

# OpenWind Tutorial

(A fictional tale of wind power development)

## Background

The OpenWind tutorial is set in northern Vermont on the shores of Lake Champlain in a not particularly windy spot. However, the OpenWind team are shrewd investors and are privy to secret information which tells them that circumstances are likely to change such that a 6.5m/s @ 80m site will be pure gold by the time their prospecting efforts are complete.

With this in mind, they have installed two meteorological masts near the Eastern shore of the lake on some small rolling hills.

OpenWind have taken the genuinely smart move of commissioning a SiteWind study by AWSTruepower (<http://www.awstruepower.com>) which uses the one year of data at both their towers to adjust a wind map created by AWSTruewind's state of the art meso-modeling system.

OpenWind are currently in talks with AAER (<http://www.aaer.ca>) with a view to purchasing some of their splendid 1.5MW turbines.

## Getting Started with the Interface

With the OpenWind software open, click your mouse on the **File->Open Vector Layer** menu item and browse to the directory containing the tutorial data. Now browse to **../tutorial\_data/GIS/shape\_vectors/** and select all the shape files in that directory. Then press the **Open** button. Moments later you should see something resembling the screen shot below.

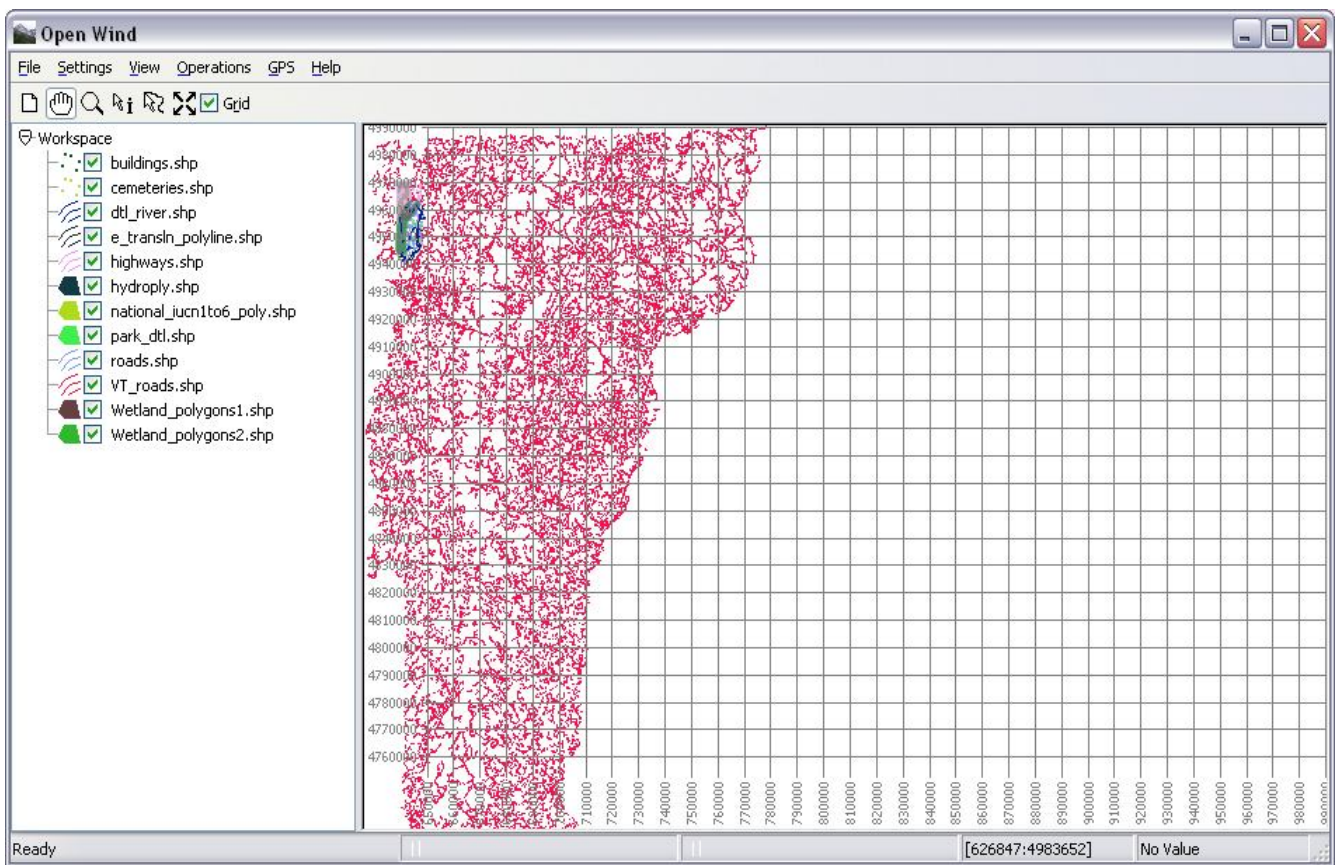


Figure 1 – Screen shot with shape files loaded.

You can reorder the layers by dragging them up or down or dropping them onto another layer.

You can use the mouse wheel to zoom in and out or the zoom mode (magnifying glass on the toolbar) to zoom in by right-clicking and out by left-clicking or to select an area to zoom in to by dragging a rectangle. Zoom in so that the little green oval in the top left fills the screen.

You can hide layers by clicking to remove their check marks or by making them children of a layer which is hidden. You can hide all the layers by right-clicking on the empty space in the left-hand panel and choosing **Hide All**. Hide all the layers then check the layers containing the buildings, cemeteries, transmission lines, hydrology, roads and wetlands. Your screen should now look similar to that below.

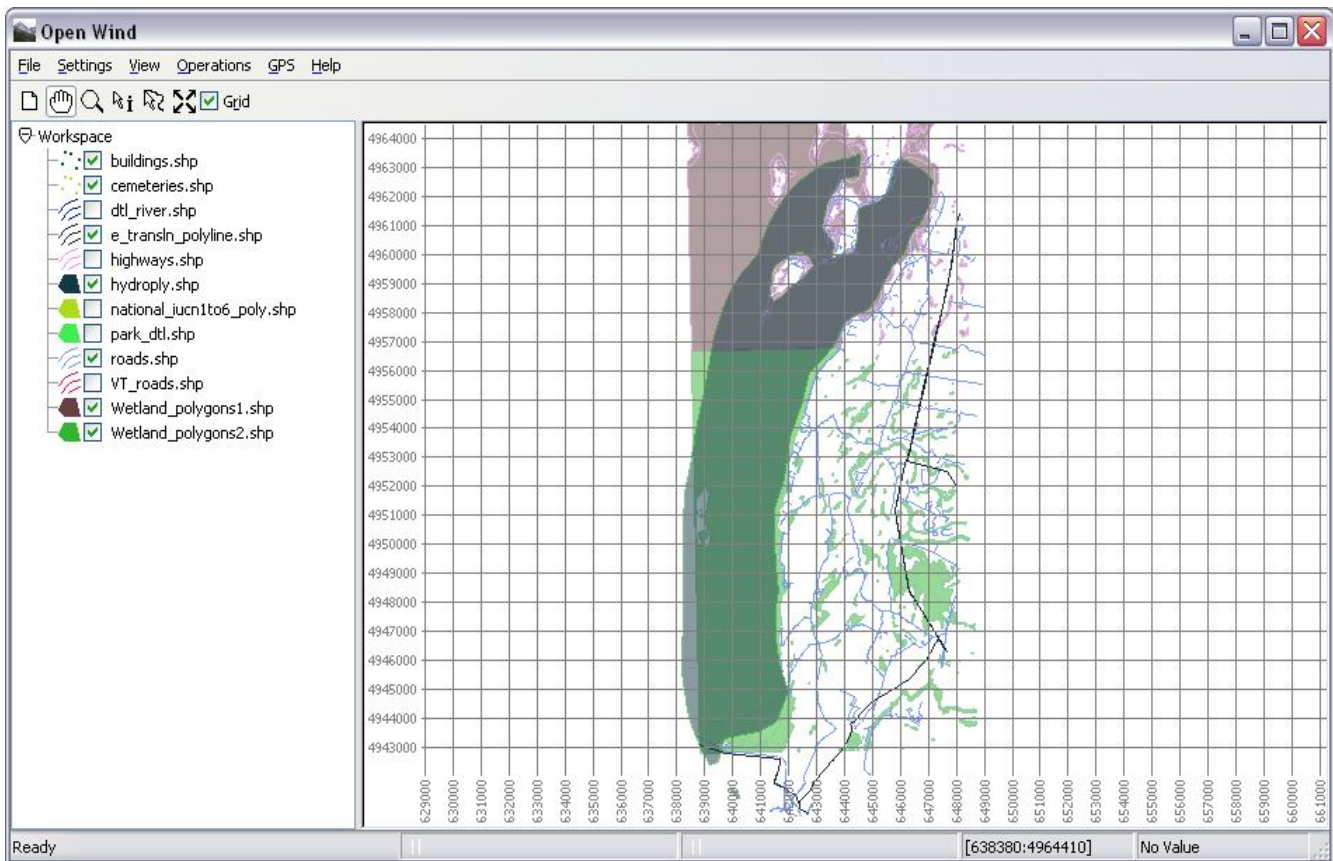


Figure 2 – a close up with some layers showing and others not.

In order to edit the properties of an individual layer, right-click on the layer in the left hand panel and choose **Properties**. Right-click on the layer called “hydroply.shp” and choose **Properties**. Now choose the **Attributes** tab and click in the checkbox next to the button marked **Label**. Now select the field whose value is “Champlain” and hit the **Label** button. Now go to the **Display** tab and check **Add Labels**. Now click OK. You will see that the hydrology polygon is now labeled Champlain. In a similar way, toponymy layers can be displayed to give more context to your project. You can also access layer properties by double-clicking on the layer in the tree view.

In order to make the rest of this tutorial clearer, please select all the currently loaded layers except the layer called “cemeteries.shp” and drag and drop the other layers onto the cemeteries layer. You may now show or hide all these shape layers by checking or unchecking the cemeteries layer.

### Loading a Wind Resource (WRG file)

Click on the **File->Open Wind Resource Grid** menu item and browse to **../tutorial\_data/Modeling/50m/Windmap/RSF80\_1E\_MERGED.WRG** and click OK. This will load up a merged wind resource grid adjusted to the two masts mentioned earlier.

The WRG itself is displayed as nothing more than a black rectangle showing the limits of the grid points. However, three raster layers have been added as children of the WRG showing the mean wind speed, mean power density and elevation data derived from the WRG. These layers can be treated as any other layer. Leave checked the child layer named "Wind Speed" and uncheck the other two. Now select the wind Speed layer by clicking on it and then press the button on the toolbar with the **i** and the arrow next to it. This moves the interface into information mode. With the "Wind Speed" layer selected, moving the cursor over the WRG should show the wind speed at each point in the lower right hand corner of the screen.

### Loading a Wind Frequency Distribution (TAB file)

This type of layer is referred to interchangeably throughout the rest of this document as "wind frequency table", "met mast" and "met mast layer".

Click on the **File->Open Wind Frequency Distribution** menu item and browse to **../tutorial\_data/Modeling/50m/Windmap/Sample\_Mast\_0403\_80-m\_Mast\_File\_12-Months\_7.33\_m-s\_18feb08.tab** and click OK. Move the mast up to be the top child of the WRG. This ensures that the wind rose can be seen on top of the wind speed layer. Now uncheck the "cemeteries.shp" layer zoom in again so that you can see the wind rose. Right-click on the layer "Sample\_Mast....." and choose **Properties**. On the first page you can see statistics relating to the TAB file including and approximate A and K value and the altitude of the base of the mast. This altitude is taken from the WRG which acts as a terrain elevation object (more on this later). The properties dialog also allows you to view the frequency table and set turbulence intensity as well as do all the things you would be able to do with any other vector point object in OpenWind. Try using the **Display** and **Attributes** tabs to get the Name of the mast to display in the map view and change the wind rose to one displaying power density.

### Loading a Point Wind Resource

Now the only thing we need to add before using this frequency distribution is a point WRG.

With the properties of the met mast layer still open and with the met mast tab showing, press the button labeled **Load Point WRG** and browse to **../tutorial\_data/Modeling/50m/Windmap/** and open **RSF80\_1E\_SAMPLE\_MAST\_0403\_80-M\_MAST\_FILE\_12-MONTHS\_7.33\_M-S\_18FEB08\_POINT.WRG**. You will notice that you can no longer edit the X and Y coordinates of the met mast.

Your Workspace should now look something like this:

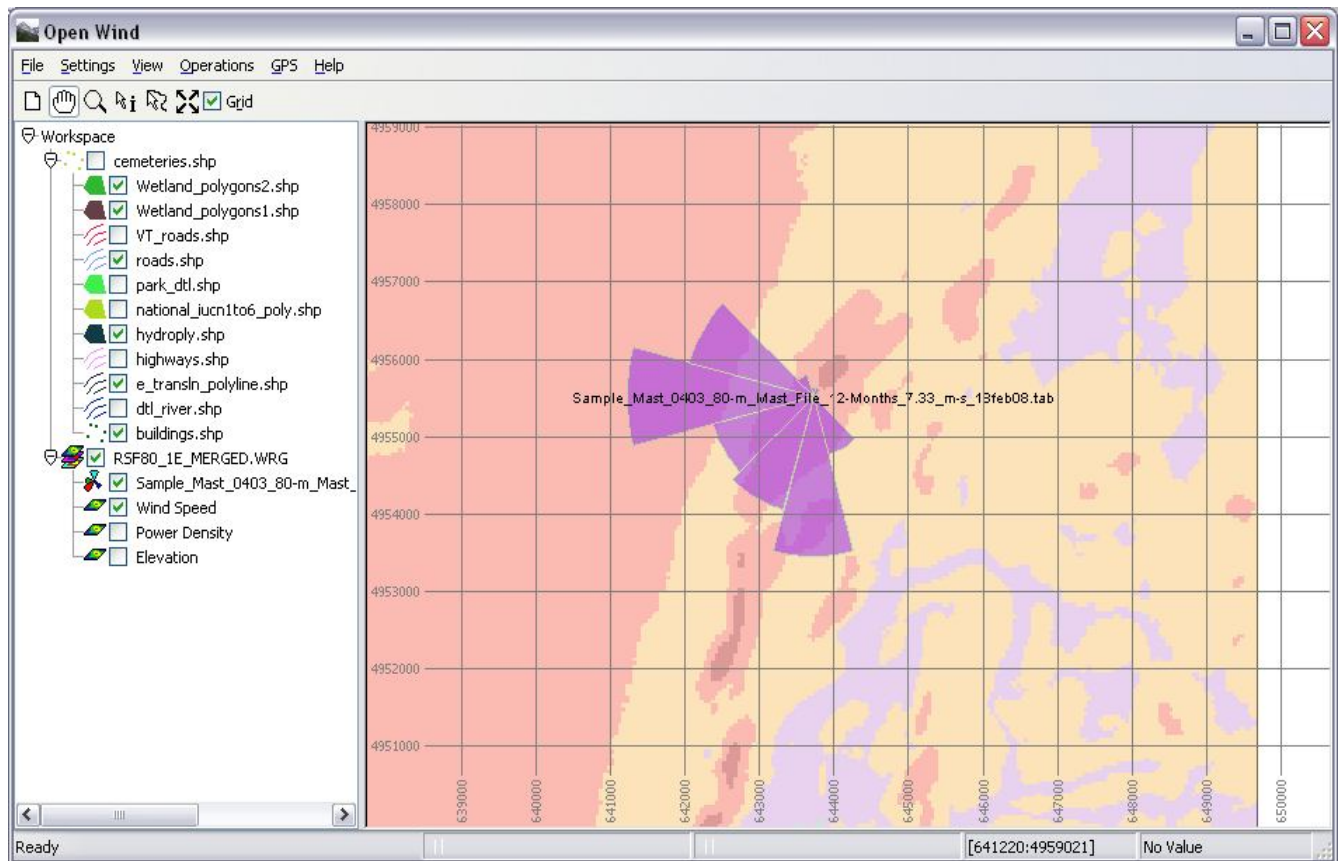


Figure 3 – showing WRG and child layers including a met mast layer.

Uncheck the “Wind Speed” layer.

### Raster Layers

Raster Layers can be used to add more context to your workspace as well adding in physical parameters such as roughness length, vegetation and obstacle height or to substitute more accurate elevation data say.

Click on the **File->Open Raster Layer** menu item and browse to **../tutorial\_data/Modeling/50m/Windmap/ned\_latlong.RDC** and press OK. As this layer is a raster grid in geographic coordinates and OpenWind only works in UTM, OpenWind will ask to resample the grid into UTM. Answer “Yes” and then set the raster resolution to be 25m, the default value for missing data to be 0.0 and choose **Bilinear** as the interpolation method.

You will notice that the new layer does not have a name. This is because Idrisi raster files can contain names and this one did not. Right-click on the layer and choose **Properties** then enter a name in the top-most edit box. I chose to name mine “Wide Elevation”.

Go to the **Color Scale** tab and choose **Rainbow** from the drop-down list on the right. Now move to the **Interpretation** tab. Choose the units “Meters” and check the **Terrain Elevation** checkbox. If you look at the elevation layer that was created when we loaded the WRG, you

will see it already had these. This is because when we load a WRG we know the meaning of the layers we are creating whereas when you load a terrain elevation raster, you will need to check these yourself. At the bottom of this dialog tab is a button **Commit Changes and Generate New Raster From Slope**. Click this and within a few moments a raster showing the gradient at every point in the Elevation layer should be added as a child.

Now move the new elevation layer to be a child of the WRG layer.

### Creating Layers

You can create a new layer by right-clicking on the empty space in the left-hand panel. Choose **New Layer** and then choose **Turbine Layout**. This creates a **Site Layer** which is the vehicle for creating a turbine layout. A Site Layer is a type of vector point layer like the “buildings.shp” that you loaded up at the beginning. It has all the same properties as a point layer except it has additional properties which are used to define the site such as turbine type, site air density, separation distance and so on.

You can add turbines to a Site Layer by switching OpenWind into edit mode using the button on the toolbar to the right of the information mode button and then right-clicking on the map view. You can also add turbines by opening the Site Layer Properties and moving to the **Turbine Coordinates** tab. Here you can choose to **Autoplace** a number of turbines or you can copy and paste a two column list of coordinates from a spreadsheet.

A fourth way to add turbines is to load the turbine coordinates as a point shape file and then right click on the point layer in OpenWind and choose to convert the layer to a Site Layer using the **Turbine Layout** option in the pop up menu.

For now, add some turbines by switching to edit mode with the Site Layer selected and right-clicking 34 times in the mapping view. Now move the Site Layer to be a child of the Wide Elevation Layer which is a child of the WRG Layer.

Go back into the Site Layer Properties and set the separation distance to be elliptical with a 4 diameters minimum separation distance and 6 diameters in the major axis with the bearing of the major axis set at 90 degrees.

After playing with a few of the layer display properties my workspace now looks like this

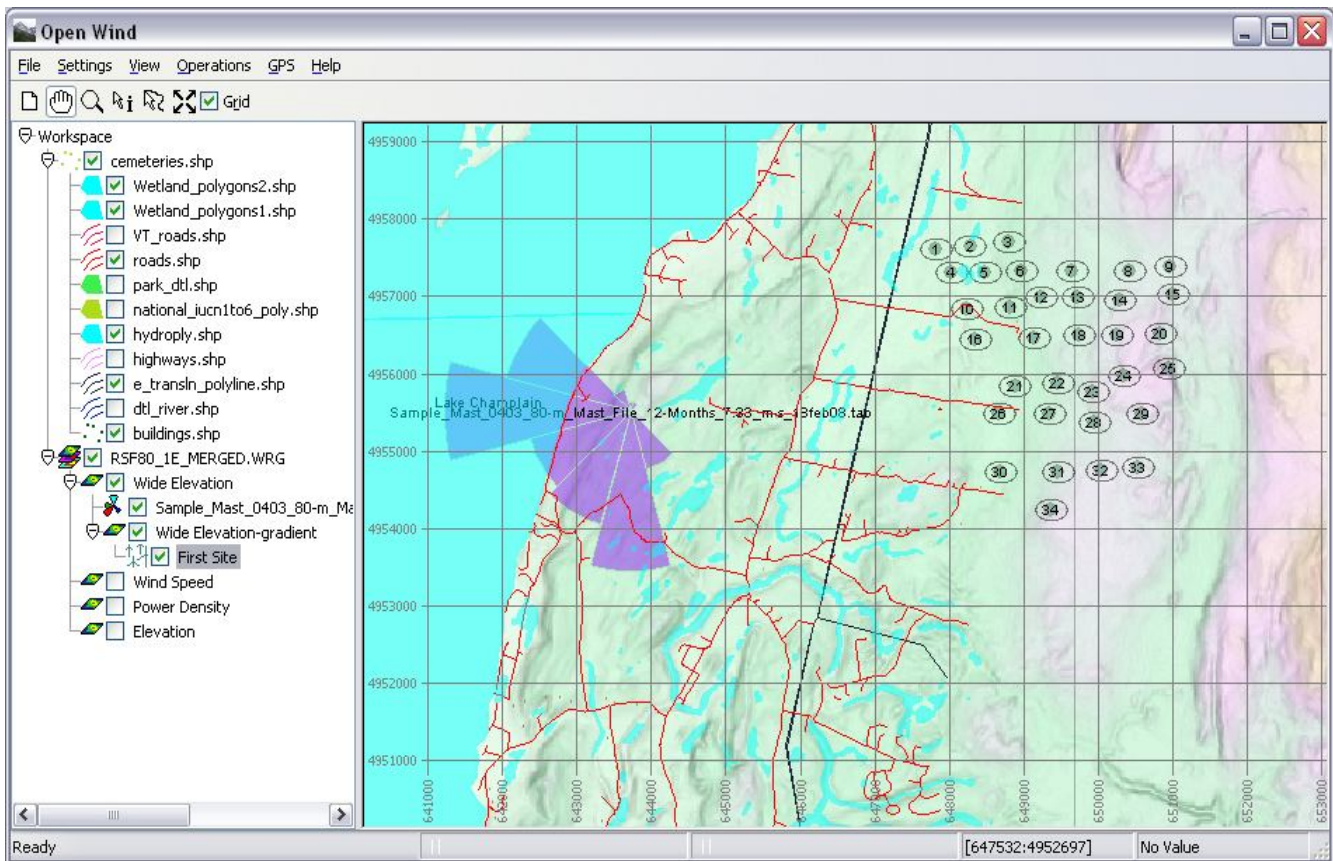


Figure 4 – showing semi-transparent elevation layer combined with gradient

Click on **Operations->Optimize Turbine Positions** to see the turbines all trundle off into the water. Click on **Operations->Stop Optimiser** to put at an end to the madness and close the **Optimiser Progress** pop-up.

We still have some work to do before we can optimise our layout.

### A Note On Layer Hierarchies Within OpenWind

This is probably as good a time as any to talk about how layers interact within OpenWind.

In the simplest sense, the layer hierarchies work like those in a standard GIS software. Layers higher up the hierarchy are drawn on top of layers lower down in the hierarchy and child layers are drawn on top of parents. However, in OpenWind this idea of hierarchies is expanded upon to give the user a powerful tool by which to organize and structure their data and assign physical meaning.

There are a few simple principles or rules which determine how layers should interact within OpenWind:

- | All interactions are bilateral. This means that any piece of logic can only exist as an interaction between two layers.
- | A layer can be directly affected by its children.
- | A layer can be affected by its parent by accessing data in the parent.

- | Layers can ask for information from other layers. When a layer asks for a piece of information, the order in which it searches is as follows:
  - First child and all its children
  - Second child and all its children
  - ....
  - Last child and all its children
  - Parent
  - Grandparent
  - ....
  - Top level parent
- | The searching of each child is done in the same order and so the first and last great-grandchildren of the first child layer are searched before the second child and its descendants are searched. All child layers (and their descendants) are searched before a parent layer is searched.
- | As soon as one layer is found which contains a valid answer, the search stops and all other candidate layers are ignored.
- | To continue the familial metaphor, layers which are not in the same line (e.g. cousins, siblings, uncles and nephews) cannot affect each other. This is very important in order to allow the compartmentalizing of logical relations within the data.

Examples:

- | An example of this then would be the met mast that we loaded earlier looking for a terrain elevation for its base. It has no children so it looks to its parent for a terrain elevation at the point it is located. Both its parent and its grandparent are qualified to provide terrain elevation information for the met mast's location. However, because it finds the information from its parent (the raster layer), it never asks the grandparent (the WRG layer).
- | Another example would be the Site Layer. In order to calculate the estimated predicted annual energy capture, the Site Layer needs access to wind resource information. It looks to its parent, then grandparent then great grandparent which provides the wind resource information for all its turbines (in this case). Of course the Site Layer would have looked through its child layers first and if it found one that covered part but not all of the site, it would have taken the wind resource information for those turbines within the child WRG from the child WRG and sought the wind resource information for the rest of the turbines from its great grandparent.

I hope you're still following. I will be giving more examples. However, the rules are just as I described above. If you can address your problem using a combination of these rules then you're in business. I will be giving more examples as we move through the rest of the tutorial.

### Adding Site Constraints

In order to stop our turbines from heading out into the water or parking themselves in somebody's back yard, we need to be able to implement site constraints.

In OpenWind, there are multiple ways to implement site constraints using a variety of input data formats. However, they all work by affecting the Site Layer's "validity". By default, a Site

Layer is valid at all points in space. However, in practice, it becomes limited by the extents of the wind resource information (in the form of WRGs) it has access to. In order to further limit the Site Layer, the simplest way is to add some child layers describing the sources of the limitations.

In the left-hand panel, drag and drop the following layers onto your site layer:

- | Wetland\_polygons1.shp
- | Wetland\_polygons2.shp
- | roads.shp
- | buildings.shp
- | e\_transln\_polyline.shp

Now before we go any further it is probably worth condensing our two Wetland layers into just one so drag one to be the child of the other and right-click on the child. A pop-up menu should appear in which **Merge With Parent** is an available option. Choose this and you should see the two layers become one. (Note: this option is usually grayed out as merging is only allowed between layers which contain the same type of data and with the same field names and types).

Now for each layer in turn we need to determine how it affects the validity of its parent (site) layer. Edit the properties of each layer and go to the **Parent Validity** tab. Check the box labeled **Use following rules to determine validity**. Then for each layer click the **Outside** option and input a buffer size in meters ranging from zero for wetlands say to perhaps 500 for buildings. I used the following values:

Name of layer	Parent Layer Is Only Valid...	Size of Buffer
Wetland_polygons2.shp	Outside	0.0m
roads.shp	Outside	100m
buildings.shp	Outside	500m
e_transln_polyline.shp	Outside	150m
Site Boundary	Inside	0.0m

You may notice that I slipped an extra layer in there at the bottom. You will want to add this too. Right-click in the left-hand panel and select **New Layer** then **Polygons** from the pop-up menus. Now with this new layer selected, press the edit mode button on the toolbar and right-click to input the site boundary. To finish editing, simply switch modes or select another layer. I switched the wind speed layer back on to help me. When you're done, your project should look like the one below. It doesn't matter if your site boundary overlaps into the water as we will implement logic to deal with that.

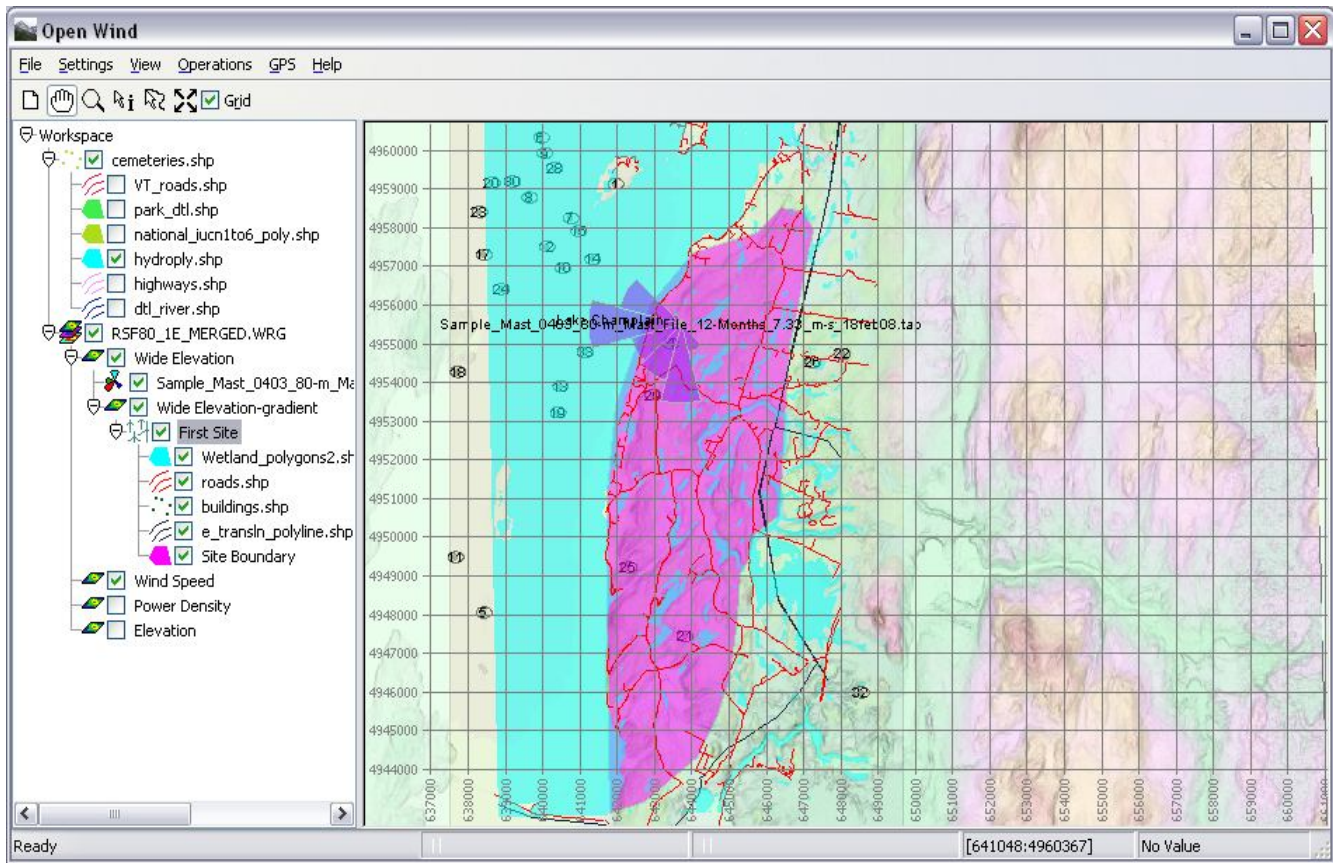
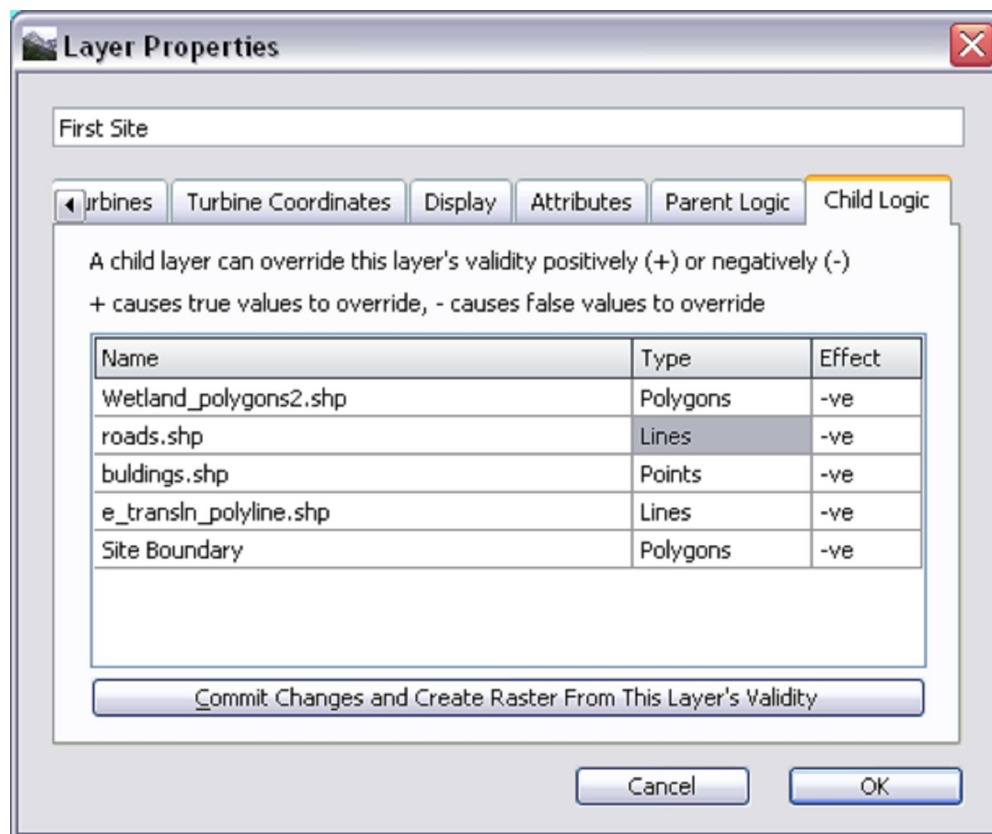


Figure 5 – showing site layer hierarchy with constraints.

So far the constraints have identified how each child layer is to be interpreted by its parent. However, we still need to determine how the parent will combine the effects of all these competing constraints. For this we need to open up the site layer properties and go to the **Child Logic** tab.



We set the “-ve” effect for each constraint layer because in each case we want to remove portions of the site area based on the logic we already set in the constraint layer itself. You will remember that for the Site Boundary layer we specified that the parent was valid *inside* the polygon whereas for the Wetland\_polygons2.shp layer we specified that the parent was valid *outside* those polygons. In general we use the negative effects of layer validity to constrain a site. We will come back to this later in the tutorial. For now, try to appreciate that we can only take either the positive or the negative effects of each constraint in order to allow the effects of multiple layers to be combined. In case of a conflict, the layer which is higher up in the list of child layers takes precedence.

Please try starting the optimiser again for a few iterations. If you stop it again, you should see that the turbines have now all moved into suitable positions, more or less. There should not be any outside the site boundary nor should there be any within 500 meters of a house or 100m of a road and slowly but fairly surely, the turbines should be shouldering their ways into the windiest parts of the site.

There is a way to check this as well as to potentially speed up the optimisation process (depending on the details of your site and whether you are happy with a slightly more approximate solution). With the optimiser stopped, please go back into the properties of the site layer and return to the **Child Logic** tab. At the bottom you will see a button which says **Commit Changes and Create Raster From This Layer's Validity**. Press this button and agree to the defaults in the next dialog box. This part might take a while, so maybe come back in 20 minutes.

## Using Rasters to Define Constraints

By now there should be a new child layer of the site called something like “First Site-derived”. This layer is a raster layer containing cells with either the value 0.0 or 100.0 (assuming you accepted the defaults). Please inspect this layer to make sure you set all your constraints correctly. Its easier to see this layer if you drag out of the site hierarchy and place it just above the WRG as a top-level layer. Uncheck the Site Boundary layer so you can see the turbines more clearly and zoom in so you can see the center of your site. Your workspace now might look something like this:

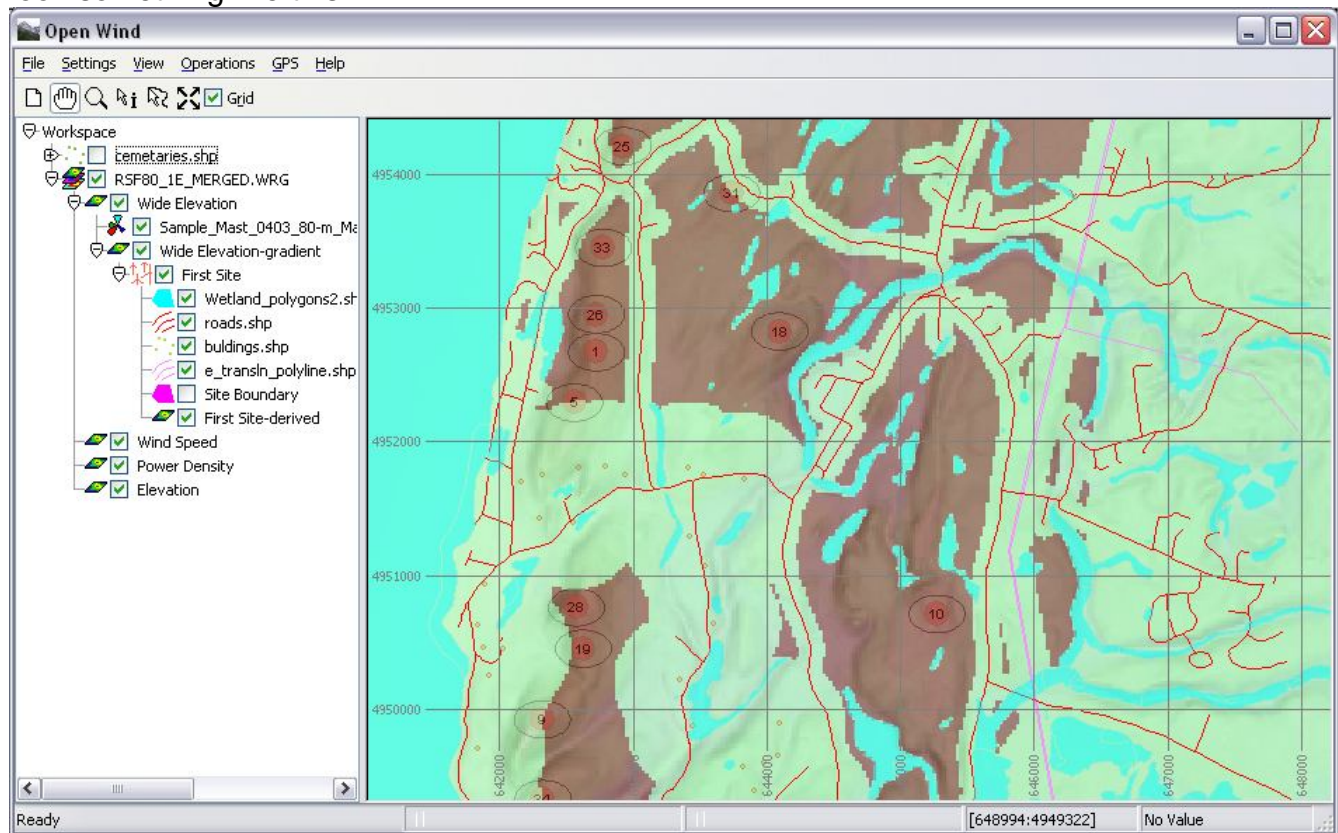
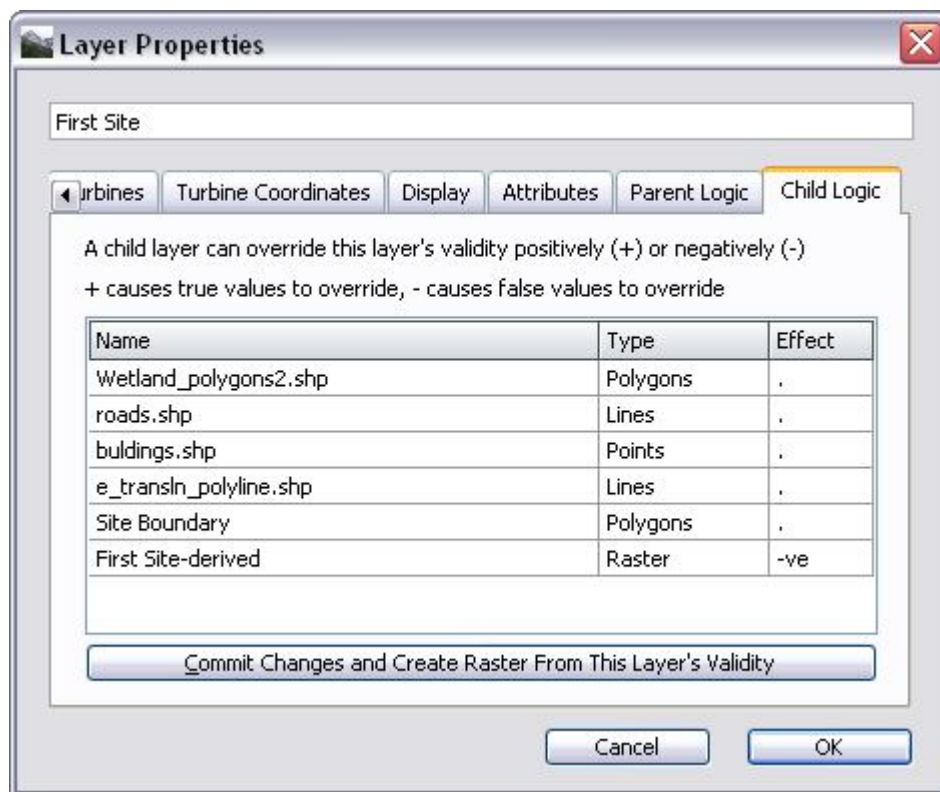


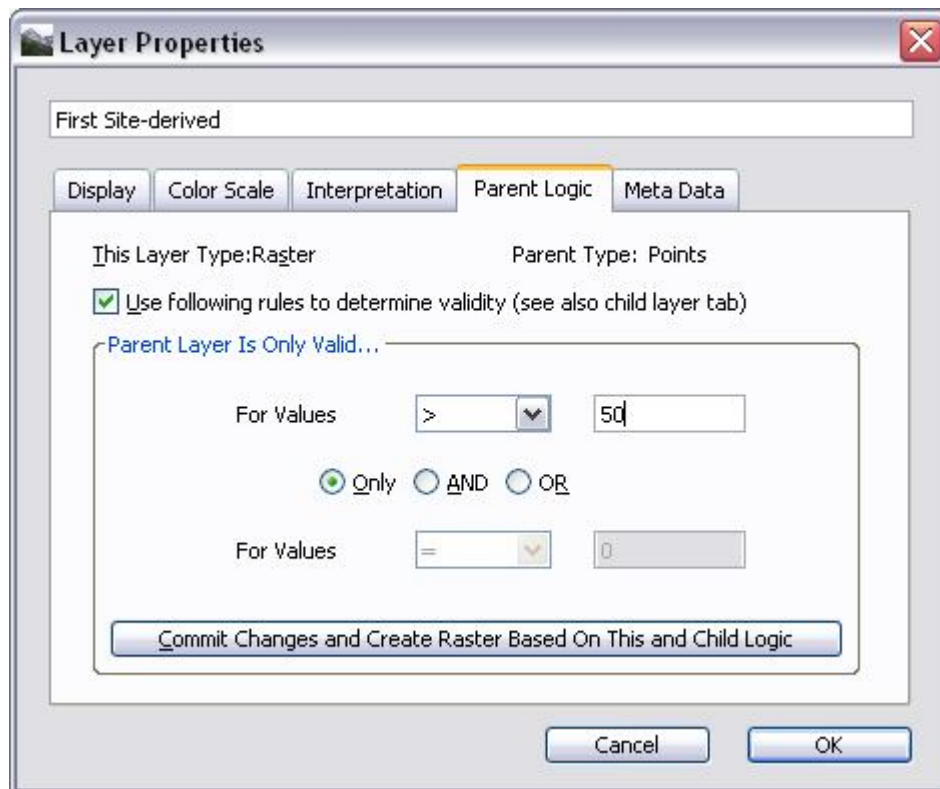
Figure 6 – showing raster constraints layer.

The brown patches in the raster are areas where the site layer is valid and the green areas indicate areas where the site layer is not valid due to the constraints we defined. As you can see, the brown areas avoid the water and stay the required distance from the roads, power line and houses. It also contains all the places where the optimiser has positioned turbines. This means our constraints are working. Congratulations on making your first OpenWind, highly constrained site. As you can see it requires very little input from your GIS department apart from to supply you with the basic layers in an acceptable format and projection.

Save your data now (if you haven't already) by clicking **File->Save As...** and saving to a file. Now, drag the raster showing the effect of the constraints to be a child of the site layer again and then edit the raster layer's properties in the following manner:



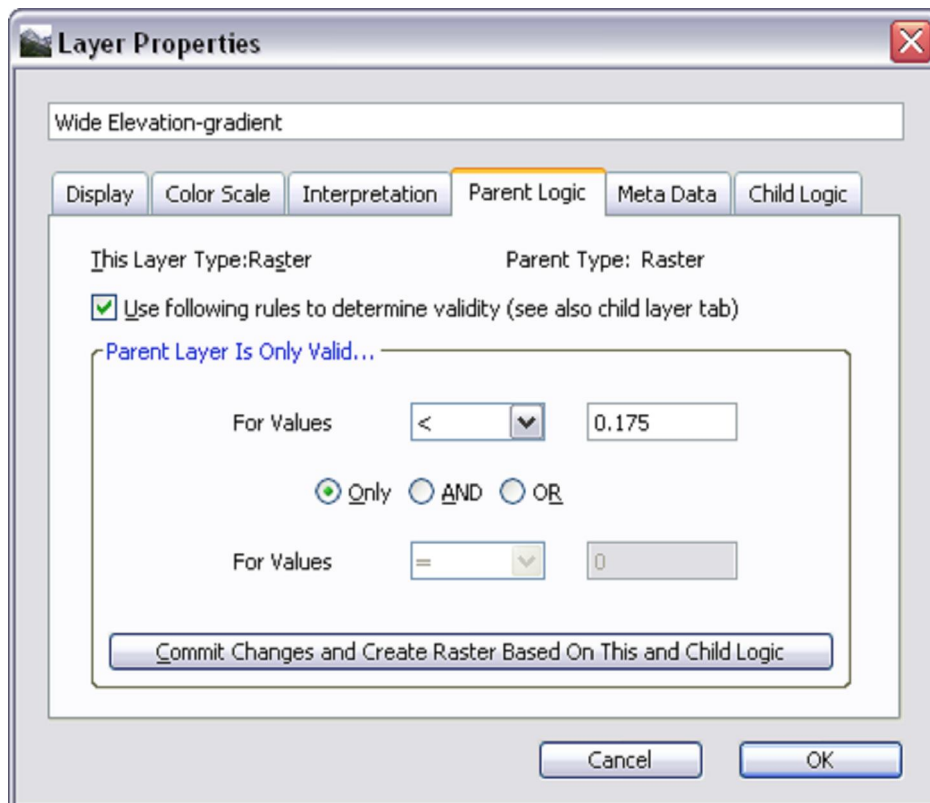
Press OK and now open the site layer's properties and change the Child Logic tab to look like this:



Press OK again. Now start the optimiser again. You should notice that the optimiser runs faster. This is because now it only has to check the value of one raster rather than interrogating five separate layers and using line-crossing algorithms to test whether it is inside or outside of various buffers and polygons. In very complex sites, it may be worthwhile to invest a few hours of computing time to creating an accurate, high resolution raster which encompasses all the other site constraints. Alternatively you can always add a small safety margin to your constraints and use a raster at fairly low resolution like that we just made. As with most things in OpenWind, its entirely down to how you want to go about it.

### Terrain Gradient as a Constraint

One common constraint which we didn't use was terrain gradient. This can be added by rearranging the layer hierarchy such that the terrain gradient is a child of the site layer. In order to do this you will need to move the site layer to become a child of the "Wide Elevation" layer (or if you're happy with the elevation data in the WRG you can move make the site layer a child of the WRG). Then move the terrain gradient layer to become a child of the site layer and edit the elevation gradient layer's properties such that it invalidates its parent when the terrain is steeper than say 0.175 or approximately 10 degrees from horizontal.



Making this a "-ve" effect in the site layer's child logic would complete the addition of limiting the site with respect to terrain gradient.

## Two or More Meteorological Masts in One Site

Well we're just about finished for this short introductory tutorial. There are many aspects of the software which you still haven't seen but hopefully you get the main ideas behind the design of the interface. Most of what you haven't yet seen is easily explored by playing with the properties of the various layers.

However, I just want to take you through a slightly more complicated and realistic configuration of the site we've been playing with.

Please click on **File->Open Wind Frequency Table** and browse to **../tutorial\_data/Modeling/50m/Windmap/** and open the file named **Sample\_Mast\_4723\_80-m\_Mast\_File\_12-Months\_6.66\_m-s.tab**. Once loaded, open this new layer's properties and load the single point WRG called **RSF80\_1E\_SAMPLE\_MAST\_4723\_80-M\_MAST\_FILE\_12-MONTHS\_6.66\_M-S\_POINT.wrg** from **../tutorial\_data/Modeling/50m/Windmap/**.

Now, we could split the site up into two site layers with two separate polygon layers acting as site boundaries. Each site would then have its own WRG and met mast layer. Such a workspace could look like this:

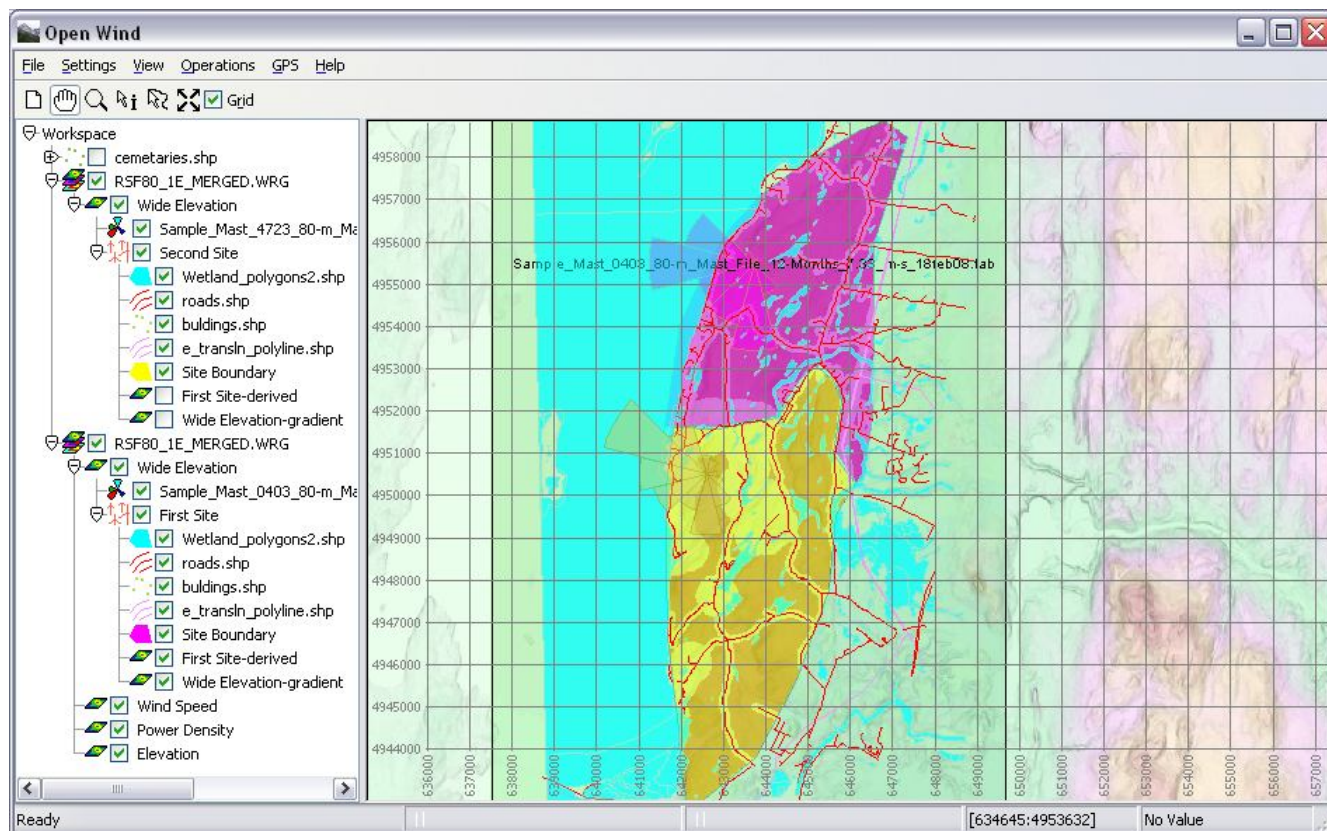


Figure 7 – showing two site layers, each with their own site boundary and constraints

In order to change to this configuration, an easy and quick way is as follows:

1. Right-click on the WRG layer and select **Copy**<sup>1</sup>
2. Right-click on the “Wide Elevation” layer and select **Copy**
3. Set the number of turbines in the site layer to 0 using the **Autoplace** function in the **Turbine Coordinates** tab of the site layer properties
4. Right-click on the site layer called “First Site” and select **Copy All** which will copy the layer and all its children
5. Rename the extra site layer to “Second Site” and rearrange the new layers as shown above
6. Modify the site boundary layers in order to split the original site in two
7. Modify the Child Logic in both site layers to include the negative effect of the new site boundary layers.
8. Change the colors of one of the site layers so you can see whose turbines belong to whom.

Now add 15 turbines to each site layer and start the optimiser....

Let it run for a while. You can keep reading while you watch the optimiser at work. As you can see the turbines from one site stick to one site boundary and turbine from the other site layer stick to its site boundary. The optimiser still makes sure not to let them violate each others separation distances. Also, you could have made the site boundaries complex shapes with holes in or consisting of several separate polygons as part of one layer or indeed several layers. So long as you specify how the site layer is to interact with these polygons, it should all work.

Now if you stop the optimiser, and run an energy capture, you will get a short report and near the top of the report you can see summary information about each turbine's energy capture. If you look at each line end, you can see which WRG and wind frequency table was used for each turbine.

So if you got this far then everything appears to work as you would expect but is it the way you would ideally like? Surely, if you are going to optimise a site using two or more masts, and you're taking the time to allocate ground space to different masts, you do not also want to have to intuit how many turbines should ideally be in each space. I thought not and nor should you have to.

Rearrange your workspace to look like the following:

---

<sup>1</sup> Note that when you copy a WRG, only the WRG layer is copied. The underlying WRG data is shared by the two layers.

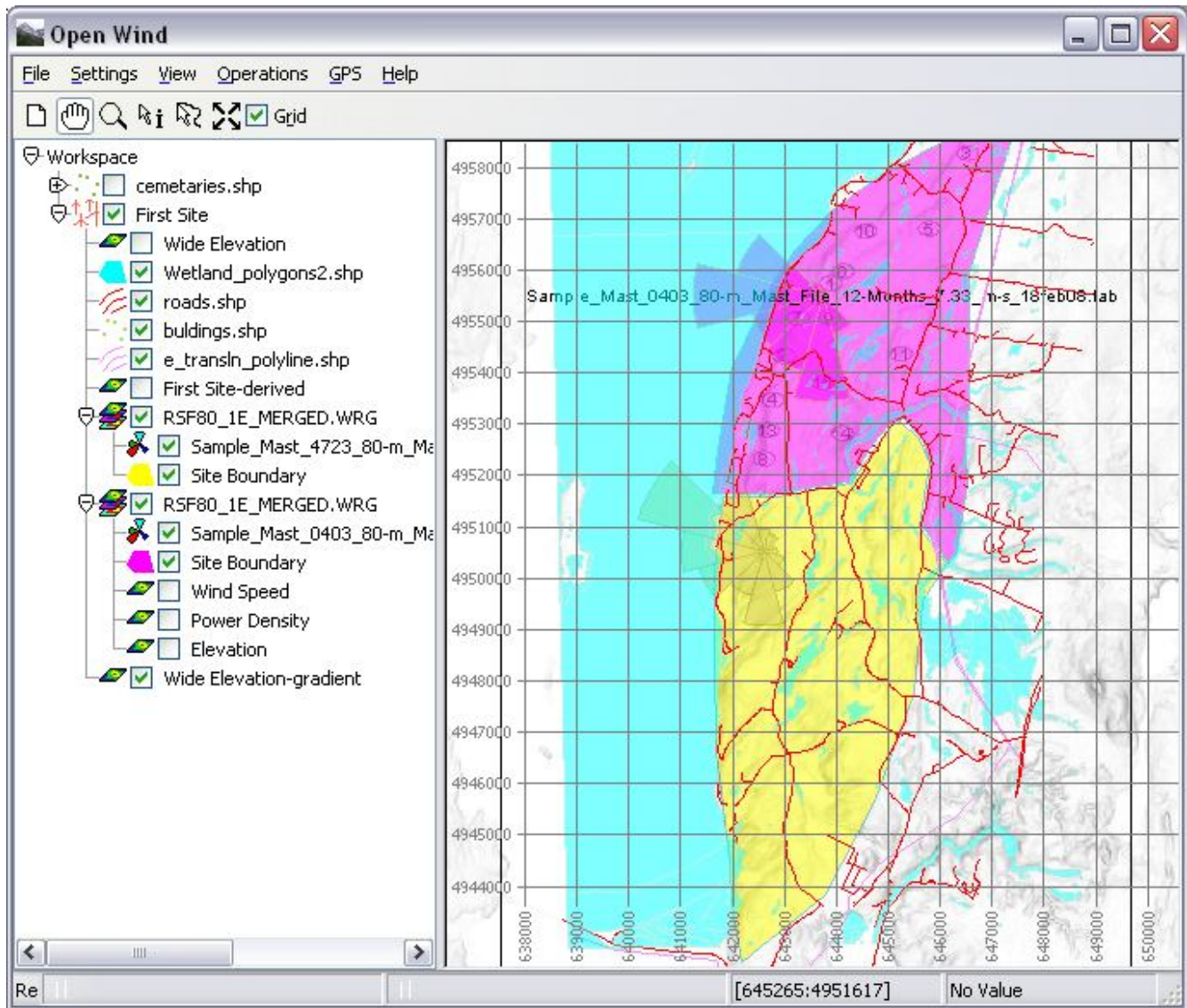


Figure 8 - showing one site making use of two met masts, each with its own area

Save your workbook before you do this as it might take you a couple of tries to get the most efficient way of doing it without re-entering data.

Now, as you can see from the structure above, The “First Site” layer remains much the same. We extracted the site boundary from “Second Site” before removing the layer and all its remaining children (when you remove a layer it gets deleted from memory along with any children it has at the time). Now, the WRG layers are children of the site rather than the other way around but the wind frequency tables or met masts are still children of the WRG layers. In this case the restrictions on the site layer are the same as in our single met mast case except now, instead of using the polygon layers to directly limit the site layer, we use them to limit the validity of the WRG layers. This means that when the site layer asks for wind resource data, each WRG will check its own validity and finds itself limited to the extents of its child site boundary. If the request for wind resource data lies outside its area of validity, it will tell the site layer that it is not valid and the site layer will continue looking for a valid wind resource. In this way, one site can optimise smoothly over two or more WRGs, each with its

own complex area of validity and its own wind frequency table.

Now this may seem a little overcomplicated compared to what you're used to but it is necessary in order to give you the freedom you need to solve your own wind farm design problems. Remember, the rules are simple and strict so once you understand them, you're only limited by your own creativity – or at least that is the intent.

If you find a case in which the software does not work the way you think it should according to the rules described here, please let us know and we will fix it.

Add a few more turbines if you feel so inclined and press optimise to see the turbines choosing which site boundary they want to be in. Once it has optimised for a few iterations, stop the optimiser and do an energy capture to check that the appropriate turbine is having its energy capture determined by the correct wind frequency table.

This concludes this introductory tutorial. We hope you have a flavour of what the software may be capable of. If you think you've come up with a novel way to structure a project, please post in the Tips and Tricks forum on <http://awsopenWind.org>

Other features of OpenWind, not covered here, include: output to Google Earth; calculation of zones of visual impact; support for GPS devices; defining validity on the basis of raster or image colour values; importing terrain and roughness values as vector or raster; and so on. The list grows as we continue to think of more things it should do.

## Epilogue

OpenWind have recently reconsidered their development strategy and have decided to stick to software for the foreseeable future.