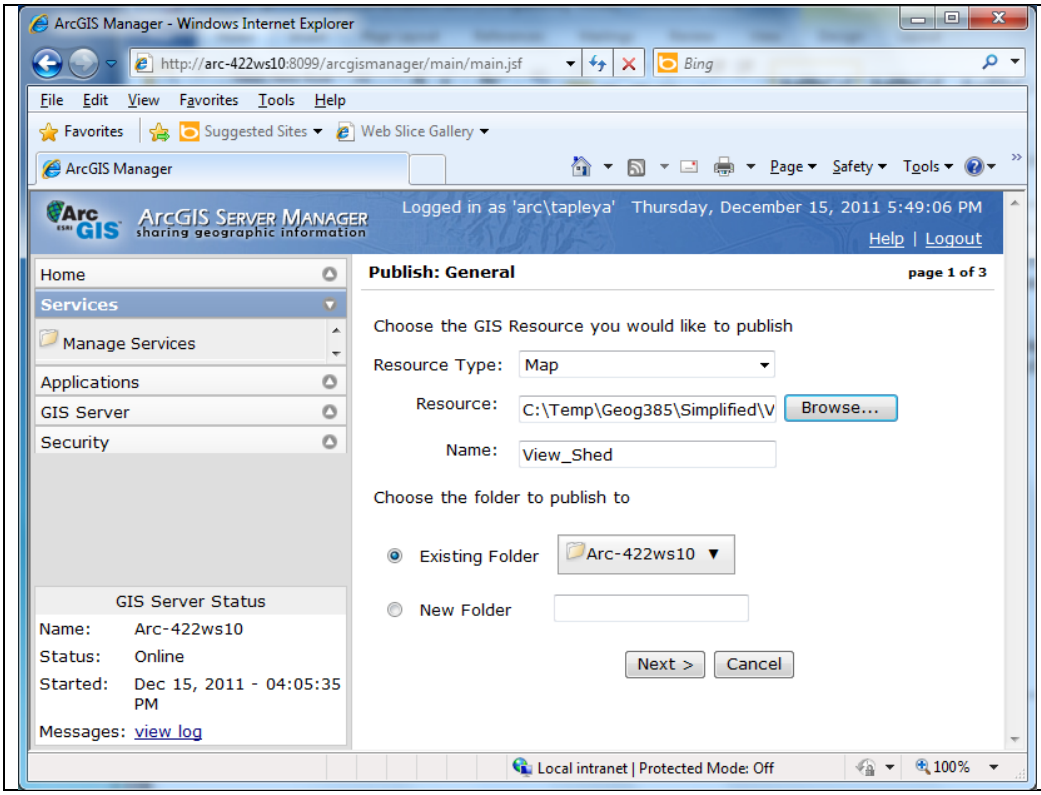
I created new file databases andfeature classes for my tools, drawings and layer files, a scratch folder with a schema file database and a scratch file database. I also created a folder for my rasters.

I created three drawings. One was just the base map while the other two were what I attached the tools to.

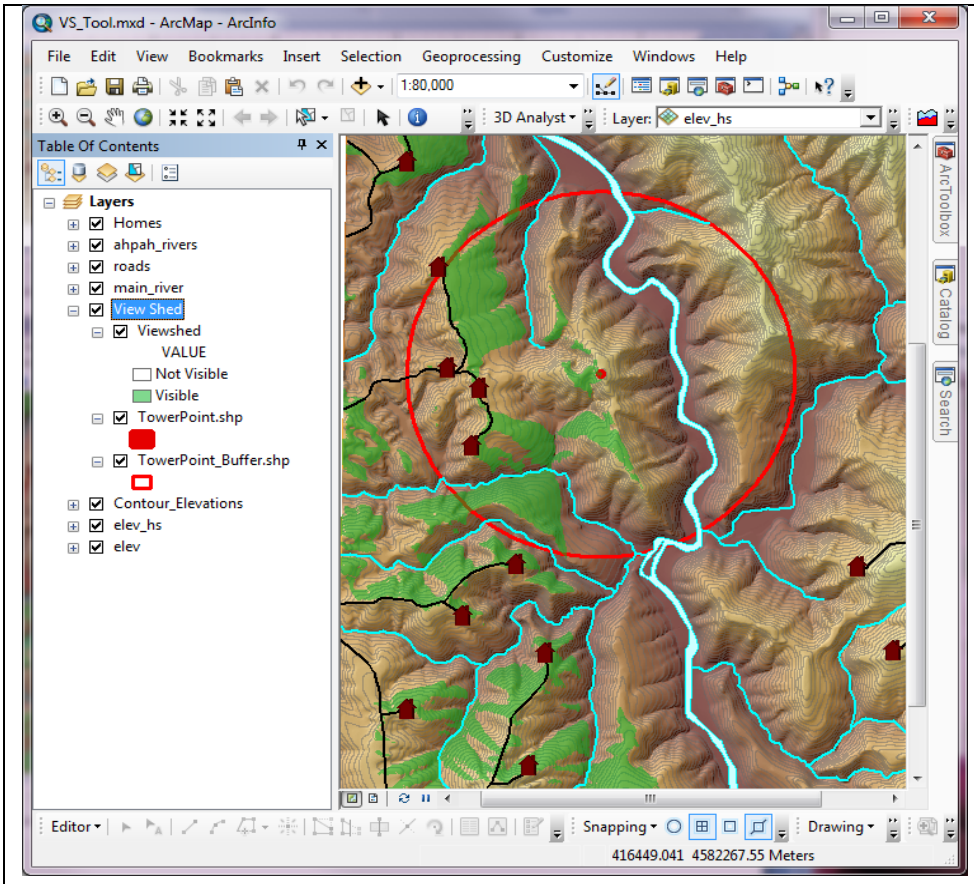


I created a separate base map drawing that only contained the only base information.

It had contours, a hillshade raster, the rivers and roads as well the local residences.



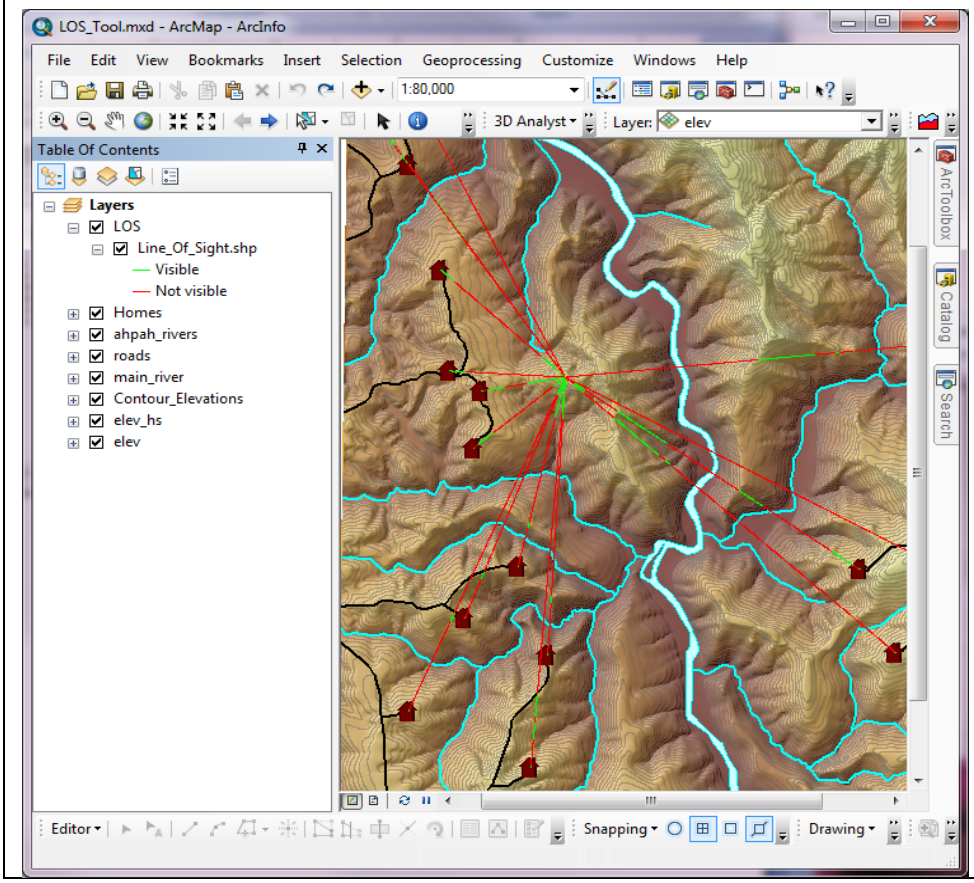
I uploaded the base map from the above image as a map service.



I took my viewshed model and dragged it into the layers menu to embed it into the drawing. I set the layer symbology properties and even exported layer files for each layer that the tool creates.

For some reason my viewshed layer would adopt these symbology settings in the published version but my buffers and tower point wouldn't.

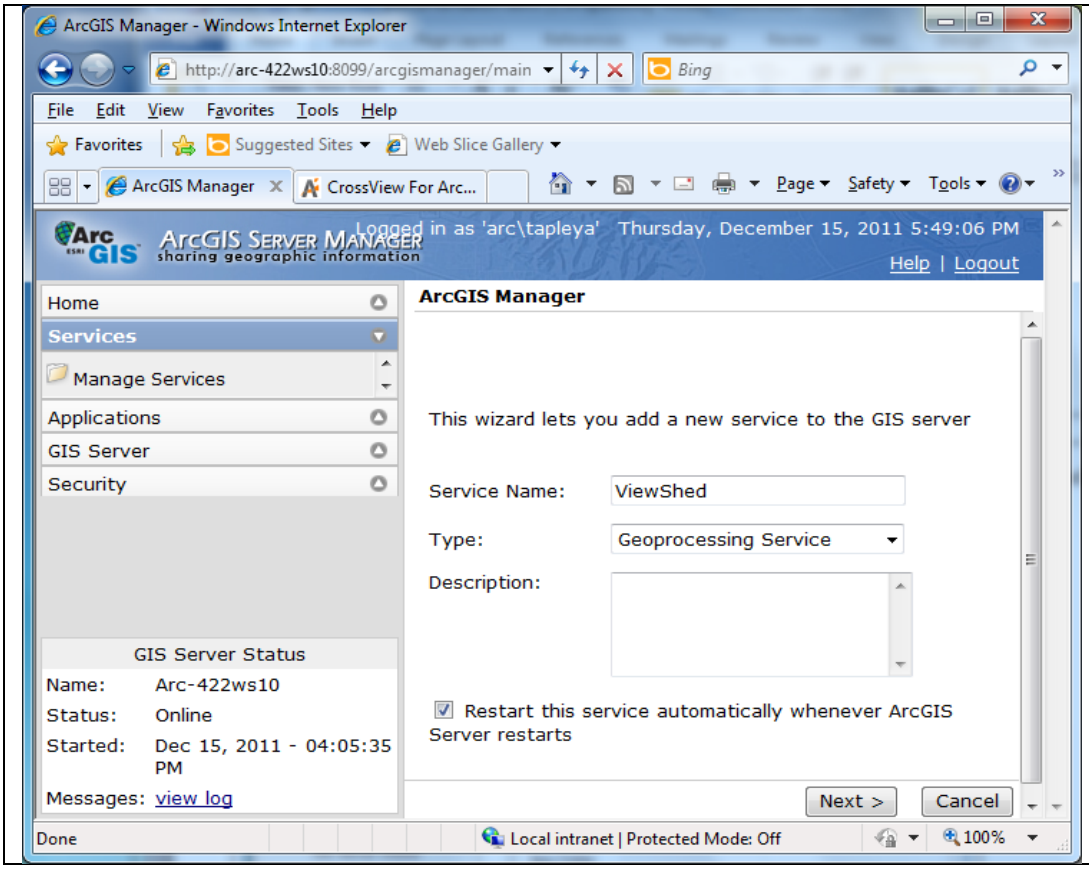
Each time I ran the application it would create new layers on top of the old layers but the buffers would always come in with random fill and line settings.



The image to the left shows my line of sight tool embedded into a drawing file.

I expected the symbology attributes to be missing for these 3D lines because potions of the lines are shaded green and other portions of the line are red.

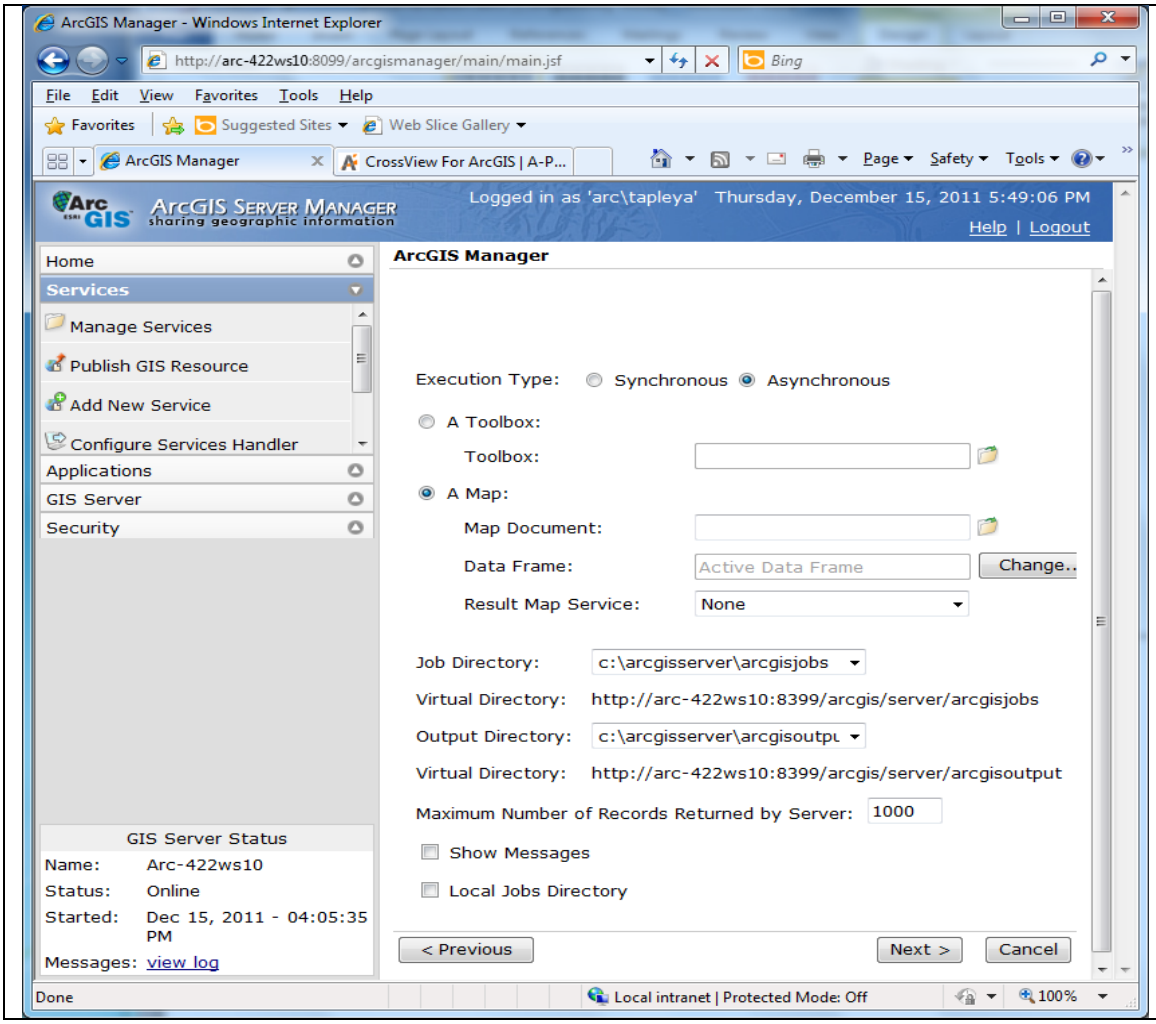
Each time I ran the tool it created a new layer for them but it kept the same symbology.



After publishing my basemap, I published my other two drawings, the ones with the embedded tools, as geoprocessing services.

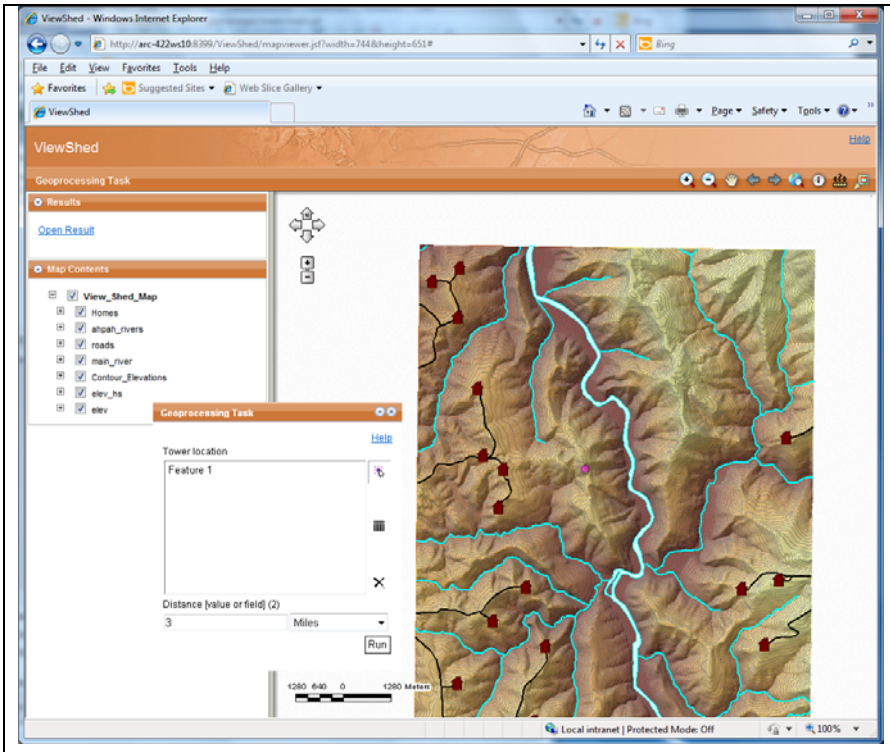
When I published everything as applications, I used my basemap as the initial map file and attached the two different tools I uploaded as the geoprocessing applications.

I took screen shots of that process but somehow I lost all of them and can't include them in this report because they can only be taken again from the computer lab.



When publishing my tools, I chose to "A Map" as my source and pointed it to the two drawings with the embedded models in them.

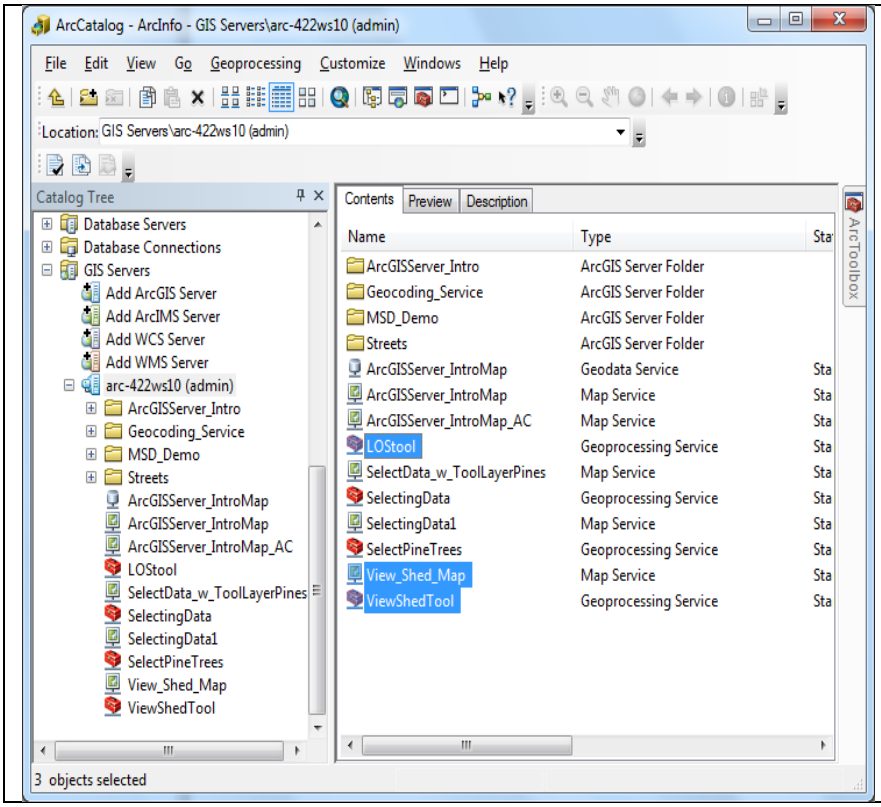
I couldn't get the application to work until I made this change. It just wouldn't work trying to upload them as toolboxes.



My basemap and geoprocessing applications would load and display in a web application but would error out every time I ran them.

This was the most frustrating part because I spent at least 20 hours in the lab trying to figure out why it wasn't working. I had followed all of the procedures they used in the watershed tutorial, and a viewshed is the same thing as a watershed from a computer input/output/data standpoint so I was convinced that it should be working.

Don't do as I did and assume that you don't need to dig deep into the help files. They are setup in a way that you would think geoprocessing tasks can be published for use by web applications but when you dig through the files, you'll find that rasters aren't supported as data input or output in anything but ArcGIS Desktop applications.

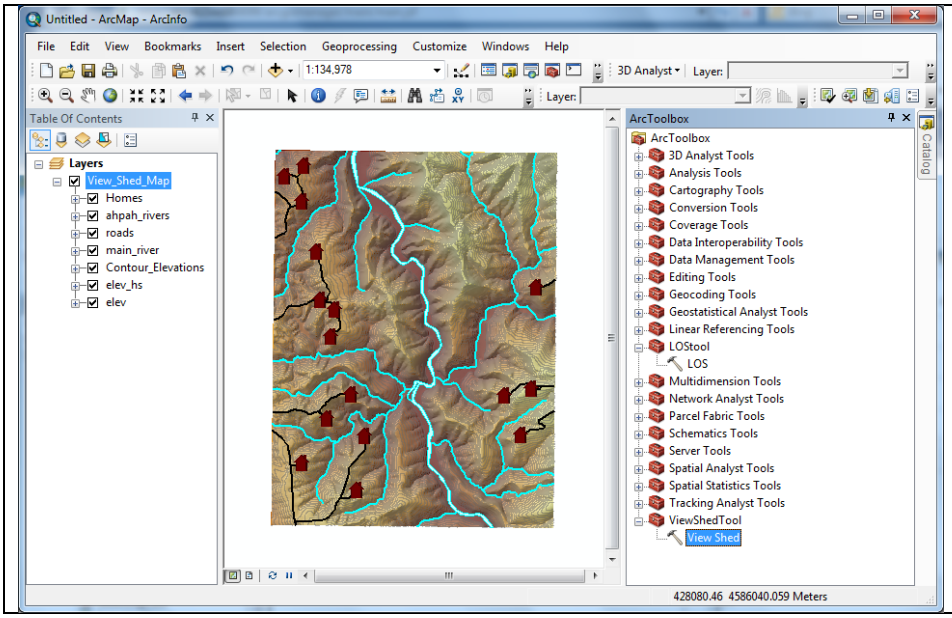


This is an ArcCatalog view of the basemap and applications I published to the server. They selected and highlighted in blue in the image to the left.

As you can see, the basemap "View_Shed_Map" has a map icon and the other two drawings with embedded tools that I uploaded as geoprocessing applications have toolbox icons.

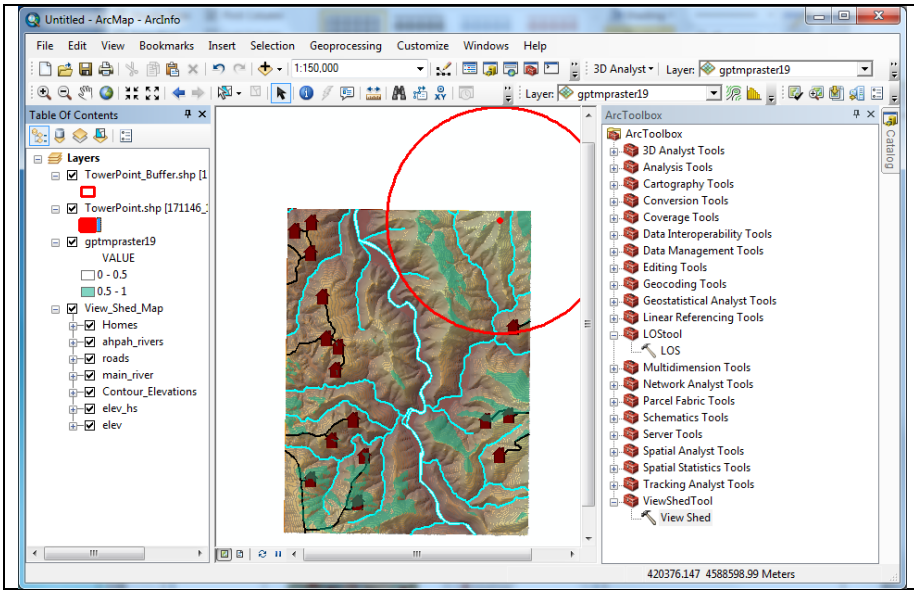
The end user opens a blank ArcMap session, connects to/logs into the server and drags the map basemap into the workspace area.

The end user then opens up the ArcToolbox menu and drags each application from the server into the toolbox area where it is inserted as a new tool.

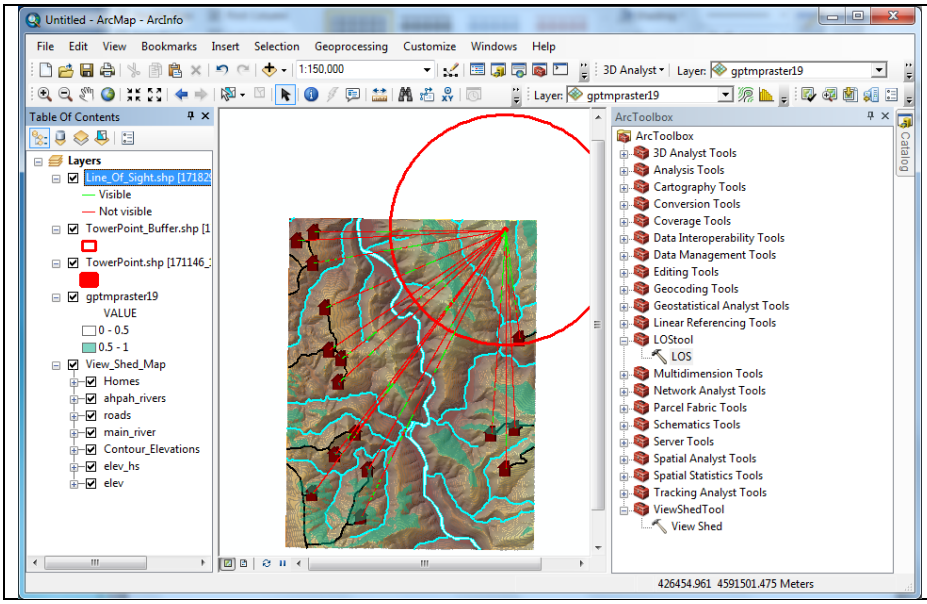


This image shows the basemap and geoprocessing applications loaded from the server into a new session of ArcMap.

Note that the icon in each toolbox now looks like a tool rather than a model.



This is the result after running the View Shed application directly from the server.



This is the result after running the LOS application directly from the server.

The biggest and most obvious issue with my project was the fact that raster datasets are not supported as input or output data types, in the geo-processing environment, for anything other than an ArcGIS end user platform. This isn't easily discernable if you rely upon the ESRI tutorials as they don't point this out in the tutorial instructions. I wasn't able to put the time I wanted into all of the geoprocessing applications or this document I had planned because I didn't dig into the help files soon enough.

One of the biggest challenges is the limited access to the server environment. Creating a model in ArcMap is the easy part but getting everything to work on the servers can be time consuming and frustrating. Even when I got the applications to work there were little things like some layers being properly symbolized and others not that I really didn't have time to figure out. If I could mess with it and try it at home, I'd come up with a solution but I must be in the lab, at my specific computer, to work on these issues and that window isn't open very often.

If I had more time and access to the lab so I could continue working on this project, I would try to implement the multipatch option to create visibility obstructions in areas where the distance between the sightlines and lines of sight come within 30 feet or less of one another but don't quite merge.

This class has taught me that the "out of the box" geoprocessing web applications are still pretty raw/buggy and that web applications require a lot of attention to all the little details if they are going to work as they should. I really wish I could work on and experiment with this more but it just isn't so.