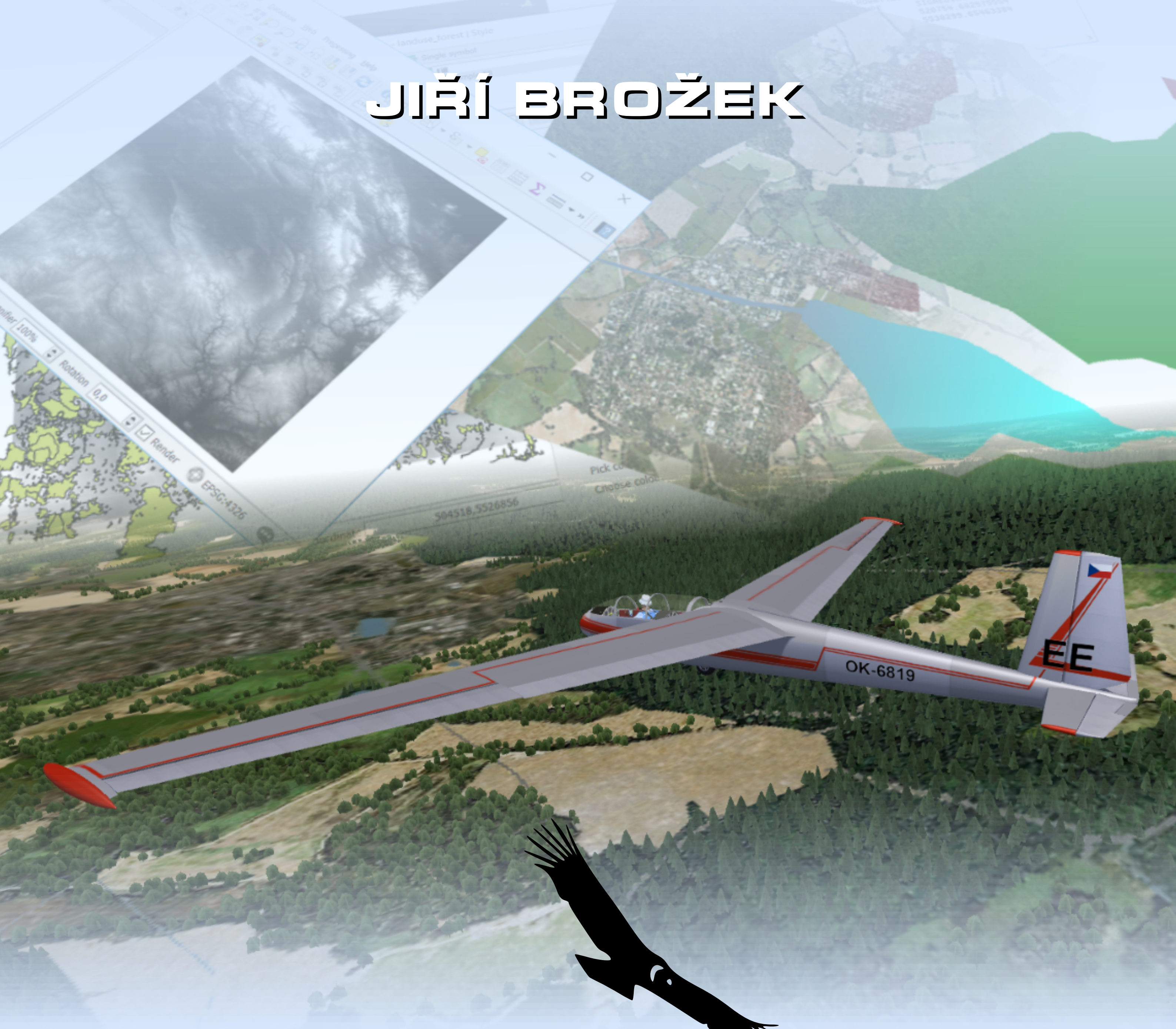
# CONDOR LANDSCAPE GUIDE

## STEP BY STEP GUIDE TO CREATING LANDSCAPES FOR CONDOR V2

JIŘÍ BROŽEK




CONDOR2

THE COMPLETE SOARING SIMULATOR

**Condor Landscape Guide – revision 2, November 2019**



Special thanks to Ben Fest, Erwin de Bruin, Michel Bernard, Miloš Koch, Chris Wedgwood and Uroš Bergant

# Table of contents

# 1. Introduction

This guide will provide necessary information for everyone interested in creating their own sceneries for the Condor competition soaring simulator version 2. The whole scenery creating process will be described on creation of a small demo scenery. While many of the steps during the process can be done in various software products, in this guide we will only use software which can be used free of charge or is open source.

The text of the guide is divided into several chapters.

- Chapter 2 we will provide brief overview of all the new landscape features Condor V2 offers in comparison with the V1.
- Chapter 3 provides overview of all technical information related to Condor V2 sceneries.
- Chapters 4 and 5 are the first part of the scenery tutorial. They describe the process of getting and processing the source height data, importing data into Landscape Editor and exporting the base terrain of the scenery into Cv2.
- Chapter 6 describes how to add textures, forest maps, thermal maps and water to the scenery.
- Chapter 7 provides brief introduction to modelling custom airports and objects, explains adding airports, custom objects and turn-points to the scenery and concludes the demo scenery tutorial.
- Chapter 8 focuses on all issues related to conversion of V1 scenery for use in V2.

## 2. What's new in Condor v2 sceneries

Condor version 2 (V2 from now on) offers the same level of openness for add-on scenery makers as the original Condor version 1 (V1 from now on). Everyone can create a scenery and provide it for use by the V2 user base.

The key differences and new features in V2 sceneries are listed below.

- Terrain resolution has increased to 30 m instead of 90 m in V1. Terrain for V2 sceneries is built from 1 arc second DEM data. Resulting terrain mesh is crisper and better portrays various landscape features that were "averaged" in V1 sceneries with terrain made from 3 arc second data.
- Standard scenery tile texture size is now 8192x8192 (i.e. 2048x2048 pixels per patch).
- Tree maps (forest maps) now support resolution up to 2048x2048 pixels per scenery tile, allowing for better precision of tree placement.
- All the trees automatically generated using tree maps now have realistic size and tree areas have higher tree density in comparison to less dense oversized trees in V1.
- Scenery textures now can contain definition of water areas that are rendered using shaders with realistic water effect.
- The whole landscape and all the objects are now dynamically lit and shaded based on the date and time.

# 3. Technical overview of Condor V2 landscapes

Before we start creating landscapes, we need to explain how Condor treats geographical data and how the landscapes in Condor work.

The first important fact is that Condor maps all the terrain on a plane. As Earth is basically a sphere (or ellipsoid to be more precise) we need some way of projecting the real geographical data to a 2D plane. Condor utilizes UTM projection with WGS 84 datum for such purposes.

Projecting large areas leads to certain level of projection error. To avoid such issues, UTM divides the surface of Earth to 60 UTM zones that are used during projection. However, Condor is not able to work with multiple zones in one landscape.

If you decide to make a large scenery that overlaps multiple zones, you must choose only one UTM zone to work with, preferably the zone where most of the scenery area lies (or center of the scenery). The part of the scenery belonging to the other zone(s) will be subject to a certain level of projection error. It is recommended not to go deeper than 2-3 degrees inside the neighbouring zone(s).

**Use UTMtools application to check whether the area you intend to include in your scenery lies inside one UTM zone or overlaps multiple zones, and to convert coordinates from other zones to the center zone UTM coordinates.**

Condor scenery is divided into “patches”. In Condor the patch is the basic physical landscape construction unit covering the square area of 5.76 x 5.76 km. As smaller sceneries consist of hundreds and larger sceneries of thousands of patches, working on the patch level could be quite inefficient and time consuming. To make creation and processing of textures and related data maps (forest maps etc.) easier, the patches are organized in “tiles”. Each tile consists of 16 “patches” or “sub-tiles” organized into 4x4 matrix, making its dimensions 23.04 x 23.04 km.

**Easting (width) and northing (height) dimensions of Condor landscape must be whole multiples of 5.76 km, thus it is possible to create landscapes where some of the texture/map tiles are actually incomplete.**

**In our tutorial, we will be working on the whole tile level, as it is the easiest way of working in the Landscape Editor. When working on the tile level, dimensions of the scenery must be whole multiples of 23.04 km.**

When working in Landscape Editor, everything is defined on the tile level. The whole scenery is basically a matrix consisting of $N_c$ columns and $N_r$ rows of tiles. Each file related to the particular tile contains tile coordinates in its filename. There is no limit for the scenery size, apart from memory space and limitations of the UTM projection.

Tile coordinates are zero-based and general structure of the coordinates is CCRR, where CC is the number of the particular column and RR is the number of the particular row – both CC and RR include leading zero if their value is less than 10. Origin zero tile coordinates are always in the bottom right corner of the scenery. Maximum column coordinate is $N_c - 1$, maximum row coordinate is $N_r - 1$.

*For example: 4x3 scenery consists of 12 tiles organized into 4 columns and 3 rows. Coordinate numbers of both columns and rows start at 0 and are counted from the bottom right corner (4 columns = 0, 1, 2, 3 and 3 rows = 0, 1, 2). Coordinates of the bottom right tile are 0000, coordinates of the left top tile are 0302.*

Patch coordinate system is based on the same principles as the tile coordinate system. Patches are automatically created from source tile files by various tools of the Condor Landscape Toolkit.

*For example: The 4x3 tile scenery from the previous example is divided into 192 patches organized into 16x12 matrix. Coordinates of the bottom right patch are 0000, coordinates of the top left patch are 1511.*

**Warning: Although the coordinate system is the same as in V1, processed textures from V1 cannot be used in V2 without re-editing as they are rotated by 180 degrees. More on this in chapter 6.**

**If you need a quick way of telling which tile the patch belongs to, just divide each of its coordinates by 4 and the whole part of the result tells you the particular coordinate of the parent tile. Both patch and tile coordinates are also shown in the Landscape Editor.**

***For example: To find the source tile of patch 1307, we divide both coordinates by 4:***

***13 / 4 = 3.25 – whole number part of the result is 3***
***7 / 4 = 1.75 – whole number part of the result is 1***

***We get the tile coordinates by combining the whole number parts of the results – this patch is part of the tile 0301.***

# 4. Creating terrain for our landscape

This chapter describes the first steps of the scenery creation tutorial. We will be using **QGIS** open source GIS software to process the terrain data and **RawToTrn** and **Landscape Editor** from **Condor Landscape Toolkit** for their import to Condor.

We have already mentioned few important facts about Condor landscapes. The UTM projection and the patch size of 5.76 km and tile size of 23.04 km will now be very important.

As previously mentioned, terrain resolution in V2 increased to 30 m (1 arc second). Data in such resolution are now available from multiple sources for the whole world for free. In our example we will use the data from USGS using their Earth Explorer service.

However, before we start downloading the data from Earth Explorer, we should first decide what area our scenery would cover and find at least approximate coordinates of its top left and bottom right corner.

To keep our example simple, we will use only a small area in the center of the Czech Republic. You can of course use different location and increase the size of the scenery. All the steps described in the tutorial will apply.

Coordinates of the top left and bottom right corner can be estimated from a paper map or read from various electronic map sources or even GIS. In our case we will use Google Maps.

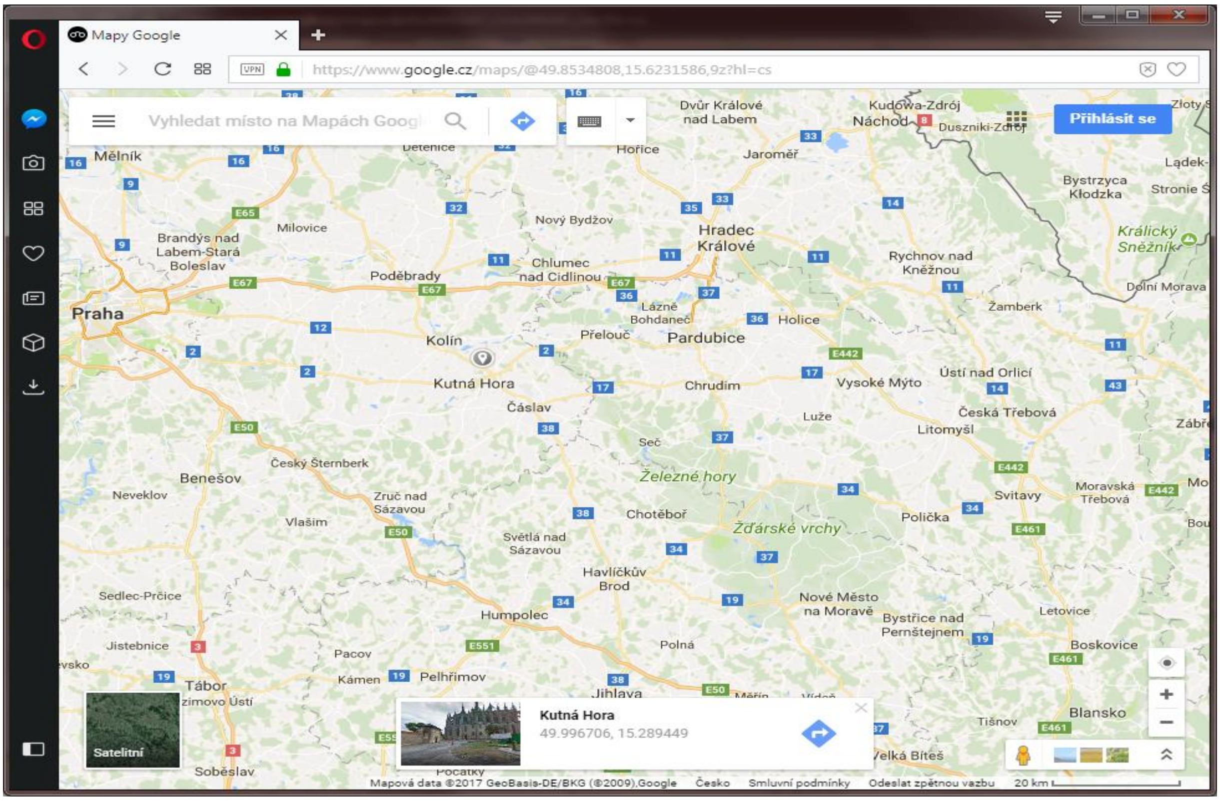



*Figure 1 – Finding the coordinates in Google Maps*

Open the map and zoom in or out to show the whole area you want to include. Then click on the place in the map where the top left corner of your scenery will be located. If you want to pinpoint the location more precisely, just zoom in. Small popup window should appear at the bottom of the map, showing the name of the town or village area this location belongs to and the decimal geographic coordinates of the selected point. Write down the coordinates or copy paste them to UTMtools. In our case the top left coordinates are N 49.996706 and E 15.289449. Those coordinates are where the top left corner of our scenery really will be.

We will repeat the same steps for the bottom right corner of our area of interest. As our area contains the ridge of Iron Mountains, which is used for ridge soaring, we will try to include as much of it and our estimated bottom right corner will be a bit south east of Ždírec nad Doubravou at N 49.666849 and E 15.848711.

Unlike the top left corner coordinates, which really represent the top left corner of our scenery, the bottom right coordinates only represent boundary of our area of interest as the dimensions of the final landscape must be whole multiples of tile size.

**As terrain height data is available for the landscape area only, Condor doesn't allow flying closer than 5 km (one patch distance) to the landscape border. There, the glider will face an invisible wall which will prevent it from flying farther.**

**This must be taken into consideration when planning what area will your landscape cover. If you want to include a point of interest (airport or important turn-point) that turns out to be very close to the border, it is recommended to expand the area in this direction. Otherwise it may not be possible to fly safely in its vicinity.**

Once we have both top left and bottom right corner lat/lon coordinates, we need to convert them to the UTM coordinates. We can use one of the various online lat/lon to UTM converters for this purpose, such as http://home.hiwaay.net/~taylorc/toolbox/geography/geoutm.html or we can do all calculations automatically using UTMtools. We will show both approaches in this tutorial. Be advised that if your scenery spans multiple zones, you may need to calculate the UTM coordinates using the center zone, not the zone the point really belongs to. This usually isn't supported by online converters. Use UTMtools for this purpose.

**See http://geokov.com/education/utm.aspx for nice overview of UTM projection, UTM zones and UTM coordinates theory.**

## Manual calculations

Conversion of the coordinates gives us results as decimal numbers with many decimal places. For the scenery making purpose we only need the whole part of the number. UTM coordinates consist of **easting** value (**x axis** – longitudal coordinate) and **northing** value (**y axis** – latitudal coordinate). We also need to know which zone the coordinates belong to and at which hemisphere (N or S).

UTM coordinates of our top left corner are: x = 520745, y = 5538304, zone 33N.

UTM coordinates of our bottom right corner are: x = 561243, y = 5501936, zone 33N.

As we are inside one UTM zone, we can take advantage of the fact that UTM coordinates are in fact distances in meters from a given zone origin point and simply calculate the dimensions of our area of interest by subtracting easting and northing values of both boundary points.

Thus:

**Easting dimension = $easting_{right}$ – $easting_{left}$** => 561243 - 520745 = 40498 meters

**Northing dimensions = $northing_{top}$ – $northing_{bottom}$** => 5538304 – 5501936 = 36368 meters

The dimensions of our area of interest are 40498 x 36368 meters. Given the size of a scenery tile is 23040 meters, our area then spans approx. 1,76 x 1,58 tiles (or approx. 7.03 x 6.3 patches). As we will be working on the tile level and the landscape must consist of whole tiles, we have two options. We can either discard the part of the area belonging to incomplete tiles and make a smaller landscape or include more terrain to make the tiles complete and our final landscape will be somewhat larger. For our example we choose the latter and our landscape will have final dimensions of 2 x 2 tiles.

As we now know the final extents of our landscape, we can calculate the new bottom right UTM coordinates. This time they will really represent the boundary of our Condor landscape.

We will start with our top left corner coordinate pair and will add 23040 for each tile in the easting direction and subtract 23040 for each tile in the northing direction. The general formulas will be as follows:

$$easting_{right} = easting_{left} + (easting_{tiles} * 23040)$$

$$northing_{bottom} = northing_{top} - (northing_{tiles} * 23040)$$

The calculated bottom right UTM coordinates for our landscape will then be:

$easting_{right}$ = 520745 + (2 * 23040) = 566825

$northing_{bottom}$ = 5538304 – (2 * 23040) = 5492224

We can also easily find the coordinates of the remaining two corners of the scenery. Top right corner will share the same northing with top left corner and the same easting as the bottom right corner and bottom left corner will have the same northing as bottom right corner and the same easting as top left corner. The coordinates for our example will then be:

**Top right = $easting_{right}$, $northing_{top}$** = 566825, 5538304

**Bottom left = $easting_{left}$, $northing_{bottom}$** = 520745, 5492224

Write down the coordinates of all four corners as they will be used later. We can also convert them back to latitude/longitude in decimal degrees using the same tool we used before. Decimal latitude/longitude coordinates will later be used for calibrating the terrain.

## Using UTMtools

Another approach to calculating bottom right UTM coordinates is using the UTMtools application. This approach is very simple, as all you need to do is to enter decimal latitude and longitude coordinates of the boundary points of our area of interest into the application and the rest of calculations is done automatically. If the landscape dimensions are not whole multiples of the tile size, UTMtools will suggest extension to the nearest larger landscape with whole tile multiple dimensions by default. The size of the landscape in tiles than can be further adjusted.

UTMtools also automatically calculates calibration point coordinates for each of the landscape corners that can be saved to CalibrationPoints.csv so they can be used for re-calibrating your scenery in Landscape Editor, if later needed.

Note the TL easting, TL northing, BR easting and BR northing values or keep the UTMtools open.

As we have calculated the boundary coordinates of our landscape, we can move to downloading the terrain elevation data from a selected data source.

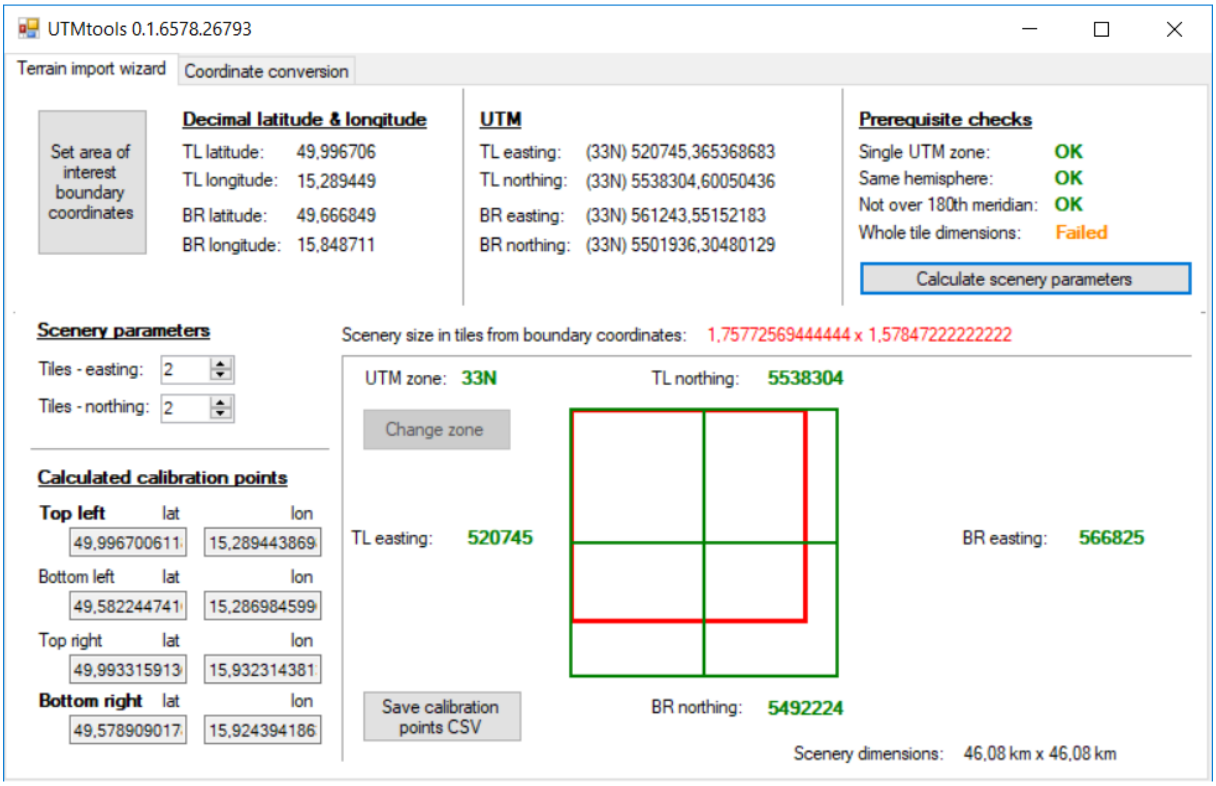



*Figure 2 – UTMtools*

## Getting the terrain data

There are multiple sources for digital elevation model data. In our example we will use 1 arc second data from Earth Explorer service run by the United States Geological Survey (USGS). The Earth Explorer is located at [https://earthexplorer.usgs.gov/](https://earthexplorer.usgs.gov/).

To download the data, you first need to register and create you user account. Registration is free.

After registering, we can proceed with selecting the area we want the data for. We can do this either by selecting the area in the map or by entering coordinates of the area. As we have calculated coordinates of all four calibration points in the previous step, we can now use them to specify the area we want to download the data for.

Now it's also time to consider expanding the area by certain margin as some distortion may occur when re-projecting the data and our intended area may not be fully useful.

Try to visualize the position of your landscape in the latitude/longitude grid. It is situated somewhere inside a square or rectangle area bordered by parallels and meridians. If any of the landscape area borders is very close to a bordering parallel or meridian, you should consider adding certain margin at that direction to include more terrain. If the landscape border is far enough from the bordering parallel/meridian, no margin is needed.

As the SRTM data are provided as 1-degree tiles and a whole tile needs to be imported even if only a tiny piece of your landscape is located in the particular tile, overlapping over a parallel or meridian will add the whole additional 1 degree area of surrounding terrain.

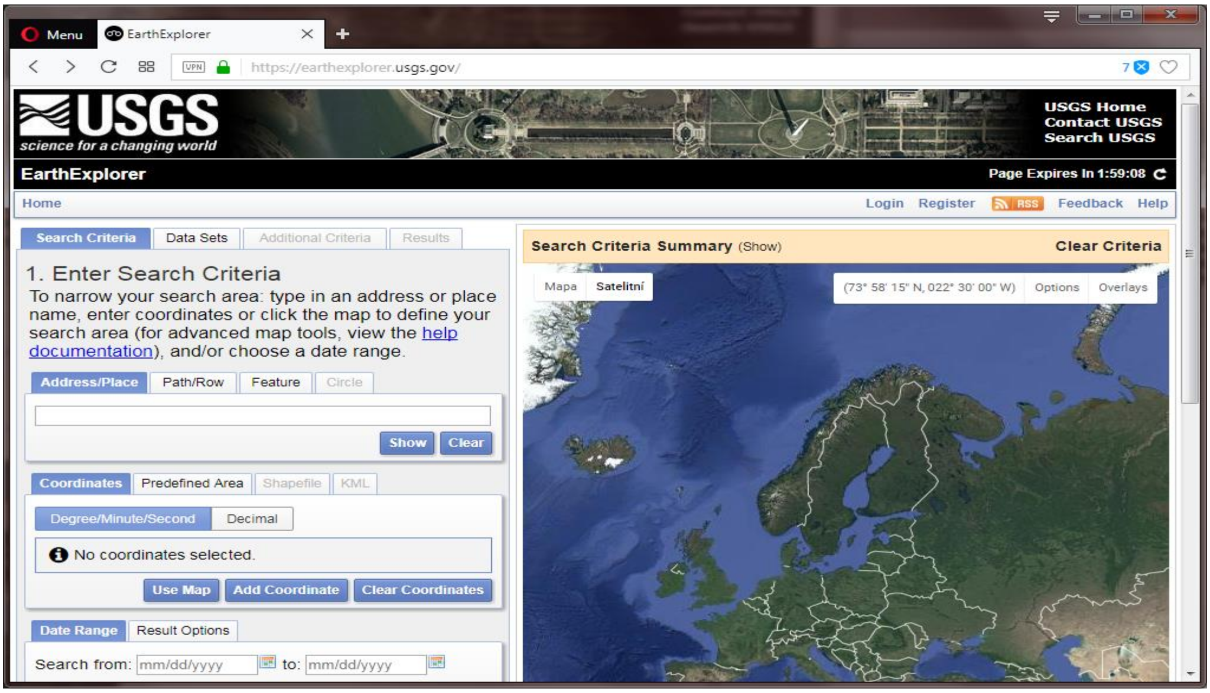



*Figure 3 – USGS EarthExplorer website*

There is no exact formula for estimating the margin value. The rule of thumb for the estimation is *the larger the landscape area, the bigger the margin*. Try to experiment with the value. If the margin turns out to be too small, increase it and re-download the data. As the dimensions in degrees of the landscape in our example are approximately 0.635 x 0.417, we will only add margin area of 0.25 degrees. Margin of 0.5 degrees should be enough for large areas.

The following picture contains a visualization of our landscape area in the latitude/longitude grid. As we can see, top and right borders of our landscape are very close to the 50$^{th}$ parallel and the 16$^{th}$ meridian respectively. Adding margin at the top and right side will ensure that the three neighboring tiles will provide big enough terrain "buffer" for processing.

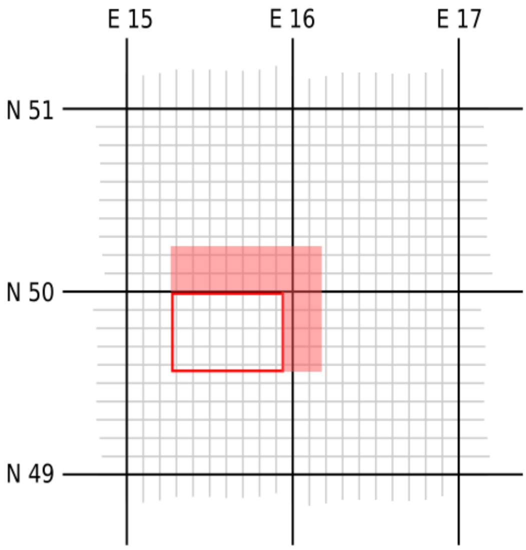



*Figure 4 – Planned landscape era in parallel/meridian grid, incl. buffer (pink)*

The table below contains all the original coordinates and the new coordinates with margins included. We have added the margin to all the sides of the landscape. However, as you can see adding the margin to the left and bottom of the area doesn't change anything – even with the added margin the area is still well inside the N49E15 tile.

| | **Latitude** | **Longitude** | **Latitude with mar.** | **Longitude with mar.** |
|---|---|---|---|---|
| **Top left** | 49,9967 | 15,2894 | 50,2467 | 15,0394 |
| **Top right** | 49,9933 | 15,9323 | 50,2433 | 16,1823 |
| **Bottom right** | 49,5789 | 15,9243 | 49,3289 | 16,1743 |
| **Bottom left** | 49,5822 | 15,2869 | 49,3322 | 15,0369 |

Now that we have the coordinates of the area, we can enter them into the Earth Explorer. Switch to decimal degrees in the Coordinates section of the Search Criteria tab and start manually adding the coordinates with margin from the table above. Start with the top left coordinate pair and continue clockwise with the remaining ones.

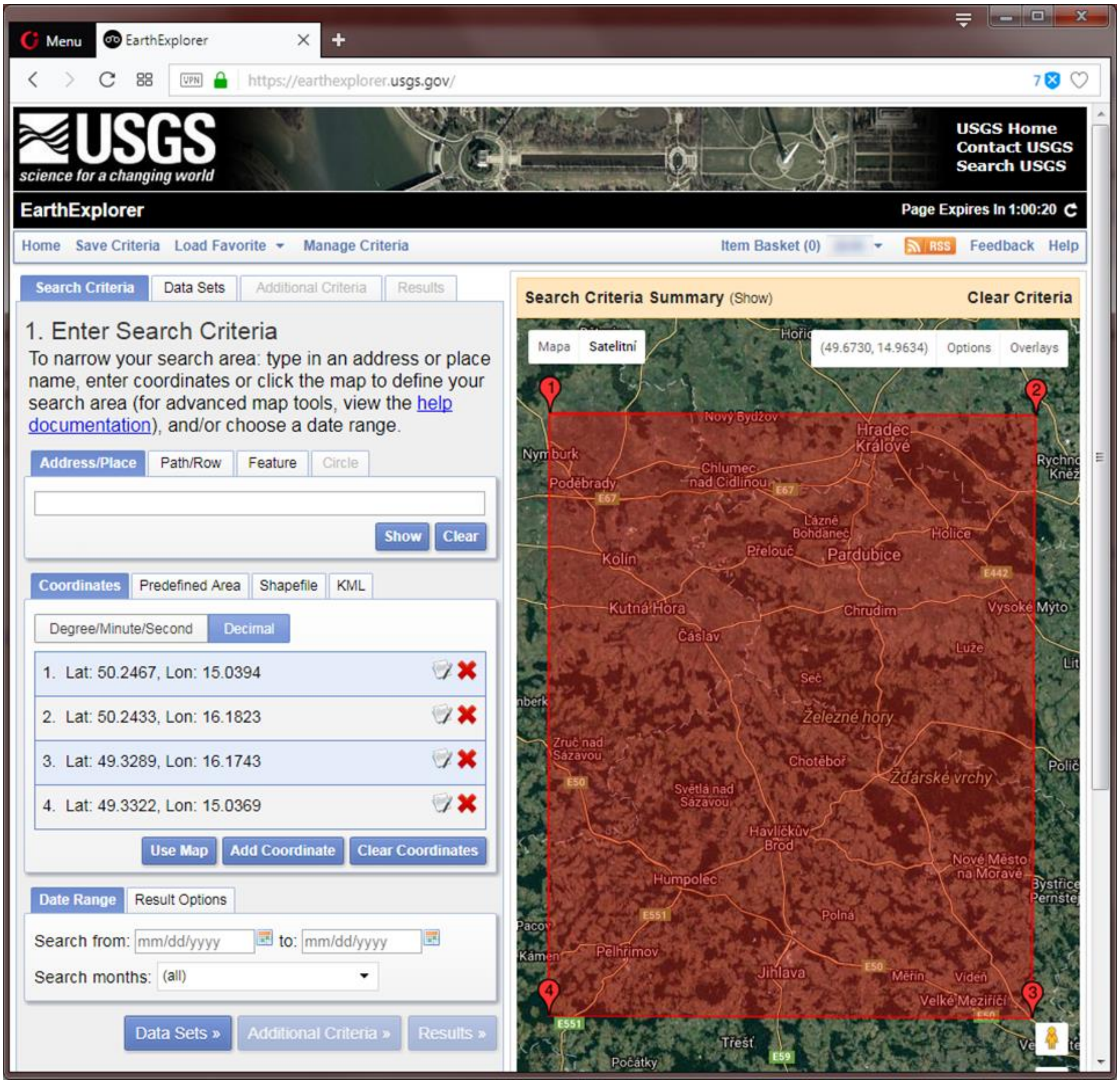



*Figure 5 – EarthExplorer with highlighted search area*

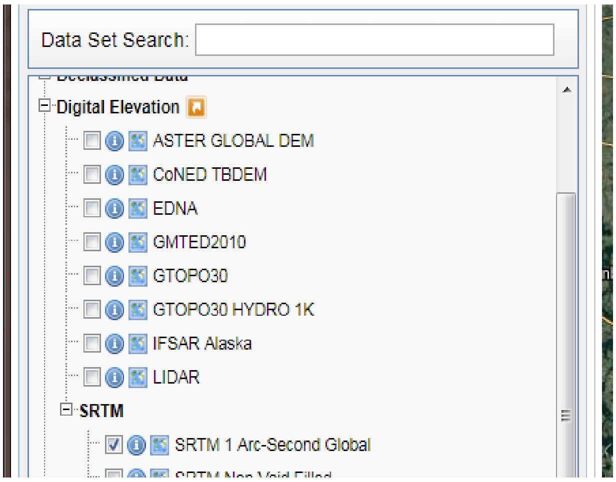
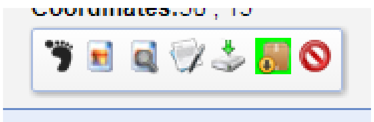



*Figure 6 – Selecting SRTM 1 Arc-Second Global data*

Now that we have outlined the area of interest, we can proceed with getting the data. Press the **Data Sets >>** button, from the list on the following page select **Digital Elevation** > **SRTM** > **SRTM 1 Arc-Second Global** and finally press the **Results >>** button.

As was already mentioned, SRTM data are provided as 1 x 1 degree tiles. As our landscape overlaps 16th meridian east and 50th parallel north, we need 4 tiles in total.

***NOTE: The list of results can show up to 10 pages, each containing 10 tiles, i.e. up to 100 SRTM tiles in total. If your scenery is very large, some of the tiles may be omitted. In such case download the tiles in two or more steps, highlighting only part of the required area at each step. However, keep in mind that such a large scenery would most likely span over multiple UTM zones, making the landscape creation process itself even more complicated.***



To download the data, we need to add all the tiles to the bulk download list. This is done by clicking the second icon from the right. The tile is added in the list if the background of the icon turns green.

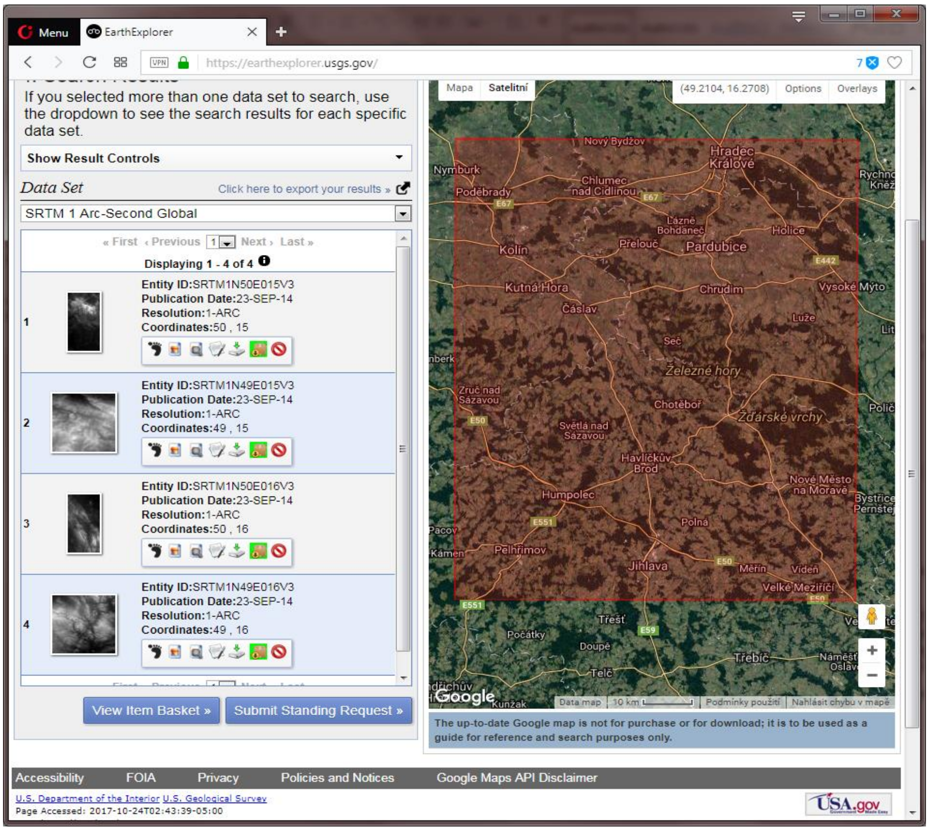



*Figure 7 – Tiles selected for download*

Once we select all the tiles we need, we can click the **View Item Basket >>** button. This takes us to the bulk download list containing the items we have previously selected. There we click the **SRTM 1 Arc-Second Global** list and check if all the files have **BIL 1 Arc-second** selected as the associated product. If so, we can click the **Proceed to Checkout >>** button.

In the Checkout form, check if the number of products matches the required number of tiles. If yes, press the **Submit Order** button. ***Don't worry, although the whole interface seems like an e-shop, you won't be charged as the data you are downloading are provided free of charge.***

After submitting the order, you will be asked to download and install the Bulk Download Application, which will be used to download all the tiles from the Earth Explorer server. You only need to do this one time. Once the BDA is installed, you can use it for any future download from the Earth Explorer.

When you run the BDA, it will ask for your Earth Explorer login information. After successfully logging in, it will show the list of active bulk orders that can be downloaded. If you haven't placed multiple orders, there should only be one active order in the list. Select it and you should see its content.

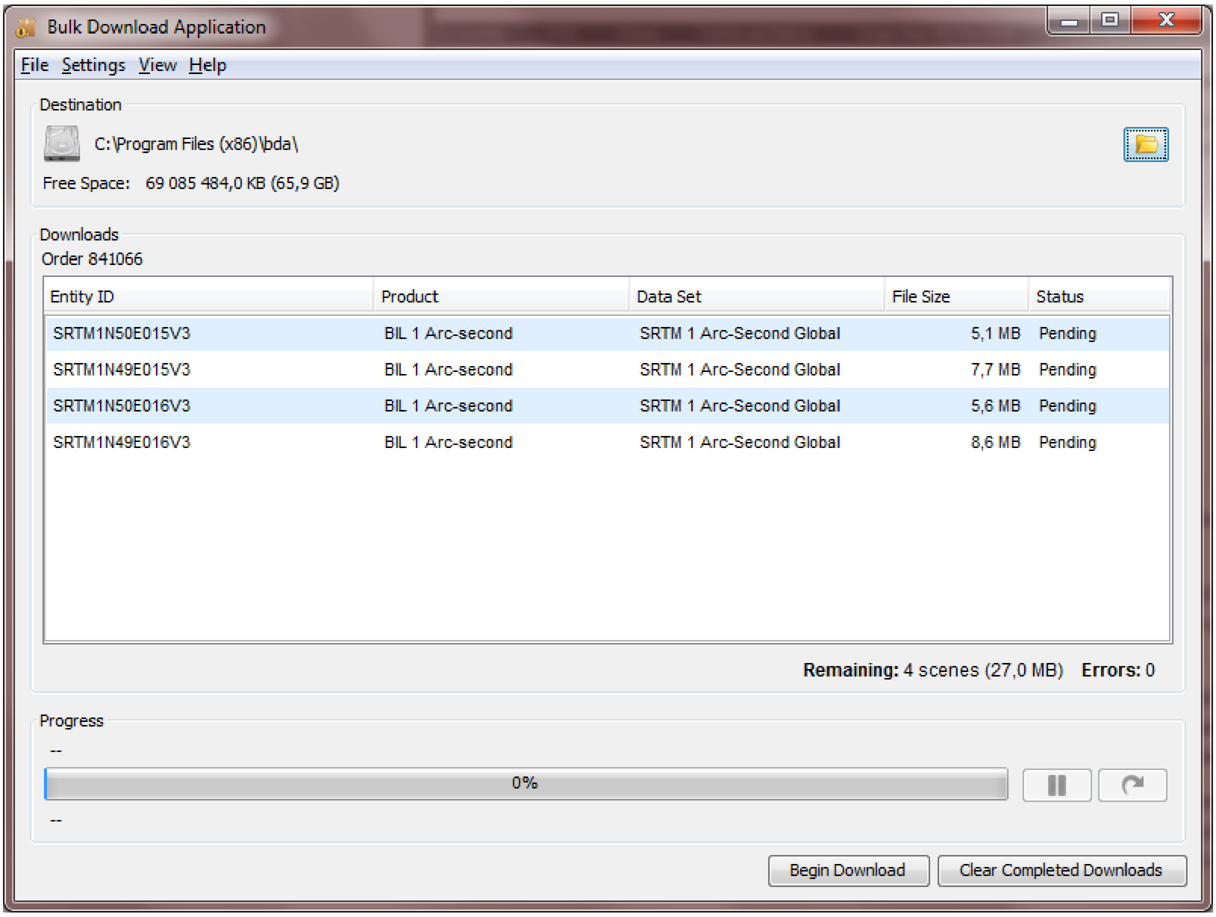



*Figure 8 – Bulk Download Application user interface*

You can change the destination folder by clicking the folder icon at the top right corner of the window. Press the **Begin Download** button to start downloading the files. After the download is completed the BDA can be closed.

We have all the files we need, but to get the final raw terrain data for import into Condor, we must process the downloaded data in a GIS software first.

## Processing elevation data in QGIS

As we are trying to use as many free and open source software products as possible, we will use the QGIS open source geographical information system for our example. If you haven't downloaded and installed it yet, now it's the time to do so. It is available at [http://www.qgis.org](http://www.qgis.org).

***NOTE: QGIS is not the only software usable for this step. As long as you can do similar operations and achieve the same result as using the QGIS, you can use any GIS software you are skilled in working with.***

First, we need to unzip the downloaded tiles into a folder. It is recommended to use the same folder for all the tiles, as it will make selecting them in QGIS easier.

After unzipping the data, we can start QGIS Desktop and begin their processing. The first step is to merge all the separate data tiles into one large elevation model. From the main menu of QGIS select **Raster** > **Miscellaneous > Merge…** to open the Merge function window.

In the Merge window click the **Select…** button next to the Input files edit box. In the file selection dialog navigate to the folder with our unzipped tiles. You will see many files when you open the folder, but only one file type is of our interest. In the file type filter select **ESRI .hdr Labelled (*.bil *.BIL)** to show only the elevation data files. You should now see four files. Select all of them and click **Open**.

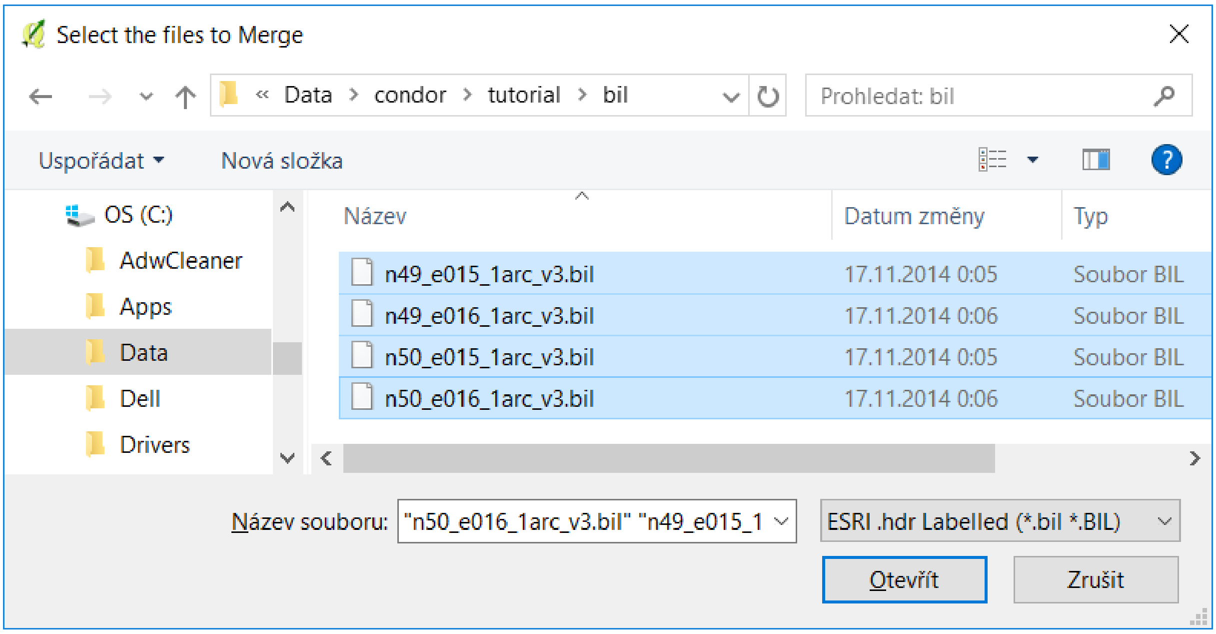



*Figure 9 – Selecting files to merge*

Names of all the selected files should now be visible in the Input files edit box. Now click the second **Select…** button to specify the output file. Set **ESRI .hdr Labelled** as the output file type and enter some name. We will use **tutorial_merged.bil** in our example.

When back in the Merge window, click **OK** to finish the process.

QGIS will work for a moment and will notify you when the process is finished. You can then close the Merge window. You have most likely noticed that our merged terrain is now shown in the QGIS main windows and the list of layers contains one layer with the same name as our output file.

The terrain as it is now is projected in the WGS84 coordinate system, but we need to re-project it to the right UTM zone.

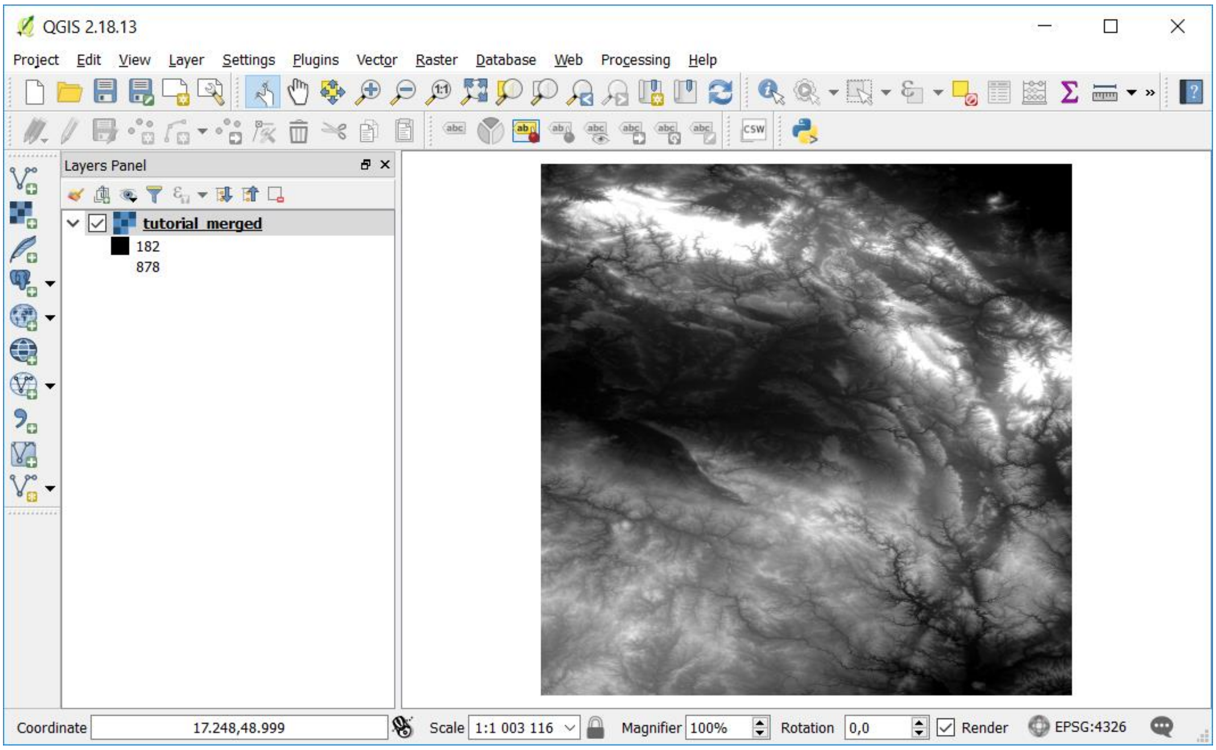



*Figure 10 – QGIS with the loaded terrain*

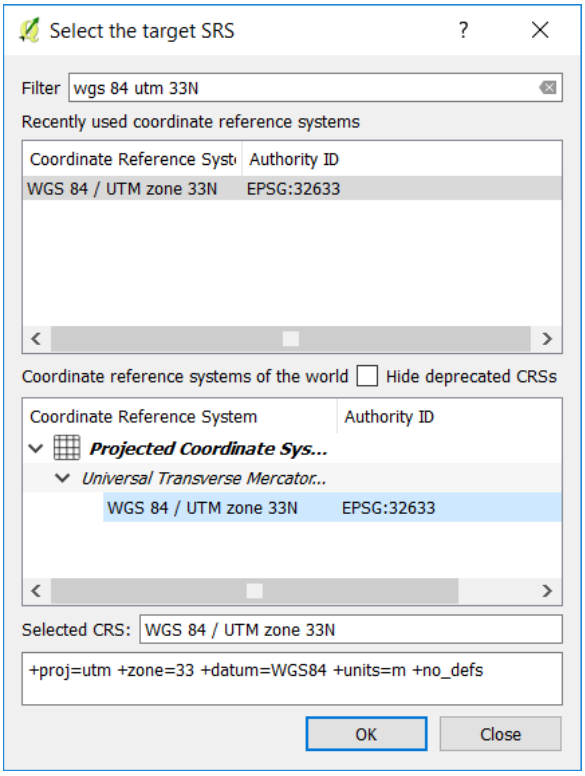



*Figure 11 – Selecting the target SRS*

From the main menu select **Raster** > **Projections** > **Warp (Reproject)…**. When Warp windows appears, notice that our **tutorial_merged.bil** is selected as the input file – we are applying the function to the selected layer.

Click Select… to specify the output file. As in previous steps, use the **ESRI .hdr Labelled** file type and enter the name of the file. We will use **tutorial_warped.bil** in our example.

Check the Target SRS check box and then click the **Select…** next to it to choose the target coordinate system for the re-projection process. We will be using UTM projection with WGS 84 datum. We know from our calculations that our landscape is situated in the UTM zone 33N, so we need to find the **WGS 84 / UTM zone 33N** coordinate reference system in the list. Fortunately, the selection dialog offers a search filter, which can narrow down the list. Enter "wgs 84 utm 33n" as filter value. The list of reference systems should now only contain the coordinate system we need.

Select the needed coordinate system and click **OK** to close the dialog and return to the Warp function window. Check the Resampling method check box and select Cubic spline from the dropdown list.

As the last step, click the pencil icon to edit parameters of the warp function. Add **-tr 30 30** after EHdr. This way we specify the output terrain resolution, which is very important. Otherwise we may end up

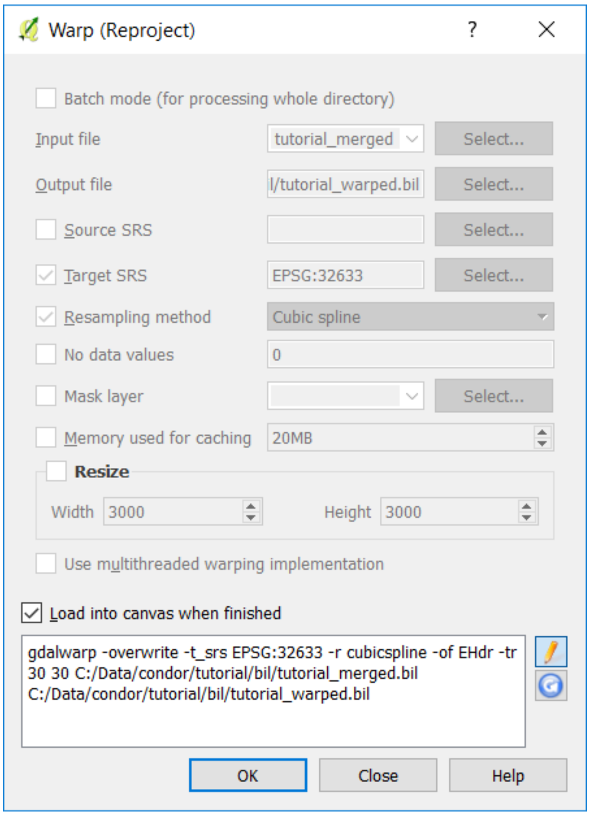



*Figure 12 – Warp function before executing*

with a useless data file several tens or hundreds of gigabytes big. Do not click the pencil icon again, it would remove all the manually added parameters! Click **OK** to start re-projecting.

As you may have noticed, the displayed terrain changed a bit. However, what we see now is the re-projected terrain rendered in the original WGS 84 reference system. Click the coordinate reference system icon in the bottom right corner of the QGIS window and select **WGS 84 / UTM zone 33N** as the coordinate system. As we have used it before, we won't need to search for it again as it will be offered in the list of previously used systems.

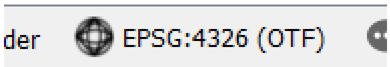



*Figure 13 – Reference system indicator*

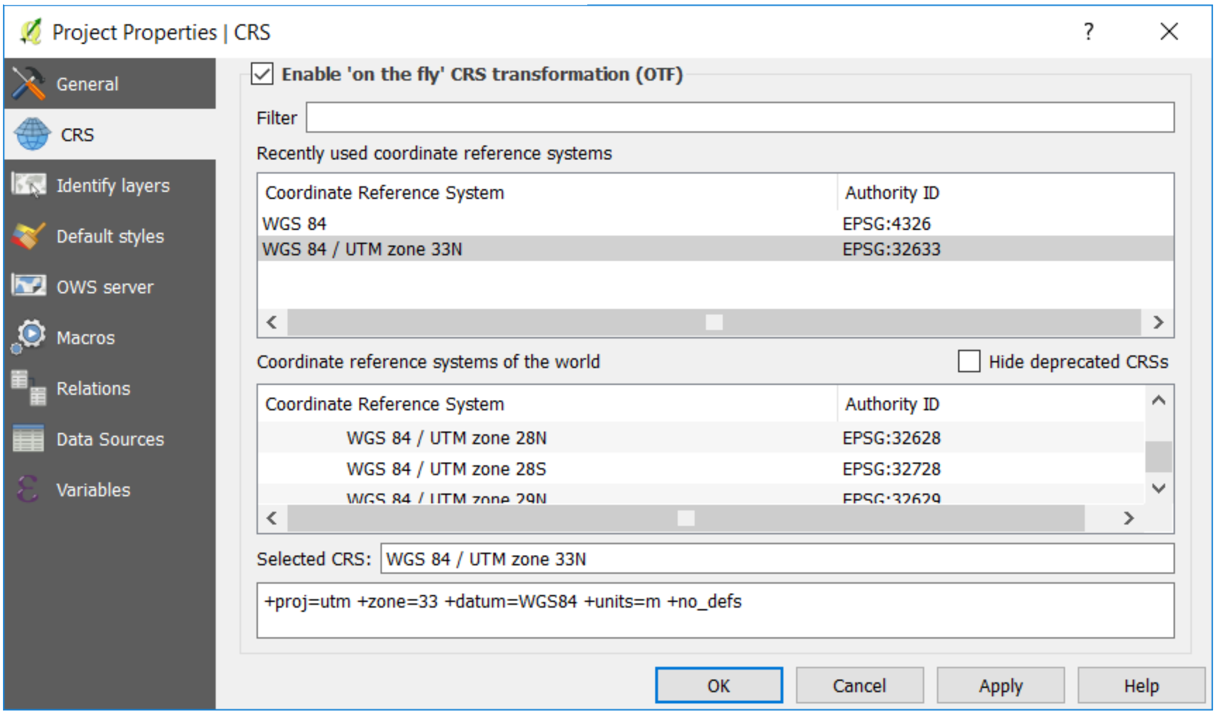



*Figure 14 – Setting the coordinate reference system*

Click **OK** and the terrain proportions now look entirely different.

Now we have the last step ahead of us. To make import of terrain data into Condor easier, we can remove the unneeded parts of the terrain and only save the part needed for our landscape.

From the main QGIS menu select **Raster** > **Extraction** > **Clipper...**. Input file is again automatically set to the last added layer, **tutorial_warped**. Click **Select...** to specify the output file. Select the **ESRI .hdr Labelled** as file type and enter filename. We will use **tutorial_clipped.bil** in our example.

Clipping mode should be set to Extent. For this mode we need to enter easting and northing of top left and bottom right corners. Use the values you noted earlier or if you use UTMtools, you can copy the values directly from the landscape diagram by double clicking on the required value.

When both coordinate pairs are filled in, the area determined by them is highlighted in the terrain window.

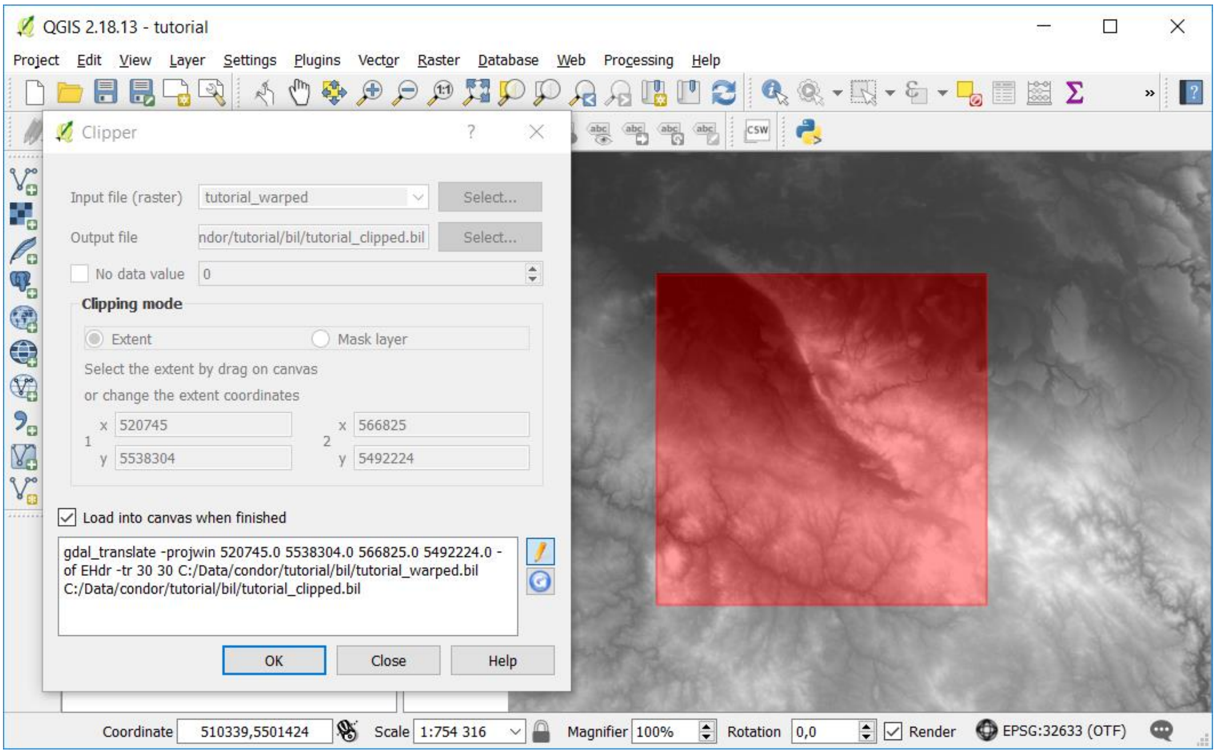



*Figure 15 – Clipper function with highlighted clipped area*

Click the pencil icon and again specify the terrain resolution by adding **-tr 30 30** after EHdr in the function parameters. Click **OK** to proceed with clipping. Once again, QGIS will notify you when the process is finished, and you can then close the Clipper window. The file created using Clipper is now ready to be imported to Condor.

As we won't need QGIS for the next few steps, save the project with our terrain layers and close QGIS.

## Importing terrain elevation data into Condor

In this step we will use the **RawToTrn** application, which is part of the Condor Landscape Toolkit, to process the raw elevation data exported from QGIS and export the terrain in Condor terrain format.

Before we load the data into RawToTrn, we need to find out two parameters required for import. Navigate to the folder we saved our bil files processed with QGIS to. Each bil file has a header hdr file describing the structure of the data stored in it. Find the file **tutorial_clipped.hdr** and open it in any file viewer or editor, e.g. Notepad.

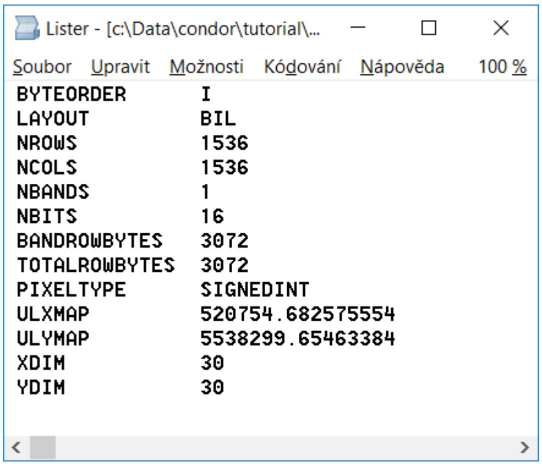


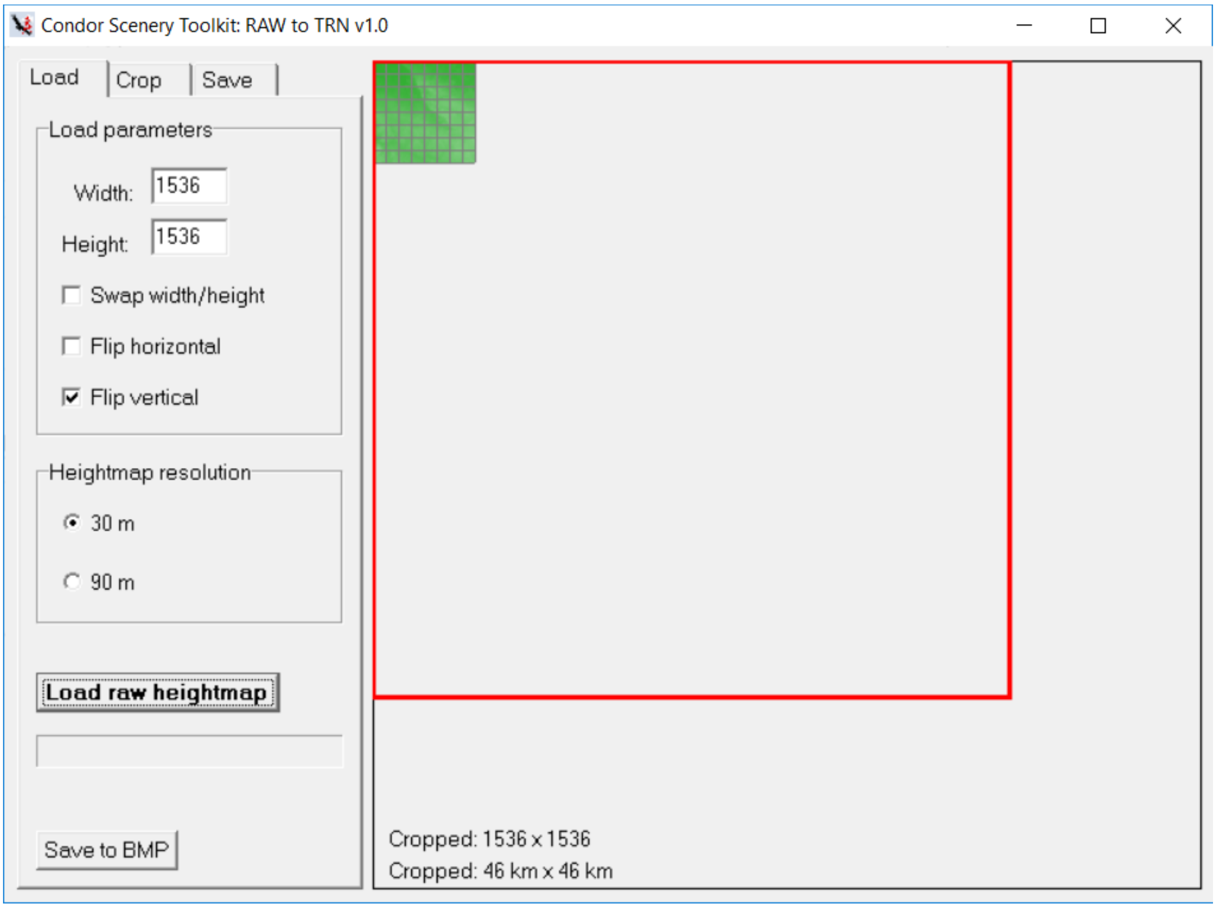

*Figure 16 – HDR file content*

For the raw data to be imported correctly, we need the values of NROWS and NCOLS. In the case of our example both values equal 1536.

NROWS and NCOLS values are actually related to the size of our landscape in tiles. We know that the tile size is 23040 meters and that 1 pixel of the elevation data equals to 30 meters. From this we can infer that we need 768 x 768 pixel grid to store elevation data for one tile (192 x 192 pixels for a patch). The values stored in the hdr file thus correspond to the tile size of our tutorial landscape, which is 2 x 2.

If and only if both NROWS and NCOLS are whole multiples of 192, the terrain data can be directly exported to Condor without any additional cropping in the RawToTrn tool. When working on whole tile level, NROWS and NCOLS must be whole multiples of 768.

Run the RawToTrn tool and enter NCOLS value as width and NROWS value as height. Check Flip vertical. As the resolution of our data is 1 arc-second, make sure that SRTM 30 m option is also checked. Once everything is set up, we can click the **Load raw heightmap** button and select our **tutorial_clipped.bil** data file.




After loading the raw terrain data, check if the scenery size is in multiples of 23 kilometers and pixel size in multiples of 768.

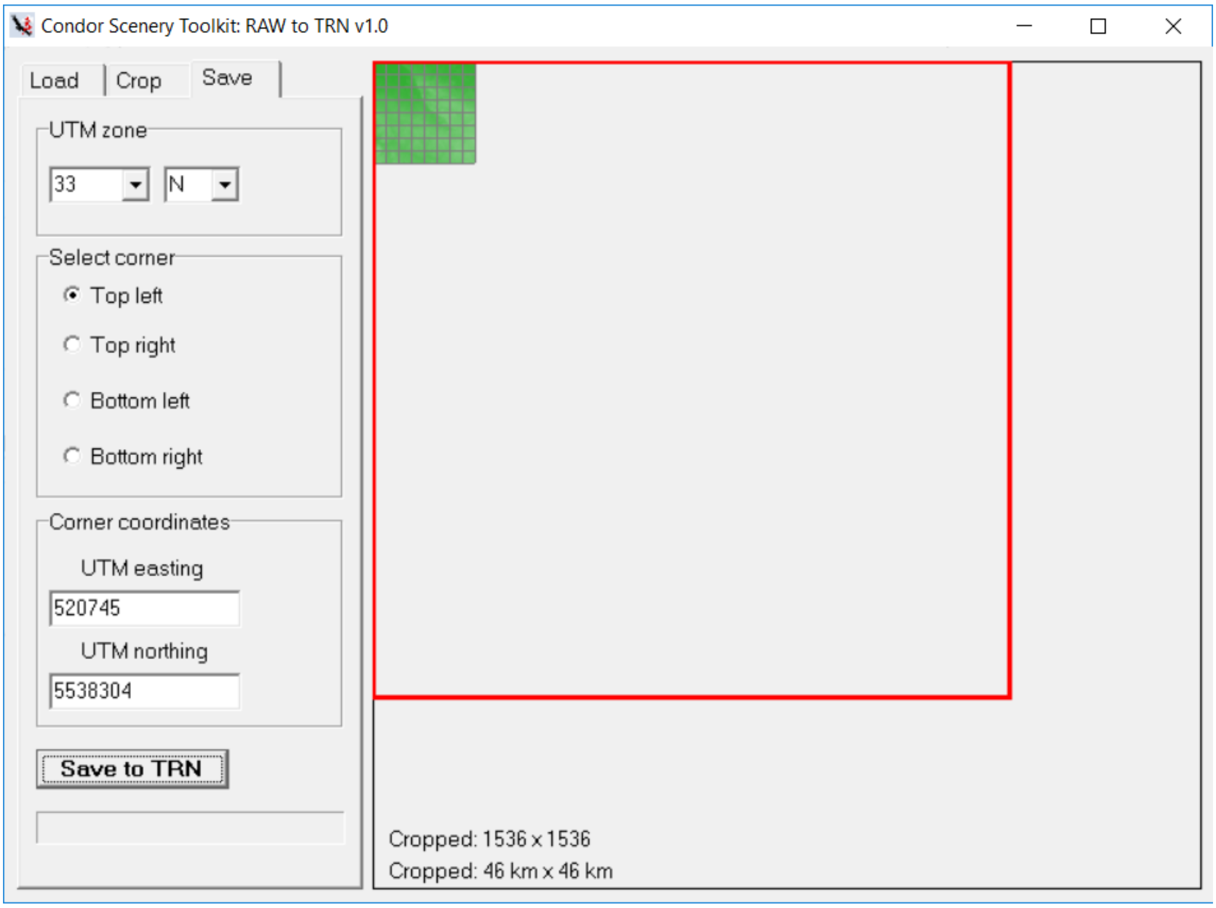
If the size of the terrain is not correct, switch to the Crop tab and adjust the size of the scenery to comply with requirements.

Border crop allows you to remove terrain by the size of scenery patches (5,76 km sized squares).

Exact crop enables precise cropping of the source data on the pixel level, by specifying the top left pixel coordinates and pixel width and height of the terrain area.

***NOTE: Although we can adjust the size of the terrain in the RawToTrn tool too, it is more comfortable to prepare correctly sized terrain data during their processing in QGIS. Not only we avoid cropping the terrain again, but the whole terrain calibration process described later will be easier.***

As we have prepared and cropped the data in QGIS, we don't need to do any size adjustment in the RawToTrn and we are ready to save the terrain in Condor landscape terrain format. However, before we do this, we need to navigate to the **Condor2\Landscapes** folder and create necessary folder structure for our landscape. First, we create **Condor2\Landscapes\Tutorial** folder, which is the root folder for our whole landscape. As the second step we need to create **HeightMaps** folder inside our landscape root folder, i.e. **Condor2\Landscapes\Tutorial\HeightMaps**.




Now we can switch to the Save tab and fill in calibration info. First, we must select the UTM zone where the scenery (or its center, if it spans over multiple zones) is located. Then we select the corner, which will be used for calibration. We will use the top left corner. As the last step, we need to fill in UTM easting and northing coordinates of the selected corner.

Click the **Save to TRN** button to open the terrain file save dialog. Navigate to the root of our example landscape and enter the same filename as is the name of the landscape, i.e. **Tutorial.trn**. Click Save and the Condor terrain is created. Now you can close RawToTrn as it won't be further needed for this tutorial.

We have successfully downloaded and processed raw elevation data using QGIS and converted them to the Condor terrain format using RawToTrn tool. However, our landscape is still incomplete and cannot be tested in Condor yet.

# 5. Building the landscape

In this chapter we discuss all the steps needed to prepare our landscape for the first test flight in the Condor V2.

We have concluded the previous chapter by creating necessary folders for our landscape and exporting the terrain in Condor terrain format. Now we can start editing it in the CLT **Landscape Editor**.

After starting the Landscape Editor, select our Tutorial landscape from the Landscape dropdown list. Our terrain will be loaded and displayed in the editor.

The **View/Modify toolbox** in the left side of the editor contains multiple functions with check boxes. Checked check box of a function means that the data related to the function will be visible in the terrain view. Multiple check boxes can be checked at the same time. To activate a function, click on it to highlight it. Only one function is active at given time.

Since we have calibrated our terrain using RawToTrn, we don't need to do it again in Landscape Editor. If you want to do manual calibration of the terrain, see Appendix 1.

Now it's the right time to save the terrain – from the main menu select **File** and **Save landscape**. Our terrain is now ready, and we are only few simple steps away from testing it in Condor.

To test our landscape, we first need to add at least one airport. At this point the airport will be purely virtual, as there will be no 3D model placed in the landscape. However, airport data defined in the Landscape editor will allow us to start in the air over the airport position.

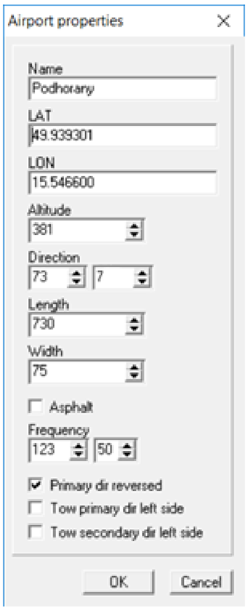



*Figure 17 – Airport data*

Highlight Airports in the toolbox to activate this function and check the check box next to it to show airports in the terrain window. Then right click on the empty airport list below the toolbox and select Add from the context menu.

Fill in all the data as in the picture above and click OK. You should now see Podhořany (in game name "Podhorany") airport in the top middle of the landscape. We will use it for our test. As the airport is located next to a ridge of Iron Mountains, you can even try some ridge soaring during your first test flight. Don't forget that the airport is currently only "virtual". There is no runway and the terrain is not adjusted for tow or winch take-offs. Use airborne start until the airports are added (in chapter 5).

Now we'll do all the remaining four steps needed for testing the landscape in Condor. In short, we need to execute all four export functions in the File menu. We will discuss all of them in detail in the following chapter.

First select **File** > **Export flightplanner map**. This will generate a map of our landscape that will be used in the Condor flight planner.

Then select **File** > **Export forest map**. We haven't created any forest map tiles yet, so we won't have any trees in our landscape, but the exported forest map is needed, even though it is empty now. Click Yes and then No to export the forest map.

Select **File** > **Export thermal map**. This will export the thermal map which tells the Condor weather engine in which areas of the landscape is higher probability of thermals. Again, we haven't prepared any thermal map yet, so the exported thermal map won't influence thermal generation.

Then select **File** > **Export textures to DDS**. For this function to work, we need to have nvdxt.exe from Nvidia legacy texture tools placed in the same folder as LandscapeEditor.exe. Click Yes to confirm that the process may take some time and then No to create all the textures. Our landscape will be covered with grey textures generated from the terrain features. Don't worry, we will replace them in the next chapter.

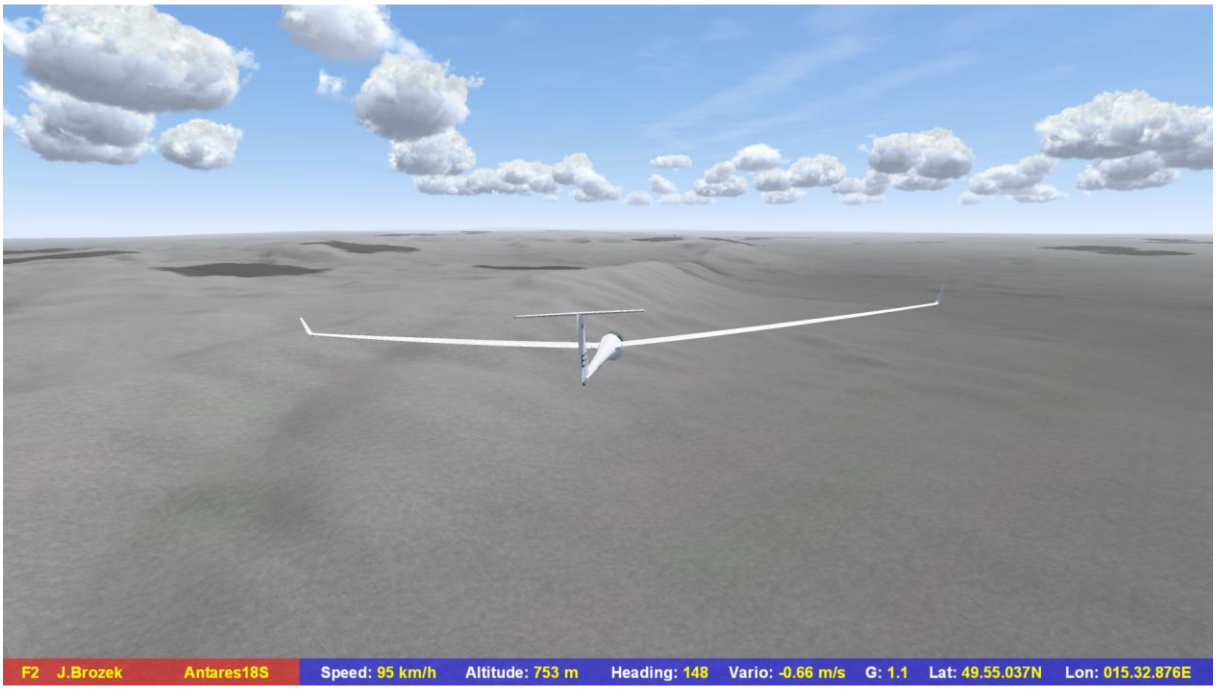
Finally, you must build the hash files for terrain and forest patches. Those files (landscapename.THA and landscapename.FHA) are very important for anti-cheat protection during multiplayer flights. Also, if they are not present, your scenery will not be usable in Condor. Select **File > Export terrain hash** and then **File > Export forest hash**.

**Generating hashes every time the terrain or forest maps are changed is very important, as Condor will report landscape manipulation both in single player free flight mode and in competition multiplayer, if the patch hash is different from the one in the hashtable.**

**New anti-cheat features were introduced in V2, related to the changes in the V2 landscape architecture. All the terrain/forest map patches are now checked against the hash table when loaded in game and if a difference is found, it is considered a cheating attempt.**
**In multiplayer flights, the hashtables are compared to the hashtable at the hosting server first, to make sure all the competing pilots have the same terrain and forest data.**

**If needed, terrain and forest hashtables can be created from command line by calling *LandscapeEditor.exe -hash LandscapeName*.**
**For example: *LandscapeEditor.exe -hash Tutorial***

Now you can save your landscape again using **File** > **Save landscape**, close the Landscape Editor and start Condor to enjoy the first flight on your own landscape. Congratulations! 😊



## Landscape settings file – LandscapeName.ini

Landscape root folder may contain .ini file with some additional information. Currently, there are two settings that can be specified there.

The .ini file contains section called **General**, which includes the following settings:

- **Version** – version number of the landscape. It is shown both in the flight planner and in the server list at the Condor website. Version number allows easier identification of the landscape needed for the flight.
- **RealtimeShading** – special parameter that can be used to turn off the real-time terrain shading of photo texture landscapes. This parameter is optional, shading is on by default. To turn the shading off, use **RealtimeShading=0**.

Example content of the landscape .ini file is below. The landscape in this case is in version 0.9 and the terrain will be shaded in real-time.

```
[General]
Version=0.9
RealtimeShading=1
```

# 6. Adding custom textures to our landscape

The previous chapter concluded with the first test flight on our tutorial landscape. However, the landscape is covered with uniform grey texture and is void of any details.

In this chapter, we will discuss adding custom textures to our landscape. We will also describe how to add waterbodies and automatically generated trees.

All the "source" files used during the development of our landscape are stored inside the Working folder our landscape, i.e. Landscapes\Tutorial\Working. We will be referencing this folder simply as "Working folder" further in the text. This folder is only needed during development, it is not needed for the final landscape to work and should be removed for distribution, unless you want to give everyone access to all your source files.

As mentioned many times previously, for processing in Landscape Editor the Condor V2 landscapes are divided to 23.04 x 23.04 km square tiles. Each tile has exactly one corresponding tile texture file. Filename of the file is the same as the coordinates of the tile, i.e. texture file 0503.bmp belongs to the terrain tile 0503. Textures are stored in the Windows Bitmap format. All the tile textures are stored in the automatically created Terragen\Textures subfolder located inside our Working folder, i.e. Landscapes\Tutorial\Working\Terragen\Textures. The standard tile texture resolution in Condor V2 is 8192 x 8192 pixels, giving the texture resolution of approx. 2.8 meters per pixel.

***NOTE: Be aware that one tile texture in the 8192 resolution saved in the .bmp format requires almost 200 megabytes of hard drive space. The space required for source textures and maps for a larger scenery can easily be in the rate of gigabyes.***

There are two main approaches to creating scenery textures:

1. "Synthetic" artwork textures – all the textures are created manually, either fully or partially. We will use this approach in our tutorial, albeit in a very simple way as it is very easy to spend tens of hours working on a texture for a single tile. The prime example of the use of artwork textures is the default Slovenia V2 landscape or any of the CondorWorld landscapes – all created by the skilled landscape maker Miloš Koch.

2. "Photorealistic" textures – the whole landscape is covered by textures made from aerial/satellite imagery of the particular area. The biggest advantage of this approach is that the landscape looks exactly as the real landscape at the time of taking the photos. However, there are multiple distinct disadvantages. Such photos are usually taken during a longer period, resulting in lighting and color mismatch. Also, all the terrain features and objects cast shadows that are recorded in the pictures. As Condor V2 brings realistic lighting and shading of the terrain and objects, any pre-shading of the terrain is not needed and is counterproductive. It is possible to disable terrain shading for photo sceneries, but this way you sacrifice one of the new features of V2. Another matter that must be taken into account is that aerial/satellite imagery is usually licensed, and its reuse is strictly limited, if it is allowed at all. E.g. Google Maps have very strict license terms forbidding reuse, redistribution and even bulk downloading of their data.

## Creating textures for our landscape

It is important to point out that creating the textures is not the main aim of our landscape tutorial, but the landscape without textures would be incomplete.

If you have already created textures for V1, you can just skip to the sub-chapter "Adding water", as water bodies are new feature in V2.

The process described here was designed just for this tutorial with the aim to be as fast and automated as possible. Of course, it can be used for real sceneries but without additional steps and manual alterations the result will be visually inferior to the manually created textures e.g. in the default Slovenia V2 landscape.

For this part of the process we will need a bitmap drawing software (we will use Paint.NET, which is free, but you can use any bitmap drawing program that can work with layers – e.g. open source GIMP or paid Adobe Photoshop). We will also use QGIS with QuickOSM plug-in and WaterAlpha tool from the Condor Landscape Toolkit.

First, we need to find some basic textures that we will use to build our tile textures. In general, we need a seamless background "farmland" texture with fields but without any villages, cities, waterbodies or large forests. Then we also need a seamless forest texture and a seamless city texture. For your own landscapes you can use any texture with a license allowing its reuse and further distribution as part of another work. For our tutorial we will use the "How in the World" free terrain textures created by Jenna Fearon / "How in the World?" and licensed under a Creative Commons Attribution 4.0 International License. You can download the texture pack at [http://howintheworld.com/textures/](http://howintheworld.com/textures/).

We will use the following textures from the HITW pack:

HITW-TS2-French-farm-1.jpg
background texture

HITW-TS2-forest-dark-grass-green.jpg
forest texture

HITW-TS2-Austr-city-farm1.jpg
city and village texture

HITW-TS2-grass-green.jpg
airport background texture

All the forests, residential areas, waterbodies and roads will be created using data from the OpenStreetMap project. We will use QGIS to retrieve them.

Launch QGIS Desktop and in the main menu select **Plugins > Manage and Install Plugins…**. Let the list of available plugins load and once its open, type "quickosm" into the Search field to find the plugin we need and then install it. QuickOSM will be added to the Vector menu.

Open the project with our terrain data we have saved earlier. We will need the **tutorial_clipped** layer to set extent of the OSM data we need to download.

***NOTE: OSM data need quite a lot of hard drive space. Be careful especially if your landscape is quite large. Dividing the landscape and downloading the data for smaller area may help, as you may process each part separately and then delete the unnecessary data. 2x2 tile area (the size of our tutorial landscape) is a reasonable compromise between processed landscape area and size of the necessary data.***

OSM data are tagged with attribute keys describing their nature and purpose. We will download the following data.

| | |
|---|---|
| **landuse=forest** | forest data |
| **landuse=farmland** | farm field data |
| **landuse=residential** | cities, towns and villages |
| **landuse=reservoir** | ponds, polders, lakes, dams |
| **waterway=*** | streams and rivers |
| **aeroway=aerodrome** | airport areas |

As the OSM data can be added by anyone, there may be occasional problem with what you see in OSM map and what you download and visualize in QGIS, as the data may have wrong or non-standard attributes. It is recommended to compare the result in QGIS with rendered map at http://www.openstreetmap.org.

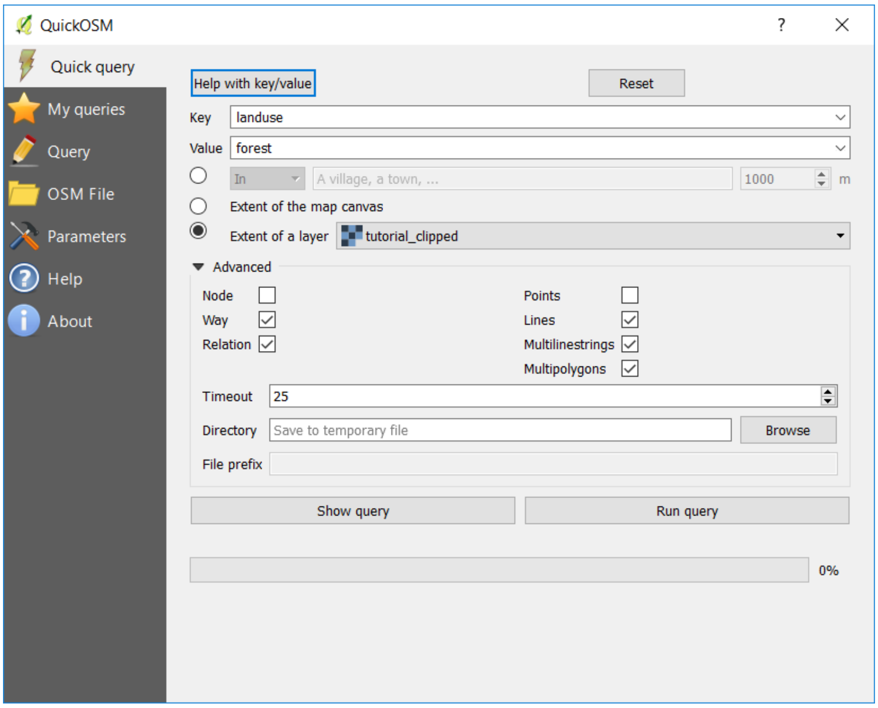



*Figure 18 – QuickOSM interface*

Hide all the raster layers except **tutorial_clipped**. Now let's start downloading OSM data. We will start with forests. In the main menu select **Vector > QuickOSM > QuickOSM** to open the plugin.

First, we need to specify the key/value of the data we want to download. We will start with the first row of the above table. Set "landuse" as key and "forest" as value. QuickOSM contains the most used keys and their related values which can be picked up from the combo box, but we can also type in any other key/value pair that is not listed.

Now we need to specify the extent of the data. Select **Extent of a layer** radio button and pick **tutorial_clipped** from the list. This way we make sure that only the map elements belonging to our area will be downloaded.

Now click on **Advanced** to show advanced options and deselect Nodes and Points check boxes. Keep all other options selected. Now we can click the **Run query** button.

Once the progress bar reaches 100% and “Successful query!” status info appears, we can either close QuickOSM or just switch to the main QGIS window and leave the plugin open.

In the main window we can see that the forest polygons now overlay the terrain. Their color is random, and each polygon has black outline. Also, there is a new vector layer called **landuse_forest** in our layer list containing all the forest polygons. Right click it and select **Properties**. We will now get rid of the black outline and set the color of polygons to green.

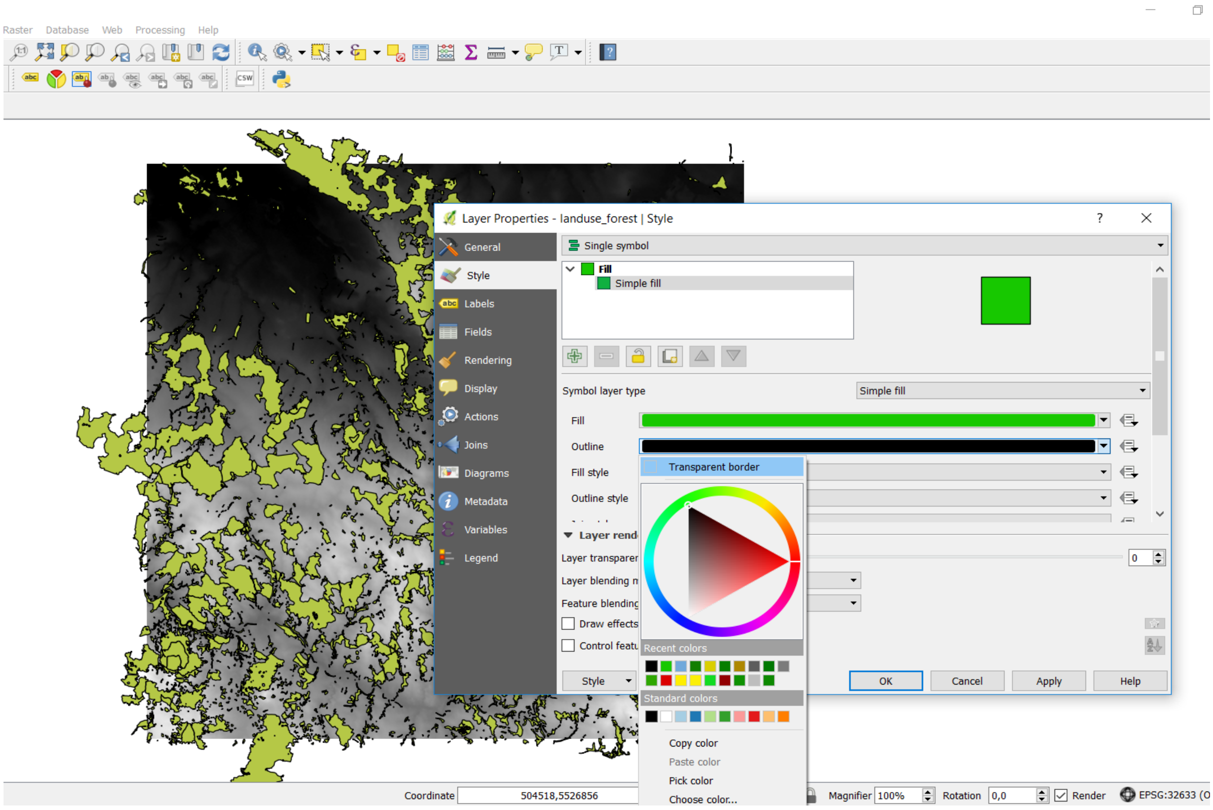



*Figure 19 – Setting layer properties*

In the Layer Properties window click on **Style** tab to show layer style. Then click on the **Simple fill** style to show its attributes. Once they show up, click on the arrow of the **Outline** color to show color selection gizmo. Click on Transparent border at the top. This will remove the black outline. Now click on the Fill color bar and set the color to some shade of green. Click **Apply** and all the polygons now should be green and without the black outline. If by chance there is also a line vector layer of the same name, you can delete it.

Let’s continue with fields. Open QuickOSM again, keep the same settings as before, but fill in farmland instead of forest. We should now have another new vector layer, called landuse_farmland. Remove its black outline and set its color to some shade of yellow or orange. If there is a line vector layer of the same name, delete it. Unlike the other data layers, fields won’t be used for generating textures but only for creating the thermal map for our landscape.

Now let’s download residential areas. Open QuickOSM and this time use residential as a value for the landuse key. Run the query and another layer should appear, called landuse_residential. Again, delete the line vector layer of the same name if it exists. Remove outlines of the polygons and set their color to red or similar.

The last landuse value we will download is reservoir. After downloading the polygons, remove their outline and set the color to some shade of blue (note the RGB values or html color value). This will be the first of multiple layers containing water information.

Open QuickOSM and instead of landuse use the waterway key. Do not enter anything into value. We want to download all the types of water related polygons and lines. Once the query is done, we should see larger river bodies as polygons and small streams and rivers as blue lines. We should also have two layers called waterway in the layer list – a polygon layer and a line layer. Remove the outline of polygons and set their color to the same one we used for landuse_reservoir. Then switch to the line layer and use the same color for the lines. Set pen width to approx. 1.25 pixels.

Unfortunately, as OSM data can be edited by anyone, some polygons of the same type can have different attributes than others. This is also the case of the Seč dam, the largest water body in our landscape, and many other ponds and reservoirs, that is still missing from our data. To load them, we must query the OSM again, this time with water key and no value. Finally, we have most of the water bodies of our landscape loaded and displayed in QGIS as water layer. Edit its properties to remove the outline and set the color to the same shade of blue we used for all the other water related layers.

As airports in V2 do not contain any background polygons, we need to prepare airport backgrounds in our textures. We will use aeroway key with aerodrome value. We will most likely get not just a polygon layer but also a line layer. Delete the line layer and remove black outline of polygons in polygon layer properties. Also, set the color of this layer to brown or similar (use shade that is distinctly different from the one used for residential areas).

The last layer we will download will contain roads. In QuickOSM use the key highway with no value set. We should get two layers as in the case of waterways. Set color for both to the same shade of grey, remove black outline of polygons and set pen width of the line layer to 2 pixels.

We should now have all the data we need to make our textures. We just need to export them from QGIS in some format editable in Paint.NET. We will use **Print Composer** for that.

Hide the tutorial_clipped raster layer and press Ctrl+P to open new Print Composer. Click on **Composition** and set page size preset to **Custom** and set the size to 200 x 200 millimeters. In the left toolbar click **Add new map** and place it on the canvas by clicking and dragging. Don't worry that the content looks strange, we will sort this out later.

Keep the map selected and click **Item properties**. Scroll down to **Extents** and enter left bottom and right top UTM easting and northing pairs (520745, 5492224 and 566825, 5538304).

Now scroll down a bit again and click **Position and Size** section to open it. Set X and Y to 0 mm and Width and Height to 200 mm. The map should now fill the whole canvas.

Now we just need to export it as image. Click Save as Image from the main toolbar. Set PNG as the file type and enter some name. Let's call the file masks.png. In the following dialog set Page width and height to 16384 pixels. As our scenery is 2 x 2 tiles, this is all we need. Save the image. Don't panic, it will take some time.

*NOTE: Be careful when exporting the masks for larger area. Although the saved PNG file from our example is relatively small, it will take up 1 gigabyte when loaded into the memory!*

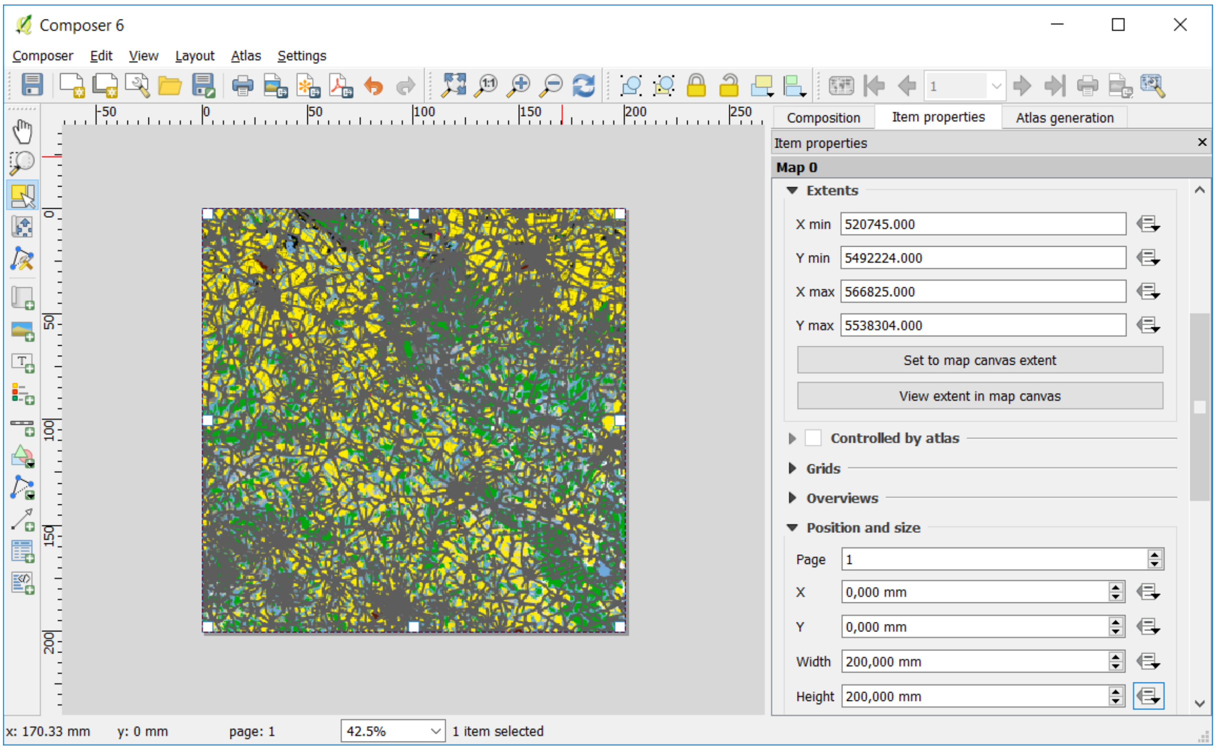



*Figure 20 – Print Composer ready for export*

We are now done. Save the project to keep the data and close QGIS.

Start Paint.NET and open the farmland texture from the HITW pack. Its size is 1024 x 1024 pixels. Because our tile texture is 8192 x 8192 pixels, we need to fill this area with the farmland texture.

Select the whole texture and copy it to the clipboard. Press **Ctrl+Shift+R** to open **Canvas Size** dialog. Set anchor to bottom right and canvas width to 2048 pixels. Keep height unchanged and press **OK**. Now press **Ctrl+V** to paste the tile from the clipboard to the extended canvas. It should now be filled. Press **Ctrl+A** to select the whole image, **Ctrl+C** to copy it to the clipboard and repeat the canvas extension, this time with 4096 as the width. Paste the content of the clipboard to the extended image and repeat the same process with width 8192 pixels.

Once you have the texture tiled in 8192 x 1024 pixels image, select it whole and copy it to the clipboard. Again, expand the canvas, but this time change the height to 2048. Paste the single row image from clipboard. Now select the whole image again and copy it to the clipboard. Extend the height to 4096, paste from the clipboard. Now, for the last time, select the whole image, copy it to the clipboard and expand the canvas height to 8192 pixels. Now paste the image from clipboard and save the final image.

Now do the same with the city texture, forest texture and airport grass texture. As the grass texture from HITW pack is quite dark, lighten it a little bit (or use a different seamless grass/meadow texture).

Once this is done, we need to do exactly the opposite with the mask image we exported from QGIS – we need to cut the tiles out of it. We will use the same function.

Open masks.png and in Canvas Size set width and height to 8192 pixels and anchor to bottom right. Save this tile as 0000.png. Undo the canvas change to get the original image back and repeat the same with anchor to top right (0001.png), bottom left (0100.png) and top left (0101.png).

Now we can prepare the final textures for our tiles. Load the tiled farmland background texture. Add a new layer, open the tiled forest texture, copy it to the clipboard and paste it into the new layer. Do the same with the tiled city texture and airport background texture. We should now have a file with four layers.

Next add a new layer and fill it with blue-grey color. This color will be used for water surfaces in our textures.

Add another layer and fill it with grey color. This will be the color of roads.

Finally, add a new layer and leave it empty. Save the file as template.pdn. This will be the template for all our tile textures.

## Compositing the tile textures

Let's demonstrate the compositing process on the example of the tile 0000. Load our template.pdn and save it as 0000.pdn. Now copy the mask 0000.png into the top-most empty layer of the template. We will now create the final texture just by using the **Magic Wand** tool.

Activate the Magic Wand in the toolbox, set the tolerance to 25% and flood mode to global. If the top-most layer with the mask image is not the active one, click on it in the layer panel to select it. Now click to any forest (green) area in the mask image. Magic Wand should select all forest in the mask.

Now switch to the layer containing our seamless forest texture. The selection made using the Magic Wand is now active on this layer. Press **Ctrl+I** to invert the selection – this will select all areas that are not forests. Press Delete to remove the selected parts of the texture, leaving the forests only.

Now switch back to the layer with the mask image, use the Magic Wand to select all residential areas and switch to the layer with the seamless city texture. Again, invert the selection and delete unwanted parts of the texture.

Now repeat the same process on the water areas (use the blue-grey layer) and airport backgrounds (use the grass texture layer).

The last step is to add roads. Use Zoom tool to zoom in on any of the roads, then switch to Magic Wand and click on it to select all the roads in the mask. Invert the selection and switch to the all grey layer. Press Delete to remove unneeded parts of the road texture.

Now hide the mask image layer and check the final look of the texture. Save the file in PDN format, so you can return to it to if there's a need for additional adjustments. We will also need it very soon for creating forest maps and water alpha channel for our texture. Then save the file again, this time in BMP format as 0000.bmp. This will be the final texture.

Now we must repeat the same process for the remaining three tile textures. Once we are done, we can move our four BMPs to the **\Terragen\Textures** folder in our Working folder.

Let's try to apply our textures on the terrain. Run the Landscape Editor and open our Tutorial landscape. From the main menu select **Tools > Import Terragen size textures**. Our textures will be loaded and displayed in the editor. We can also re-export the bitmap shown in the Condor flight planner window (**File > Export flightplanner map**).

If you want, you can now export the textures to DDS and try the landscape with textures in Condor. We can use this function for now, however, since we will be adding water information to our textures, this won't be the way to export the final DDS textures to Condor.

## Adding water

Unlike V1, where every water surface was just a spot on a texture, Condor V2 introduces water surfaces with shader generated water effects. To achieve this, we need a way to tell Condor that a given part of the texture is supposed to be rendered as a water with all the effects. For this purpose, every texture can have an alpha channel containing water information.

**Never use the *Export textures to DDS* function in the Landscape Editor for processing textures containing water alpha channel. Landscape Editor uses DXT1 compression by default, which produces smaller files, but only uses 1bit alpha channel, which is not suitable for realistic waterbody representation (borders of lakes and rivers would be jagged).**

Unfortunately, adding the alpha channel to a bitmap is not a trivial process in most of the graphic software, as the alpha channel is mostly used for encoding transparency in the image. Graphic programs usually automatically calculate the alpha value and replace the original pixel color information with calculated values. This causes inadvertent effects in Condor.

We will use the **WaterAlpha** tool to add the water information to our original textures. This tool was developed specifically for adding water alpha channel to textures saved as Windows Bitmap image. It also handles automatic cutting of processed textures to patches and their conversion to DDS format.

However, to use it, we need to create alpha channel bitmaps first. We only need two colors for encoding the water information – white for the regular surface and black for water. We can easily create such bitmaps from our PDN source files.

Let's start again with the tile 0000. Open 0000.pdn in Paint.NET and hide all the layers except the one containing water texture. From the main menu select Adjustments > Auto-Level. This will turn the blue-grey color to black. Save the file as a000.bmp. Do not save any changes to the original 0000.pdn. Repeat the same process for the remaining three textures.

Once you are done, move all four files to the **\Terragen\Textures** folder in the Working folder.

Now run the WaterAlpha tool. Select our **Working\Terragen\Textures** folder and set tile dimensions to 8192 pixels. Click **Process Tiles**. WaterAlpha will start processing the textures. It first cuts both textures and their alpha images to patches. Then goes through all the patches and if an alpha patch contains black pixels, it is added to the related texture patch. If the alpha patch is all white (it doesn't contain any water information) it is discarded. Finally, all the patches are converted to DDS format with DXT compression using **nvDXT.exe** and stored in the **\Terragen\Textures\dds** folder. We can then copy or move them to the landscape's Texture folder, i.e. **Landscapes\Tutorial\Textures**, replacing the old texture files.

Unless we decide to do some changes in the design or correct some error, those will be our final terrain textures.

## Adding trees

Just as V1, Condor V2 also uses forest maps to specify areas covered with automatically generated tree objects. V2, however, increases size of the forest map tile to 2048 x 2048 pixels. This allows more precise placement of tree areas in the terrain.

In contrast to water alpha images, black in forest maps identifies areas with no tree coverage. Any other color indicates presence of trees.

Each landscape has two forest maps – one for coniferous and one for deciduous trees. Filenames of the forest map tiles for coniferous trees start with letter s, e.g. s0100.bmp. Filenames of tiles for deciduous trees start with letter b, e.g. b0100.bmp. All the forest map tiles are placed in the **\Terragen\ForestMaps** folder in our Working folder.

When making your own scenery, get inspiration from the real forests in the particular area. Use photos, satellite or aerial images to find out how the forests really look like. Many of the natural forests are in fact mixed and only cultivated forests are strict monocultures of a certain tree species.

In our example, however, we will try to keep the things as simple as possible. We will use coniferous forest map for all the forest areas we have added to our textures and deciduous forest map for solitary trees and small tree groups present in our background farmland texture.

### *Creating coniferous forest map from the tile mask image*

We will create coniferous forest maps for our tiles using the mask images we have created earlier. Open one of the mask files in Paint.NET, for example we will use 0000.png. Add new layer and flood fill it with black color. Then move the layer below the current background layer. Select the **Magic Wand** tool and set flood mode to global (if it already isn't). Now click on any forest area to select all the forest on the mask. Press **Ctrl+I** to invert the selection and then Delete to remove everything else than our forests. Now pres **Ctrl+R** to resize the tile to 2048 x 2048 pixels and then save the resized image as Windows Bitmap – BMP, naming it s0000.bmp.

Repeat the same process for files 0001.png, 0100.png and 0101.png, saving the processed forest map images as s0001.bmp, s0100.bmp and s0101.bmp.

### *Creating deciduous forest map from the farmland background texture*

The process of creating the deciduous forest maps for our landscape will involve more work. We will use **Magic Wand** again, but this time it will be about finding the right setting that would select as many tree areas in the texture as possible without selecting parts of the image that do not contain trees.

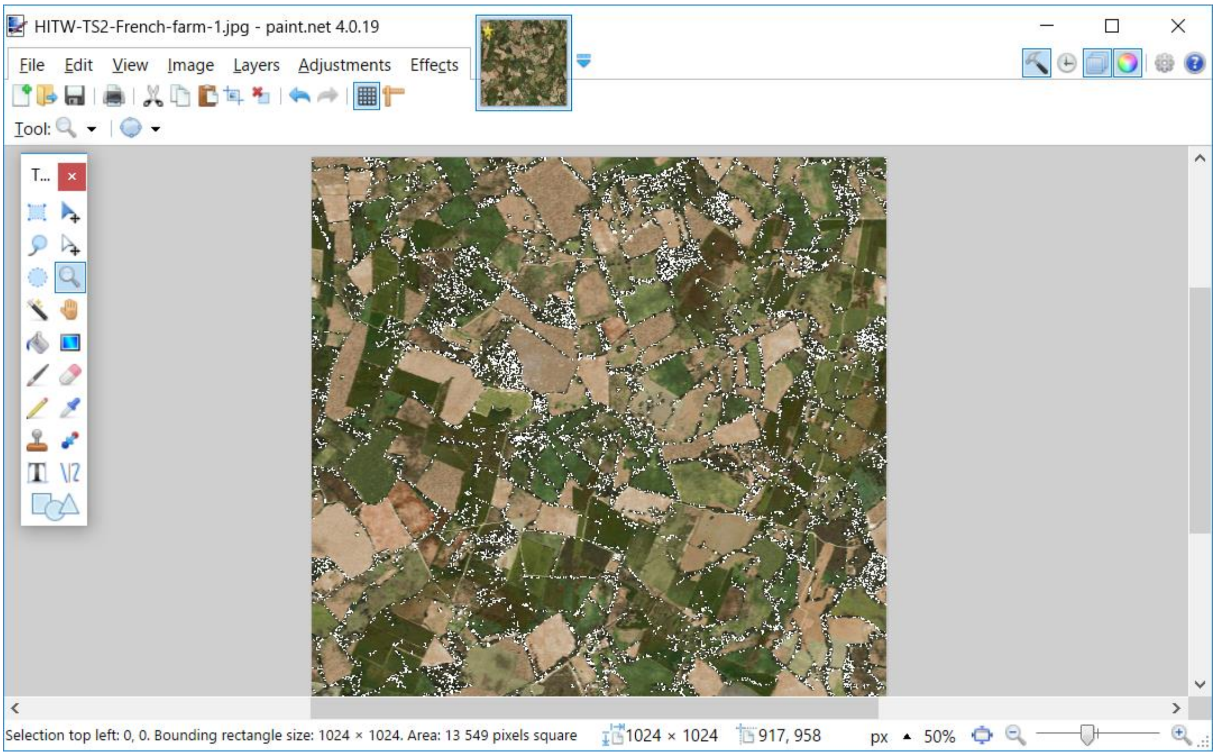



*Figure 21 – Selected deciduous trees in the texture*

We will start by opening the original texture tile from the HITW pack HITW-TS2-French-farm-1.jpg in Paint.NET. Pick the Magic Wand tool, set its tolerance to 12% and flood mode to global. Then click inside one of the tree areas in the texture. Check if the selection only contains tree areas and if not, adjust the tolerance or try to click on a different spot.

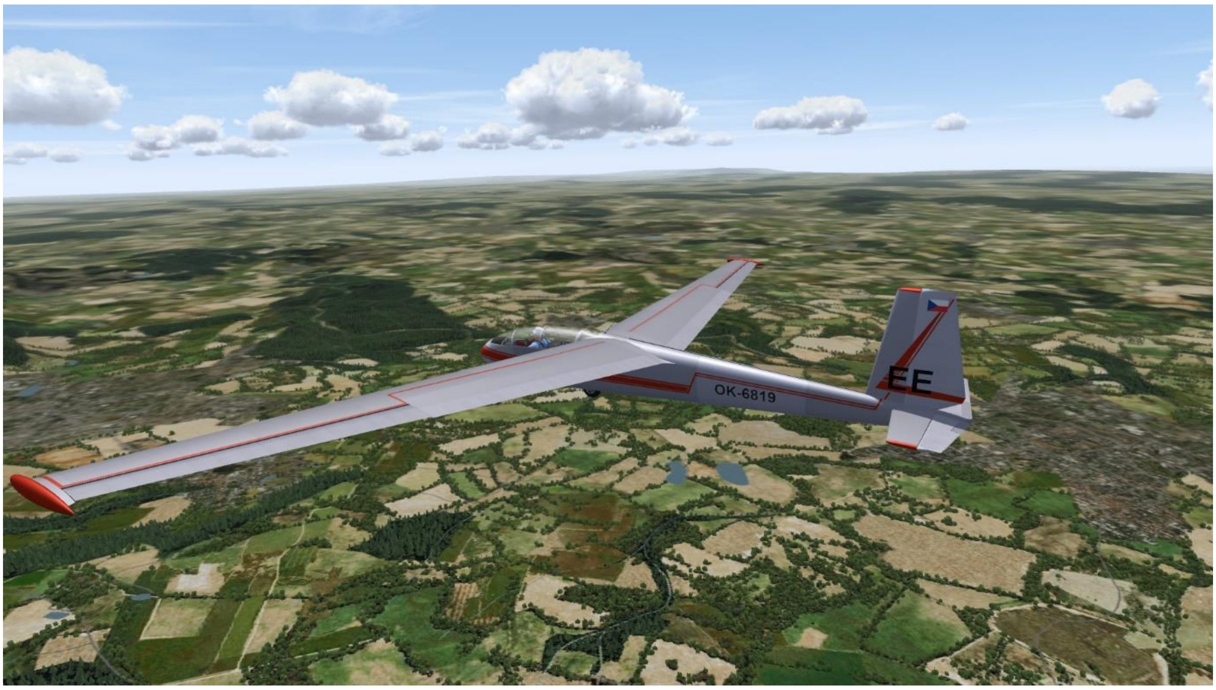
The tree areas in the texture don’t need to be fully covered with the selection, but at least some pixels in each area should be selected. We are not aiming for dense forests. In this case, less is more. Once the selection is to our liking, press **Ctrl+I** to invert it and delete unwanted parts of the texture.

Now add a new layer, fill it with black color and move it below the original background layer. You should now see faint green dots on the black background. Select the layer with dots again and from the main menu pick Adjustments > Levels. Adjust the input levels in such way that the faint dots become more visible. This will become the base for our deciduous forest maps.

However, the current size of the image is only 1024 x 1024 pixels. We need to fill the whole 8192 x 8192 pixels tile with it. Flatten the image and apply the same copy/paste & canvas expansion process we used for creating the base farmland tile texture.

Once you have created the 8192 x 8192 pixels sized image filled with our small map tile, add a new layer filled with black below the current one and save the file in .PDN format as a template for the deciduous forest map tile.

If we use the forest map as it is now, it will generate trees over the whole landscape, including places that should be void of them. To avoid this, we need to remove tree data from areas covered by residential areas, forests, waterway and waterbodies and roads. We will do it using our mask images created in QGIS.




Let’s again start with the tile 0000. Open the mask image 0000.png and paste it into our template as a new layer. Save the template as b0000.pdn. Now using the Magic Wand with flood mode set to global, select all the forests, residential areas, waterbodies, airports and roads. Hide the mask layer, switch to the black layer with dots and delete the selected areas. Save the file to keep the layered version for possible future updates. Now resize the file to 2048 x 2048 pixels and save it as BMP – i.e. b0000.bmp.

Now repeat the same process for the remaining three tiles, making b0001.bmp, b0100.bmp and b0101.bmp in the process. Those are our final deciduous forest map tiles.

#### *Exporting forest maps to Condor*

Move all the sXXXX.bmp and bXXXX.bmp files to **Working\Terragen\ForestMaps** folder. Now run **Landscape Editor** and open our Tutorial landscape. From the main menu select **Tools > Import Terragen size forest maps**. Landscape Editor will process all the forest map tiles. Check the Forest maps check box in the toolbox. You should now see our forest maps overlaid over the terrain.

Now we need to export the forest maps to Condor. From the main menu select **File > Export forest map**. Landscape Editor will process both forest maps and export them to Condor. Don't forget to save the forest hashtable too, using **File > Export forest hash**.

Close the editor, run Condor and enjoy your first flight over our landscape covered with trees and with waterbodies rendered with water effect. 😊

### Creating thermal map

Not every area of the Earth surface supports the convective activity in the same way and with the same strength. To help simulate this, Condor uses a thermal map of the landscape surface. The map has a resolution of 90 meters per pixel (i.e. 256 x 256 pixels per tile) and is implemented as grey scale image. The color of each pixel determines the probability and strength of the thermal activity at the given point, with black meaning the lowest and white the highest probability.

Please note that the thermal calculation algorithm is different from the one originally used in V1. In V2 there is no need to denote sunny slopes in the thermal map, as V2 works directly with the terrain profile. Also, thermal map values are used in connection with the value of "flats activity" set in the weather settings of the flight plan.

In short, if the "flats activity" is set to none, the thermal map is not considered at all and all the thermals are only calculated from the terrain profile and sun exposure. With such settings, the flatlands will provide little to no thermals, and thermals will mostly be only in hilly and mountainous areas. The higher the "flats activity" is, the more the thermal map is taken into account.

We will create the thermal map for our landscape using Paint.NET. For defining the fields, grass and forests we will use our mask images again.

First, let's prepare everything we need. As we have mentioned, each terrain surface has a different probability of supporting convective activity. As the probability is determined by the brightness of pixels in our thermal map, we will use the colors from the table below for each of the surface types that is important for us**.**

| **Surface** | **RGB values** | **Hexa value** |
|---|---|---|
| Water, etc. (no thermals) | 0, 0, 0 | 000000 |
| Deciduous trees, shades | 64, 64, 64 | 404040 |
| Forests and grass | 102, 102, 102 | 666666 |
| Fields | 178, 178, 178 | B2B2B2 |

Let's make a layer for each of the colors.

Create the first layer, fill it with black color and move it to the bottom of the layer stack. Continue with the deciduous tree layer, forest layer and fields layer. Fill each of the layers with the appropriate color from the above table. Finally, add the last layer and keep it empty. We will put the mask image in it in a moment.

Open our **masks.png** file and resize it to 512x512 pixels. Copy the resized image to the last layer of our thermal map image. You can then close the masks.png without saving changes, to keep the file in the original form.

We will now compose the thermal map in a similar fashion as our textures. Switch to the layer containing masks and using **Magic Wand** (with flood mode set to global and small non-zero tolerance) click on one of the orange fields. All the fields should now be selected. Press **Ctrl+I** to invert the selection and switch to the fields layer (the brightest grey of the three). Press **Delete** to remove the excess areas and only keep the parts marked as fields in our mask image.

Now we need to select all the forests. We will also select all the residential areas, as the streets and parking lots also generate lots of hot air. It is up to you, if you decide to give cities and villages less probability in your own landscape by including them in different layer. We will include them in the same layer as forests in our example.

Switch to the mask layer again and using Magic Wand select all the forests by clicking on one of them. Now press and hold the **Ctrl** key and **click** on one of the cities to select all the residential area. We should now have all the forests and cities selected. Press **Ctrl+I** to invert the selection and switch to the middle grey layer. Press **Delete** to remove unneeded parts of the image and only keep the forests and residential areas.

Now switch back to the mask layer and using the Magic Wand select all the white areas of the mask. Press **Ctrl+I** to invert the selection, switch to the layer with darkest grey color and press Delete.

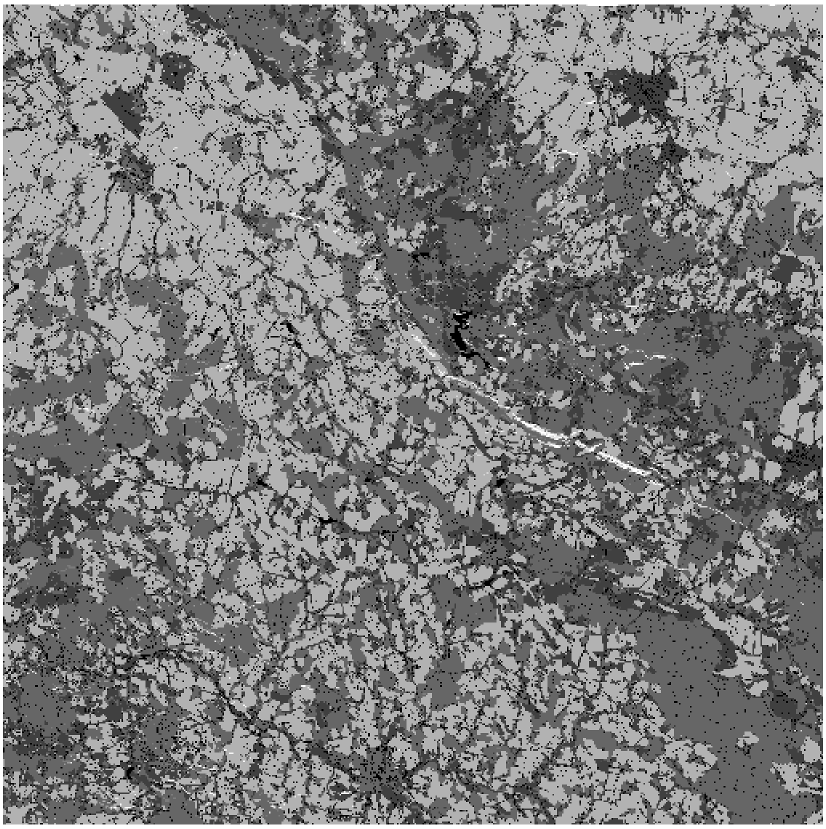


*Figure 22 – Composed thermal map for our Tutorial landscape*

Save the layered image in .PDN format to keep it for potential future adjustments. Then change the file format to .BMP and save it as ThermalMap.bmp in the root of the **\Working** folder, i.e. **\Landscapes\Tutorial\Working\ThermalMap.bmp**.

*NOTE: As you may have noticed, the structure of fields in our background texture looks different than then one in the thermal map. That's because we are using OSM data for generating thermal map. If you wanted the same field layout as in the background texture, you would have to create the field map by hand. This would be quite time-consuming, that's why we used the real-life field data to make the process quicker.*

We can now close the Paint.NET and open our landscape in Landscape Editor. Click on the Thermal Map action in the toolbox. If our thermal map appears overlaid over the terrain, we can select **File > Export thermal map** and we are done!

This concludes the first part of the scenery tutorial. However, our terrain is still void of any airports and other objects. We will discuss creating and adding those in the following chapters.

# 7. Creating and adding scenery objects, custom airports and turn-points

We have finished our terrain, but so far, the only 3D objects we see on the surface are trees generated using tree maps. In this chapter we will discuss adding custom objects and airports to our landscape. We will also mention how the objects and airports can be created, albeit very briefly, as object making is the topic of separate tutorials and guides. However, at least basic knowledge of 3D modelling is expected in this chapter.

As Condor is primarily competition simulator, we will also explain how to add turn-points that can be used for task-setting.

## Brief introduction to object making

In general, we have two types of custom objects in Condor V2 landscapes – placeable scenery objects and airports. Condor uses its internal binary format to store all objects – C3D.

Scenery object is any 3D model created to be placed at some place on our landscape. They can be anything that you find in the real terrain, like houses or whole city blocks, castles, antenna towers, bridges and even special objects like trains, parked aircraft or agricultural machinery. The purpose of scenery objects is to make the landscape more plastic and closer to real life. Content of one C3D file is considered one scenery object, even though it may be a complete scene composed of multiple 3D objects.

Custom airports are special types of objects representing whole airports, including runways, hangars, other buildings, cars, static aircraft etc.

Both scenery objects and airports are placed into the landscape using the Landscape Editor. However, based on their purpose, the files itself are stored at different folders in the landscape folder structure.

Placeable scenery objects are stored inside the **\World\Objects** folder of each landscape. Their textures should be stored inside the Textures subfolder there, e.g.:

- object files: **\Landscapes\LandscapeName\World\Objects**
- textures: **\Landscapes\LandscapeName\World\Objects\Textures**

Airports are stored inside the \Airports folder of each landscape. Each airport consists of two files – a *G file* and an *O file. G file* contains all the ground surface objects of the airport (grass and/or asphalt runways, taxiways and their marking). *O file* contains all the other 3D objects of the airport model – hangars, tower, windsocks, aircraft, cars, trees, etc. All the objects included in O file are "crashable", i.e. they cause crash when collided with.

It is recommended to create a separate subfolder for textures unique to each airport (name of the folder is up to you, but using a comprehensible identifier, like ICAO airport id, is recommended). If needed, another subfolder can be used for common textures shared between multiple airports. The recommended folder structure looks like this:

- airport G and O files: **\Landscapes\LandscapeName\Airports**
- unique airport textures: **\Landscapes\LandscapeName\Airports\XXXX**
- shared textures: **\Landscapes\LandscapeName\Airports\Common**

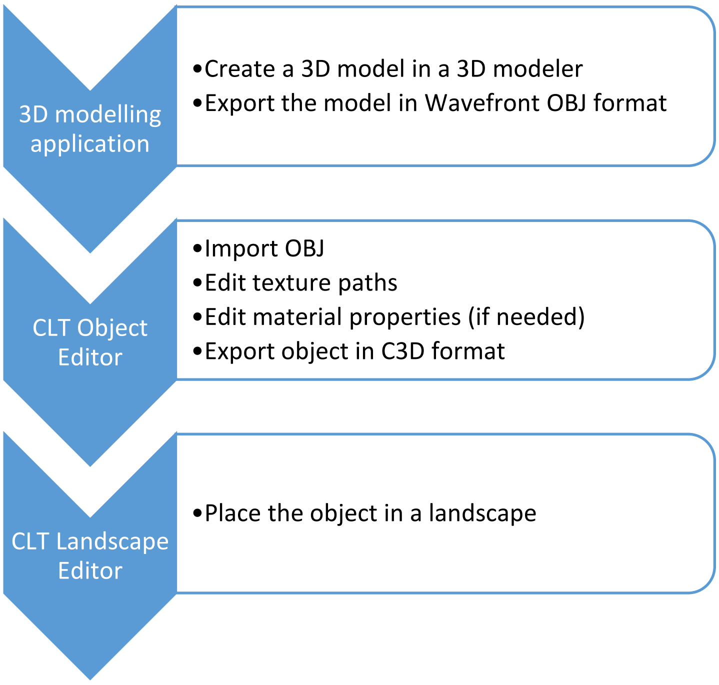
It is recommended to use relative path to the textures in your objects, as it allows easier transfer and reuse of objects, especially if you are making objects that can be used in multiple landscapes.

## Tools needed for object making

You need multiple tools to create and export 3D model to be used as scenery object in Condor. However, the work-flow changed from V1 and is now more open and straightforward. The most significant change is in using the Wavefront .OBJ file instead of 3D Max .3DS used in the V1. This allows the object creators to use wider variety of modelling software, ranging from paid (and expensive) applications like Autodesk 3D Studio MAX and Rhinoceros to open source alternatives like Wings3D or Blender.

*NOTE: Wings3D has been the preferred application in the Condor core development team since V1, but it will most likely be replaced by Blender soon. However, this in no way means that those are the only possible alternatives. Some gliders and tow planes in V2 were created in 3D Max, scenery objects in 3D Max, Blender and Wings3D. So, as long as you are able to export in OBJ format, you may use whatever software you want.*

General work-flow of creating a Condor V2 object looks like this:




## The most important general rules for creating landscape objects and airports

There are few important rules that you need to consider when making objects for V2. Even if you are a skilled author of objects for V1, do not underestimate the requirements, as some of them are very different than in V1. For the sake of clarity, from now on we will use the term **"landscape object"** for the whole C3D file and **"scene object"** for all the component objects inside it.

When modelling, keep the following in mind:

1. Condor uses meters as units of measurement. Set your modelling software accordingly.
2. Multi-materials are not supported in V2. Each scene object must have only one material assigned. This also means that each scene object can only have one texture. If you are trying

to adapt V1 multi-material object to V2, you either need to retexture it or divide it into multiple objects, each textured with only one texture.

3. Unlike V1, material color of a textured object influences the way the object is rendered in game. If the material color is different than full white (R: 255, G:255, B: 255), the texture on the object will appear darker and/or tinted. (TODO: picture with example)
4. Landscape objects in V2 are dynamically shaded. It is recommended to avoid pre-shading in object textures, if possible.
5. Ground surface polygons in airport *G file* must not overlap, otherwise the visual result is unpredictable. Scene objects from *O file* can "stand" on ground polygons defined in *G file*.
6. Up to 3 dynamic windsocks can be placed in an airport *O file*.
7. Airports don't have any underlying "grass" polygon. Airport grass area should be drawn on the landscape texture.

## Basics of creating airports

As previously mentioned, airports in Condor V2 consist of two object files, *G file* and *O file*, leaving out the *S file* used in V1 to define ground surface.

### G file – airport ground polygons

Airport G file consists of up to four scene objects, each having a specific purpose. Each object can consist of multiple polygons, but they need to be combined to form one object only (not an object group!).

#### *Grass* object

In V2, grass object is not used for drawing grass area of the airport the same way as in V1. It is used to add hi-res grass surface effect to grass runways and adjacent areas. Think of this object like if it was a special semi-opaque decal displayed over the regular landscape surface. When you get to a certain distance of the object, it will start to appear. The closer you are, the more visible it is. Once you are on the ground, you will see the grass generated by this object instead of the underlying landscape texture.

To create the grass object, make a polygon of a required shape. If needed, you can create multiple polygons, but they need to be combined into a single polygon object in the end. Change the name of the object to *Grass*.

To properly show the grass effect, the object needs to be applied UVW mapping. This procedure is different in each modelling application. Depending on application used, UVW mapping can be done with or without a real texture assigned to the object. The texture assigned to the object is ignored in game and default grass effect texture is used.

The size of the grass effect (how big the grass leaves are) is determined by the area of the texture mapped to the object – the smaller the area, the larger the grass. This can be adjusted by unwrapping UVW and changing the size of the mapped part of the texture. As each runway is different, the right size of the mapped texture is a matter of trial and error. Try changing it, until the result in game is satisfactory.

#### *Asphalt* object

Asphalt ground object should be used to portray asphalt runways, taxiways and other paved airport areas.  Unlike the Grass object, the Asphalt object is always visible. However, the detailed surface structure is shown in the same style as for grass and the same UVW mapping process applies there too.

All the asphalt polygons should be combined into one object named *Asphalt*.

### *Grasspaint* and *Asphaltpaint* objects

Runway and taxiway markings are done using *Grasspaint* and *Asphaltpaint* objects. Each of the objects is specific for the underlying surface type. "Paint" objects can be modelled both textured and untextured, but in game the assigned texture is ignored and default texture for the particular "paint" type is used.

*Grasspaint* texture represents grass leaves painted white (in similar style as lime marking on a football field).

*Asphaltpaint* texture shows crackled white paint on an asphalt surface.

Use of the *Grasspaint* and *Asphaltpaint* is not limited to the grass and asphalt surfaces respectively. As many airports use solid runway markers even for grass runways (concrete blocks or tiles painted white), we can use *Asphaltpaint* on *Grass* object to portray this too. However, using *Grasspaint* on *Asphalt* is not recommended.

### Notes on making a *G file*

Not all the objects need to be present in a G file, but when they do, it is important to make sure that the *Grass* object and the *Asphalt* object do not overlap. As mentioned in the previous chapter, it leads to undesirable results. It is also recommended to spread the *Grass* object over the whole grass area of the airport, so the hi-res grass effect is present over the whole airport, not just the runway. *Grasspaint* and *Asphaltpaint* objects can be placed over the *Grass* and *Asphalt* objects as needed, without any unwanted effect.

## O file – all airport objects

Airport *O files* contain all other airport objects – buildings, hangars, ground equipment or vehicles. *O files* can also contain up to three dynamic windsocks.

### *Adding windsocks*

To place a windsock in the airport model, we need to make a proxy object for it. As we can have three windsocks, the proxy objects are called **Windsack1**, **Windsack2** and **Windsack3**. The position and size of the proxy object determines the point where the windsock is attached to the pole and its size.

The windsock proxy object is an equilateral triangle. The center of the triangle is the point where the windsock ropes are attached to the pole. The size of the windsock itself is the **centuple of the triangle's circumcircle radius**.

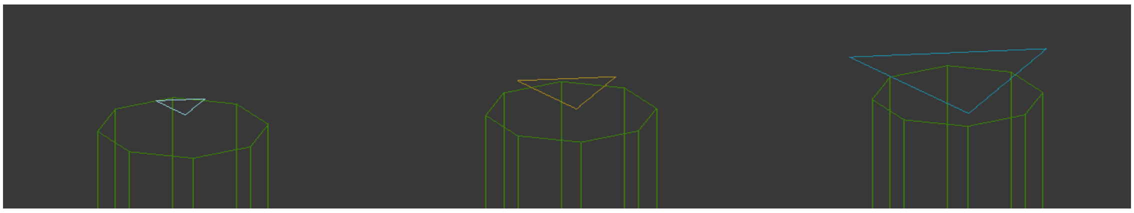


*Figure 23 – 2.5 cm, 5 cm and 10 cm windsock proxy objects*

Figure 23 shows triangle proxies defining three windsock objects. Each triangle is placed 2 cm above a 10-meter-high pole with 15 cm diameter. From left to right, the radii of their circumcircles are 2.5 cm, 5 cm and 10 cm. Figure 24 is an in-game screenshot of the same poles placed at an airfield. From left to right, their lengths are 2.5 meters, 5 meters and 10 meters.

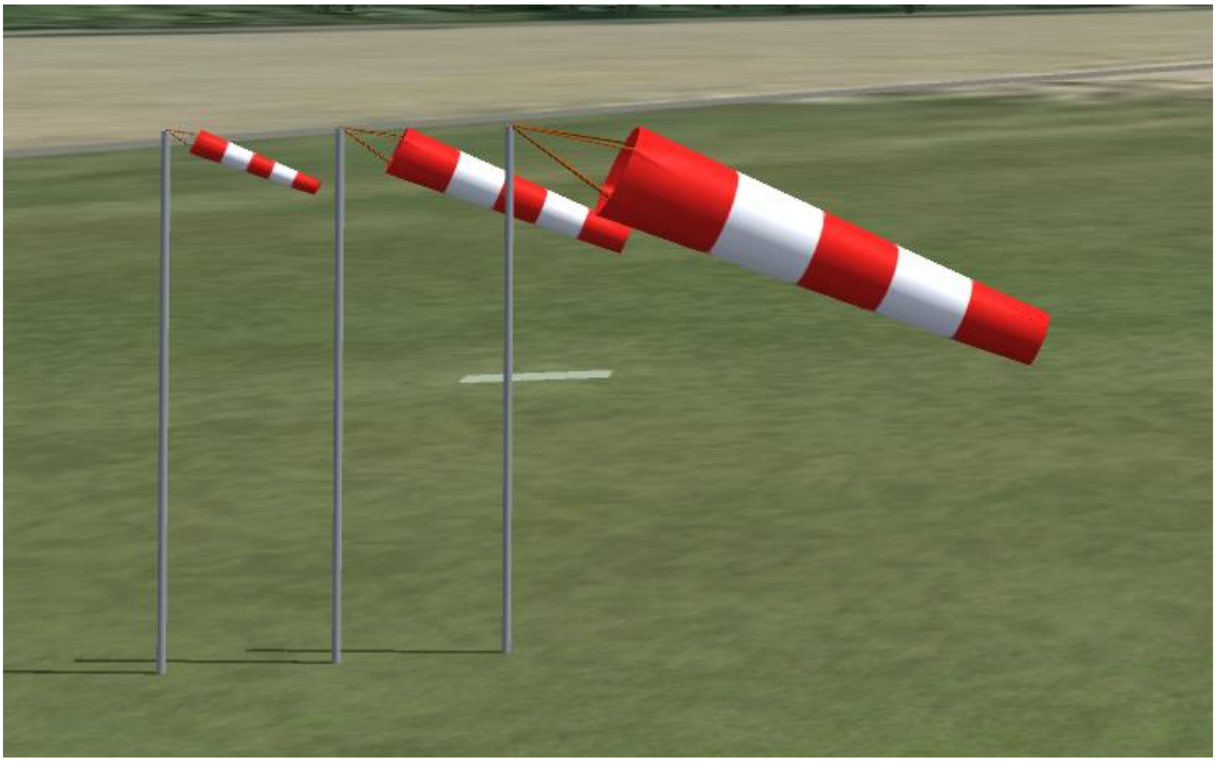


*Figure 24 – Windsocks generated by proxy objects from Figure 23*

## Modelling a simple generic airport for our tutorial landscape

This chapter will briefly describe the workflow of modelling a simple airport that we will be able to place in our tutorial landscape.

Please note, that although the images illustrating this chapter are taken in 3D Max, the process itself is described in general terms. At least basic knowledge of any 3D modelling software is expected.

Our airport will only have a grass runway with runway markers, one hangar with small asphalt area in front of it and a windsock with ground marker.

### Creating ground polygons

Airports in V2 don't have any underlying surface polygon defining the actual airport area and rely on the area already being depicted in the landscape texture (we have addressed this issue in the chapter about landscape textures). This approach was in fact used by many V1 sceneries too.

***NOTE: Before you start modelling, verify that you are using meters as units in your modelling software. If not, change unit settings to meters.***

Let's start by making the runway. It will be 800 meters long and 60 meters wide. It will be a helper object, as it will only be used for reference and won't be exported to Condor. The length of the runway should be along the X axis. Create a polygon of the above given size and place it to have its center point at the origin coordinates: 0, 0, 0. This is our reference point for the whole airport.

Now we will prepare the *Grass* object. We will make another polygon, this time with a little bit more complicated shape (see the picture below). We will name this object "Grass".

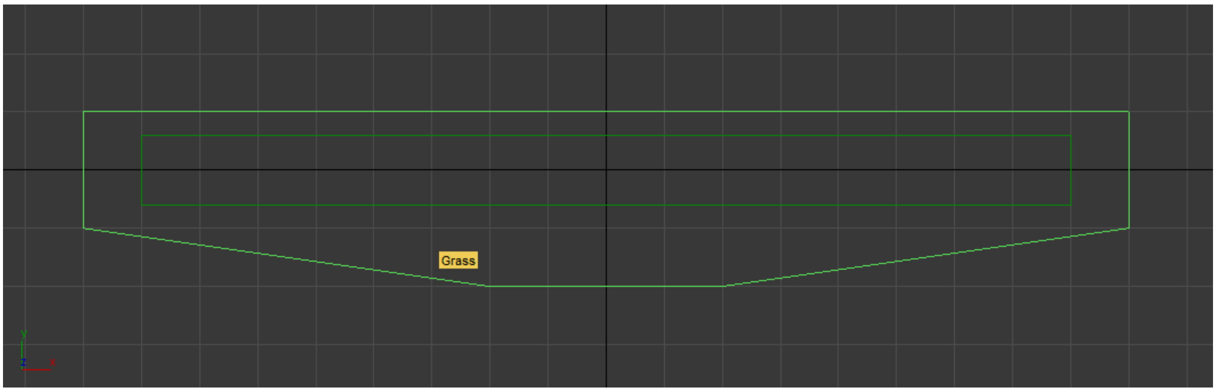



*Figure 25 – Grass polygon with runway helper polygon*

Now, let's make the runway markers. We will make them as separate polygons first and then combine them into single *Grasspaint* object. In total we will have 4 corner markers and 7 pairs of regular border markers. Border markers will be 3 meters long and 1 meter wide rectangles and will be placed along the runway border every 100 meters. Corner markers will be L-shaped polygons, 2 meters wide and 4 meters long, and one will be placed at every corner of the runway.

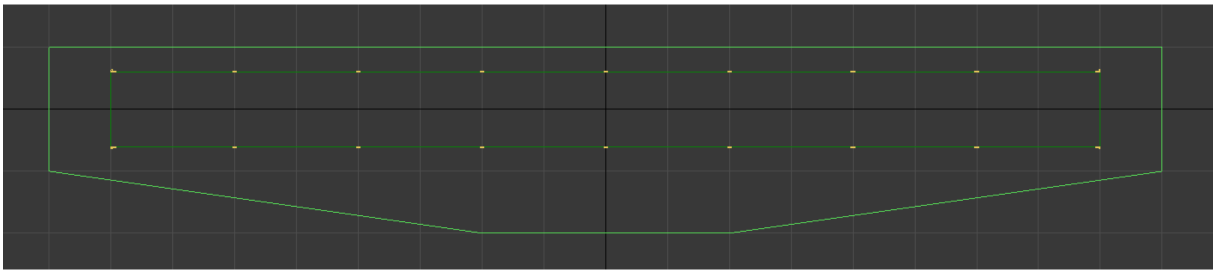


*Figure 26 – Grasspaint runway markers*

Another ground marking that we need to create is the circle around the windsock. It will have the outer diameter of 10 meters and thickness of 0.5 meter. We will later place the windsock in the center of it.

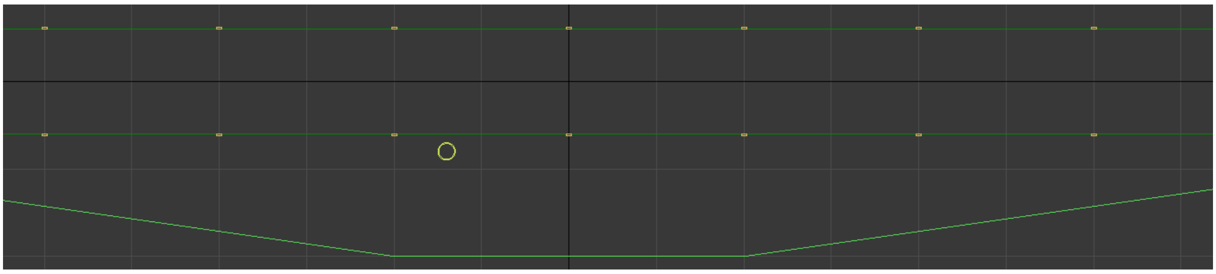


*Figure 27 – Windsock circle marker*

Let's add the last object that will be part of the *G file* of our airport – the asphalt area around the hangar. Horizontal dimensions of the hangar will be 40 meters by 15 meters. The size of our asphalt area will be 60 meters by 35 meters and the name of the object will be *Asphalt*. We must not forget that *Grass* and *Asphalt* objects cannot overlap. We must cut a hole for the asphalt area inside our *Grass* object (e.g. using Boolean cut).

We have all the ground polygons we need. Now we need to fuse all the ground markings into the *Grasspaint* object. Once we are done, we can move to another very important step – we need to apply UVW mapping to all three ground objects.

First, we need to create a new material and set its color to white. It doesn't have to be textured, but it may help for better visualization of the airport in your modeler. The textures will be ignored in game. Once the materials for all the objects are set, we can start mapping.

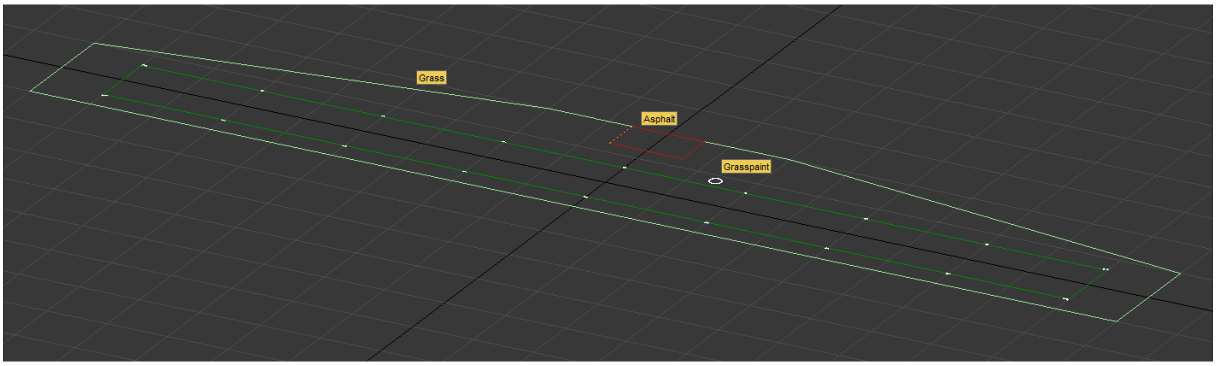



*Figure 28 – All the ground objects of our tutorial airport*

Let's start by applying UVW mapping to the *Grass* object. The larger the portion of the texture taken by the object, the smaller the grass hi-res detail rendered in game will be. This principle applies to all the *G file* objects.

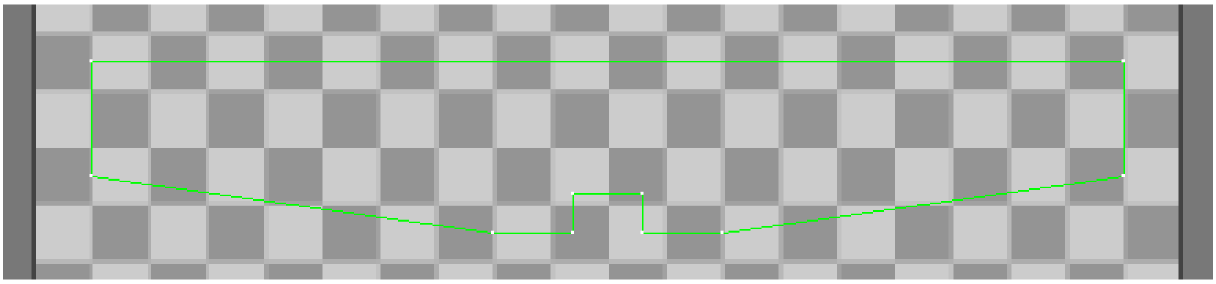


*Figure 29 – UVW mapping the Grass object*

After we map all three objects, we can consider the content of the *G file* of our airport to be ready.

We will now add a hangar and a pole with the windsock proxy object to our scene.

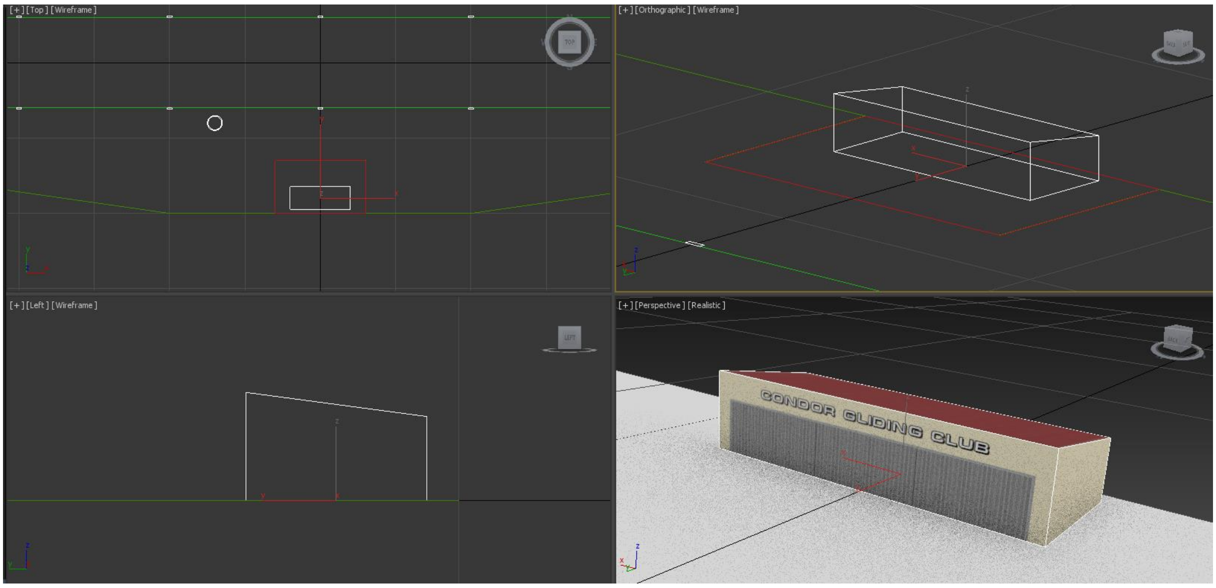



*Figure 30 – The hangar object*

The hangar will be built as a box, 40 meters wide, 15 meters deep and 9 meters high. Change the height of the back wall to 7 meters and apply some texture. You can use the hangar_texture.bmp available with this guide.

Now we will create the windsock pole and windsock proxy object. We have already created the ground marking for the windsock and the pole will be placed right in the center of it. We will make it as 8 meters high and 10 centimeters wide cylinder. The proxy triangle object will be placed on top of it and the radius of its circumcircle will be 4 centimeters (it will generate 4 meters long windsock). We will change its name to Windsack1. 3D Max users beware, all the objects need to be converted to editable mesh or editable poly before exporting! Otherwise the objects may not be exported.

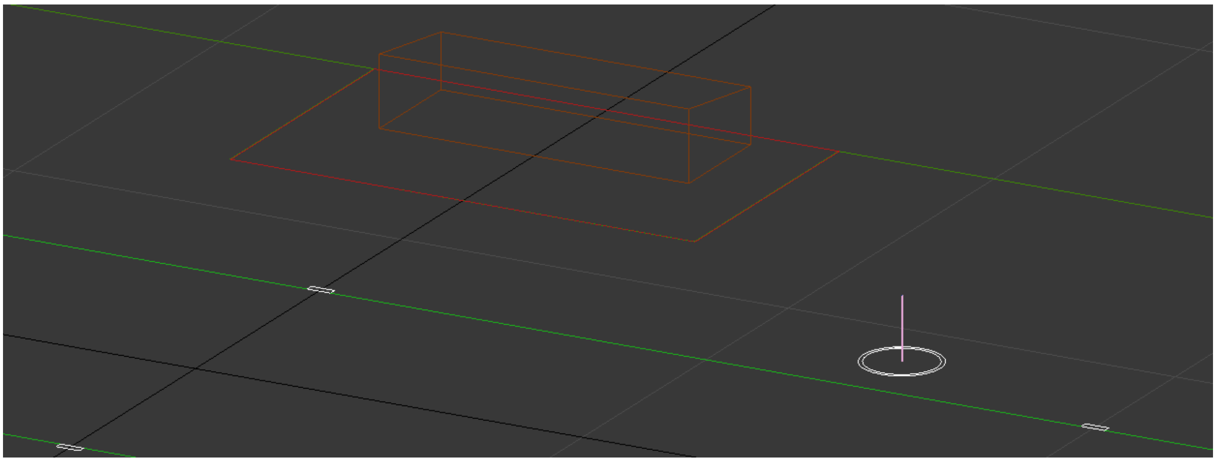
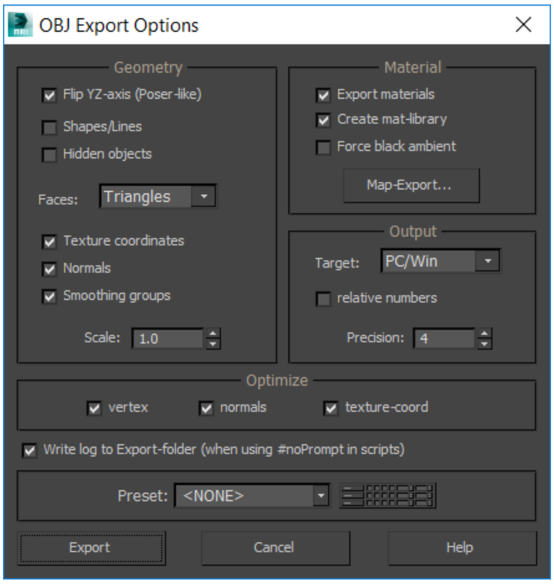


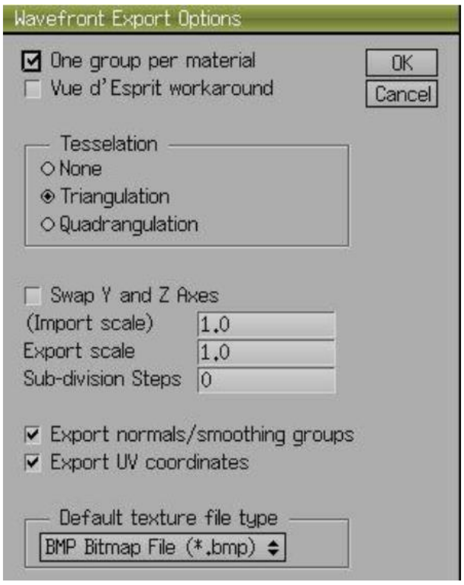
*Figure 31 – Added windsock pole and windsock proxy triangle*

Save your file. Now we can export both the *G file* and *O file*.

Select the *Grass*, *Asphalt* and *Grasspaint* objects and export them in the Wavefront OBJ format. For the *G file* the filename must be XxxxG.obj, where Xxxx is the name of the airport we will use in Landscape Editor. We will export the file as TutorialG.obj for now.

It is very important to export the faces as triangles during the export. Also, for applications like 3D Max, where X and Y coordinates are on the horizontal plane and Z coordinate is in the vertical direction, don't forget to set the export to flip YZ-axis too. 3D Max and Wings3D export settings are below:



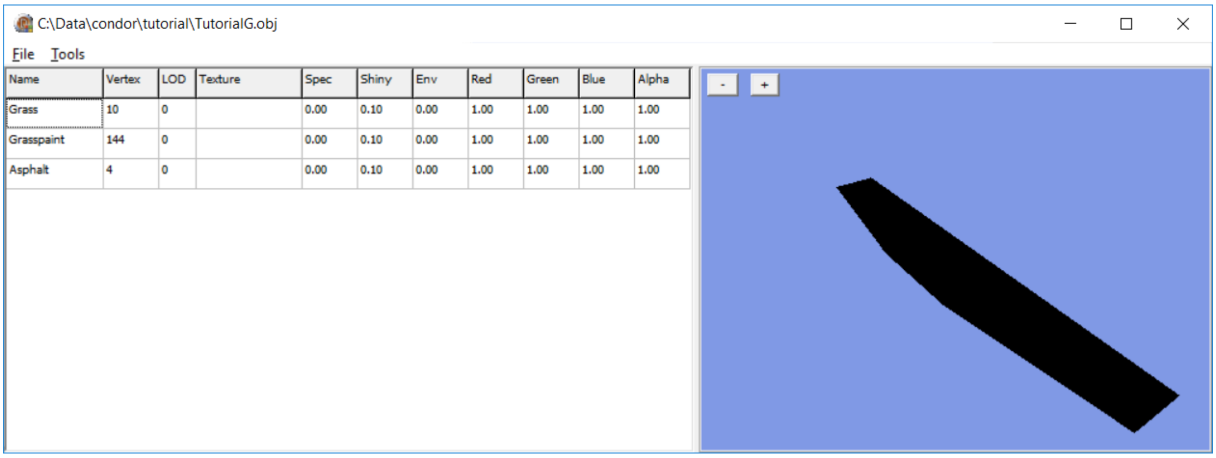



Now select the hangar, windsock pole and windsock proxy triangle and export them to OBJ using the same process too. Save the file as TutorialO.obj.

Save your airport and close the 3D modelling application, it won't be needed now (unless you decide to go back and change something right now).

Now we need to convert both OBJ files to Condor's internal C3D format. Start **Object Editor** and open the TutorialG.obj. Don't be afraid that the object is displayed completely black, Object Editor shows all the untextured objects this way after OBJ import. All the material colors will be present in the saved C3D.

Check if all the objects have 1.00 value in their Red, Green and Blue attributes. If not, change them to this value. Then save the file to C3D format using **File > Save to C3D**.




| Name | Vertex | LOD | Texture | Spec | Shiny | Env | Red | Green | Blue | Alpha |
|---|---|---|---|---|---|---|---|---|---|---|
| Grass | 10 | 0 | | 0.00 | 0.10 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Grasspaint | 144 | 0 | | 0.00 | 0.10 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Asphalt | 4 | 0 | | 0.00 | 0.10 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Load the TutorialO.obj. For *O files*, always look at all the objects with texture (Texture attribute is not empty) and check if their material setting is 1.00 for Red, Green and Blue, as in the previous step. If not, change it this way. As for untextured objects the color is determined by the color of their material, keep their settings as they are.

In our case, we only have one textured object – the hangar. Apart from checking the R, G, B values, we also need to adjust the texture path. As our airport files will be used for every airport in our landscape, which means it will eventually be saved under multiple names, we will set the texture path to **"Common\hangar_texture.bmp"**. Of course, we need to copy the file to this location. For our example it would be in **Landscapes\Tutorial\Airports\Common**.

As we have already placed one airport in our landscape, let's also copy both C3D files to the Airports folder as PodhoranyG.c3d and PodhoranyO.c3d.

## Working with airports in Landscape Editor

Start the Landscape Editor and open our Tutorial landscape. Tick Airports to see airports and select the only airport we have added so far. Tick Draw objects to see wireframe of all the airport objects. You will see that our generic airport is little bit longer than the real runway and that our hangar is outside the grassy area of the real airport. Easy fix is to turn the runway by 180 degrees, i.e. to enter 253 degrees as runway direction, instead of the original 73 degrees. Then the airport should look like below.

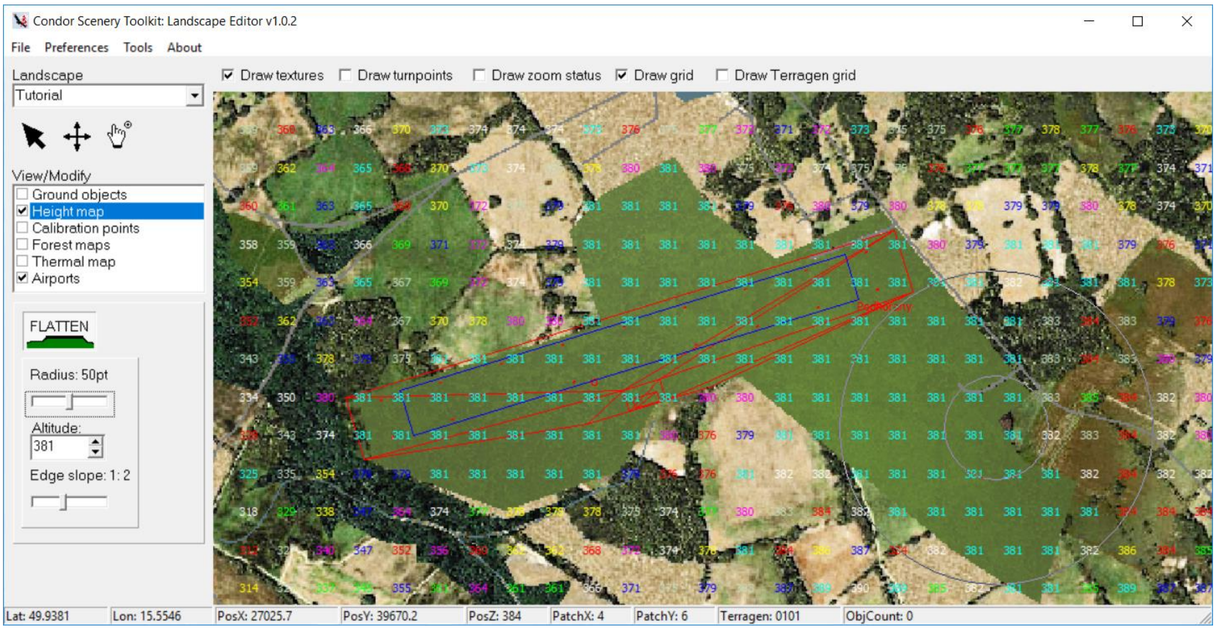

**Starting with Condor 2 version 2.0.8 the direction of airport runway is expressed as a decimal number. This gives scenery makers more flexibility when aligning airport models with textures (especially photo-textures). This feature is supported by Landscape Editor version 2.0.1 and above.**

**Landscapes created using Landscape Editor 2.0.1+ can only be used in Condor 2 version 2.0.8 and above! Condor 2 is still compatible with older landscapes created using the Landscape Editor version 2.0 and as long as there is no need to use the decimal direction, no changes need to be mad to already existing landscapes.**

We still haven't won yet. You would see that if you tried to use the airport for take off or landing. To be able to use it properly, we need to flatten the surface below the airport. Remember the airport altitude (381 meters), as we must flatten the area to this level.

Switch to the Height map view and zoom to the airport. Then click the **FLATTEN** button to activate the flattening function. Set radius to whatever value suits you and altitude to 381 meters. Then edge slope to some other value than 1:0. This will activate surrounding altitude smoothing. The higher the ratio, the greater the area affected by smoothing. This helps blending the flattened and the surrounding not flattened surface. Note the cursor meaning – the inner circle is the area where the altitude is set to the entered value, the outer circle denotes the area where the terrain mesh is smoothened to blend with the flattened area.

Remember the flattening function, it will come handy when you will be placing landscape objects.




Finally, once we flatten the whole area under the grass patch in the texture, we can save the landscape and try it out in Condor. Don't forget to export the new terrain hash table too!

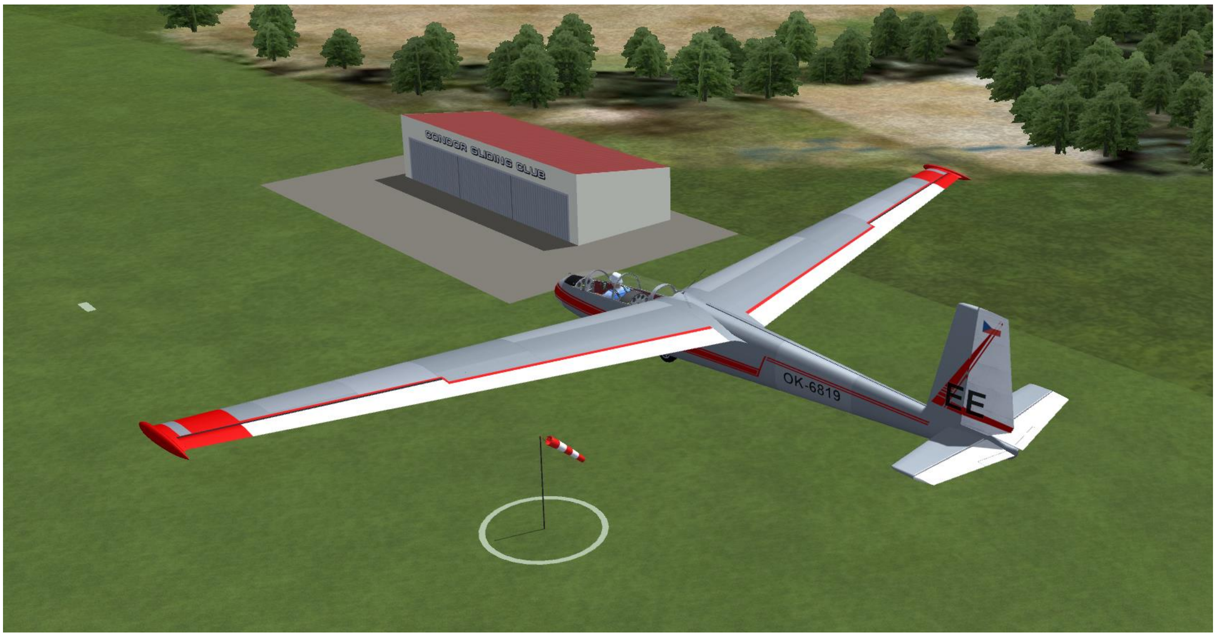



*Figure 32 – Our custom airport in Condor*

Here is the list of all the airports that are still missing in our landscape. You can add them the same way as we added Podhořany.

Havlíčkův Brod (in game name "Havlickuv Brod")

> Lat: 49.601898
> Lon: 15.536500
> Runway - length: 1000 m, width: 50 m, direction: 113 degrees
> Altitude: 450 m
> Filenames: Havlickuv BrodG.c3d, Havlickuv BrodO.c3d

Chotěboř (in game name "Chotebor")

Lat: 49.684925
Lon: 15.675970
Runway - length: 1010 m, width: 50 m, direction: 167 degrees
Altitude: 580 m
Filenames: ChoteborG.c3d, ChoteborO.c3d

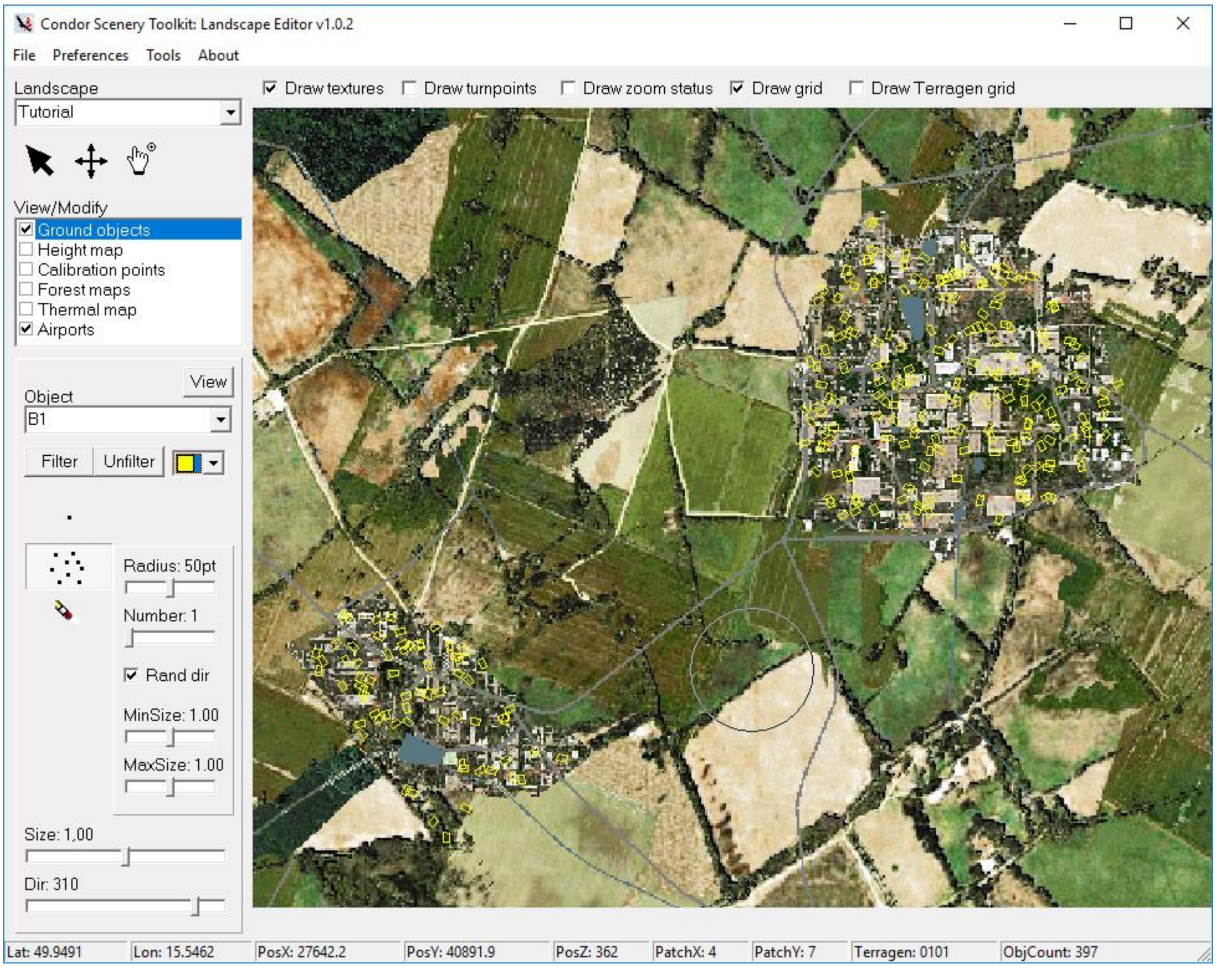
Chrudim

Lat: 49.945400
Lon: 15.771000
Runway - length: 980 m, width: 160 m, direction: 230 degrees
Altitude: 290 m
Filenames: ChrudimG.c3d, ChrudimO.c3d

## Adding landscape objects

Landscape objects are created the same way as airports. As in airports, one c3d object can consist of one or more scene objects.

The Slovenia2 scenery contains tens of objects that are free for every scenery maker to use. To use them, they just need to be copied to our landscape's **\World\Objects** folder. Don't forget to check the texture paths in **Object Editor** and change accordingly. Textures used by the objects should be placed in **\World\Objects\Textures** folder.



To place objects in the landscape, select the Ground objects function. Pick the object you want to place from the Objects drop down list. You can either place one object or a cluster of multiple objects at a time. Cluster option allows you to randomly placed objects in an area of the given radius.

When zoomed in entirely, you can place single objects very accurately (put trains on railway tracks etc.).

To remove objects, use the selection mode (big black arrow), click on an object or select multiple objects by dragging selection rectangle over them and then press Delete. You can also use the eraser tool.

If you are placing larger objects, it may be necessary to flatten the surface underneath them. If the surface beneath the object is not flat, part of the object can be in the air and another part sunken in terrain.

Placing landscape objects is another trial and error process. Eventually, every scenery maker will find the right balance between too many/few objects and good looks of the landscape. But as you can see in the picture below, landscape objects will make the scenery feel more alive.


*Figure 33 – Terrain with added village houses (top left)*

## Adding waypoints

After all, Condor 2 is a competition soaring simulator, so we need to provide task setters with waypoints that can be used for creating competition tasks. To be able to create task with turn-points other than airports, we need to add a list of custom waypoints.

Waypoints are stored in a CUP file placed in the root folder of the landscape. The file has the same name as the landscape, e.g. **Tutorial.cup**, and is in fact a standard See You waypoint file. There are multiple ways to create or get waypoint file for your scenery, from compiling it manually to generating it using specialized software.

One of the sources of waypoint databases is the Worldwide Soaring Turnpoint Exchange website available at http://soaringweb.org/TP. One of the features of Condor is the reduction of waypoint lists – if you provide a waypoint list with waypoints outside your landscape, Condor will automatically remove those waypoints and will only keep those inside the landscape area. This happens automatically when you start a flight in the landscape. To get some waypoints to our landscape, just download any CUP file for the Czech Republic from the website above, rename it to Tutorial.cup and place it to the Tutorial landscape folder. Next time you will plan a flight in the landscape, you should see the waypoints in the flight planner and after you finish your flight, only the relevant waypoints will remain in the waypoint file.

The other option to get customized waypoint file is using the See You application made by Slovenian company Naviter. See You is currently one of the standards for task setting, flight planning and post flight analysis. However, it is payware, available freely only for 14 days trial period. See You can be downloaded from http://www.naviter.com/products/seeyou/.

***This concludes our landscape tutorial. Congratulations, if you have read this far, you should know all you need for making your own scenery 😊. For everything else there is the Scenery development forum at the Condor website.***

# 8. Converting V1 scenery for use in V2

Many scenery makers will ask: "How do I convert my old V1 scenery for use in V2?" To be honest, the answer should most likely be: "You don't!" But as this isn't the answer most people would like to hear, we will discuss possibilities and required steps for converting V1 scenery to V2 in this chapter.

The first question you should try to answer is how much you expect from the converted scenery. Is a mere conversion with the same terrain data as in V1 enough? Or do you want to improve the terrain by using 30 m data? Create better textures? All of this influence the conversion work-flow. Eventually you may find out that in certain situations making a new scenery from scratch would be easier.

Let's have a look what is needed to be done if V1 scenery is to be converted to V2:

1. Terrain needs to be converted to the resolution used in V2.
    a. If you intend to keep the old 90 m terrain, then all you need to do is to convert it using the **TrnToTr3** tool. This will save the old terrain in the new format but keeping the original low resolution. The terrain will remain calibrated and all you need to do is to check the flattened areas. If you choose to do this, you can skip to step no. 4.
    b. If you want to increase the terrain resolution by using 30 m terrain data, you need to create the whole terrain from scratch. There are two possibilities:
        i. You know the exact coordinates of one of the corners of your scenery. In such case you can create the terrain the way it is shown in our tutorial.
        ii. You don't know the coordinates. In such case you must use the procedure described later in this chapter.
2. Terrain needs to be recalibrated (either during export in **RawToTrn** or manually in **Landscape Editor**).
3. Terrain under large objects and airports need to be flattened.
4. Terrain textures need to be revised. Processed textures from V1 cannot be used directly as they are rotated by 180 degrees. Again, multiple scenarios are possible:
    a. If you are satisfied with texture resolution of your V1 scenery and you still have the original unprocessed tile textures, you can use them. If you import them using **Landscape Editor**, they will be cut to patches correctly.

b. If you are satisfied with texture resolution of your V1 scenery but you don't have the original tile textures, you need to reconstruct them from the processed patches. Use **TextureRecomposer** for that. You can then import them using **Landscape Editor**.

**Beware – the terrain textures are stored in the DDS format, which is a lossy compressed format. TextureRecomposer reads the DDS patch textures and restores the tile textures in BMP format. When exporting correct textures from the Landscape Editor, the images are compressed back to DDS, which may result in certain level of image degradation.**

c. If you don't like the original textures due to their looks, resolution etc., you must create new ones. You can use up to 8192 x 8192 resolution per tile.

**If your textures are preshaded, consider reworking them from scratch. Condor now shades the terrain and objects in real time and preshaded textures may look weird. If you want to keep them this way, you can turn off terrain shading, but this way you lose one of the new cool features of V2.**

5. If you want to have water bodies with water effects, you need to add alpha channel with water information to your textures. See chapter "Adding water" of our tutorial.

6. All the landscape objects need to be converted from the original V1 CX format to the current C3D format. Fortunately, Condor does that for you. It won't spare you from checking and editing the files one by one using the **Object Editor** though. As the name of the landscape will most likely be different, texture paths will need to be changed too. Don't forget to set material color to white. Also consider rearranging folder structure to fit the proposed way of storing objects and textures. See the chapter about objects and airports of our tutorial.

7. All the airports need to be remodeled as their structure is different in V2. Only the O file can be used after conversion to C3D (see the previous step for what not to forget), but you still need to create a new G file. See the chapter about modeling custom airports for more details.

8. You need to create new coniferous and deciduous forest maps, as V2 uses much higher resolution than V1.

9. You need to create new thermal map of the scenery.

So, as you can see, converting V1 scenery to V2 really isn't a piece of cake. When doing so, you should really consider if the effort is worth it.

### Replacing original terrain data with 30 m version using the original terrain file

This process is not easy and if not done properly, it may introduce precision errors to your new landscape.

First, you need to download the 30m height data as described in the landscape tutorial. Do all the steps up to using the clipper function, but instead of using the coordinates of our tutorial landscape, use the coordinates of your V1 landscape. Merge and reproject the data to the appropriate WGS 84 UTM zone, but do not clip them. We will need them all. Note the UTM coordinates of the top left corner of the uncropped terrain.

Find the BIL file created during reprojection and read its width and height from its HDR file. Open **RawToTrn**, set the width and height from the HDR file, tick **“flip vertically”** and **“SRTM 30m”** and load the BIL file. When the file loads, save the terrain heightmap as BMP image.

Then tick **“SRTM 90m”**, **“Swap width/height”**, **“Flip horizontal”** and **untick “Flip vertical”**. Now load the TRN file from the original V1 scenery. Save it as bitmap too.

For the next step we will use Paint.NET (you can use any other graphics editor capable of working with layers). Load the bitmap saved from the 30m height data BIL file.

Next, open the BMP file saved from the original TRN file. Resize the image to **300%**. Select the resized image, copy it to clipboard and paste it as a new layer over the large 30m image. Set blending mode of the added layer to **Difference**.

Now move the content of the layer to align it with the 30m data. The layer is aligned when the smaller image turns pitch black. This means all the terrain features of the old terrain are aligned properly atop the new 30m data.

Once the layer is aligned, we need to find out the pixel coordinates of the left top corner of the black area. They are the X and Y offsets from the 0,0 origin in the top left corner of the uncropped terrain image. We will use them together with dimensions of the resized bitmap made from the original TRN to cut out the 30m heightmap for our landscape terrain.

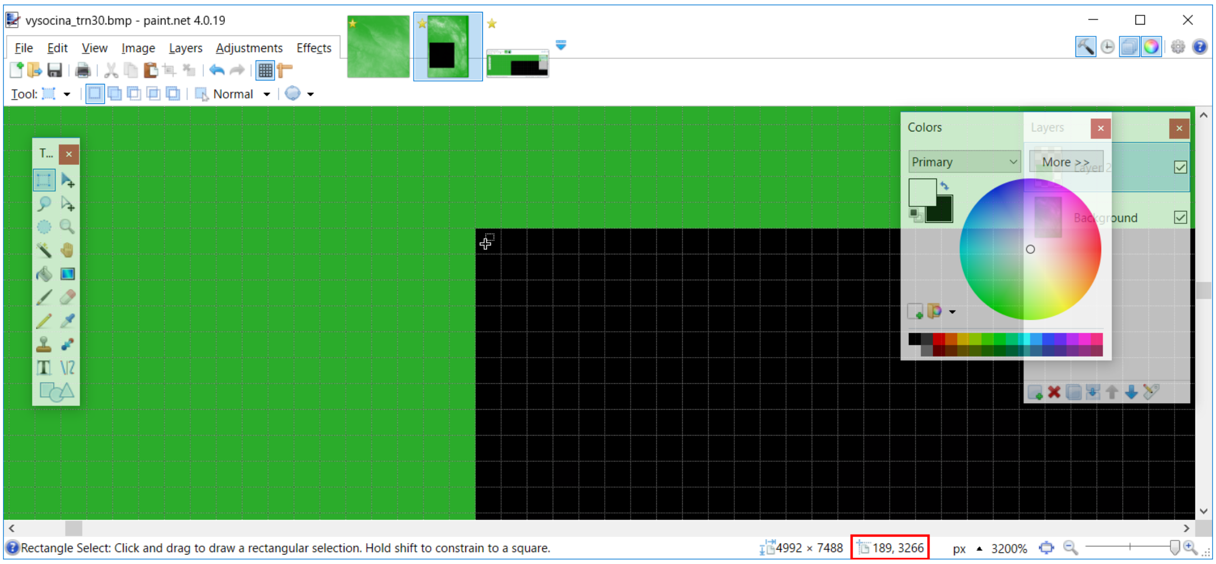



*Figure 34 – Finding the top left corner pixel coordinates (highlighted in the red rectangle) in Paint.NET*

Open **RawToTrn** and load the large BIL file again, using the same settings as before. Switch to Crop tab and select Exact crop. Enter the X and Y offsets and width and height of the resized terrain.

Switch to the Save tab. We will calibrate the new terrain during export. Fortunately, we know the UTM coordinates of the top left corner, so the top left calibration coordinates can be calculated as follows:

TL calibration point easting = $TL_{easting}$ + (30 * X offset)

TL calibration point northing = $TL_{northing}$ - (30 * Y offset)

Fill in the UTM zone, select Top Left corner and fill in the calculated easting and northing. Click Save to TRN button. Navigate to the folder you want your TRN file to be saved in and don’t forget to create **HeightMaps** subfolder there. RawToTrn will save TR3 terrain patches there.

As we have calibrated the terrain during export, we don't need to do it again in Landscape Editor.

**NEVER use the old TRN file from the original V1 scenery with the new TR3 heightmap files! It may cause unpredictable behavior.**

## Restoring the rest of the landscape

Now that we have the new calibrated terrain, we can apply the tile textures. As we have mentioned in the beginning of this chapter, there are multiple scenarios. Let's assume you have prepared the textures and they are placed in the **\Working\Terragen\Textures** folder. Load them to Landscape Editor and you should see your new terrain covered with them (more details of working with textures in Landscape Editor are in the related chapter of our tutorial). You may export them to Condor for the first tests.

If you want to add water alpha channel to the textures, read the "Adding water" chapter of the tutorial. Textures with water alpha channel must not be exported from the Landscape Editor! Use Water Alpha tool instead.

Create both forest maps (see the related chapter in the tutorial for more details), import them to Landscape Editor and export to Condor.

Put all the CX objects from V1 scenery to the same folder in your V2 scenery. Copy the landscape.obj file with object placement definition to the new V2 landscape and rename it accordingly. When you start a flight on the landscape, Condor will convert them to C3D format. You still need to check every converted object manually using the Object Editor, though. Set material color of all textured objects to 1.0, 1.0, 1.0 (white) and correct the texture paths. Otherwise the textures won't show up in Condor.

V1 airport definition file can be used in V2 too. However, you will need to remodel at least the G file for each of the airports. Sorry.

As V2 thermal model differs from the V1, you need to adjust or completely rework the thermal map. See the related chapter of the tutorial.

***Voilà, your old scenery now should be ready for use in V2.***

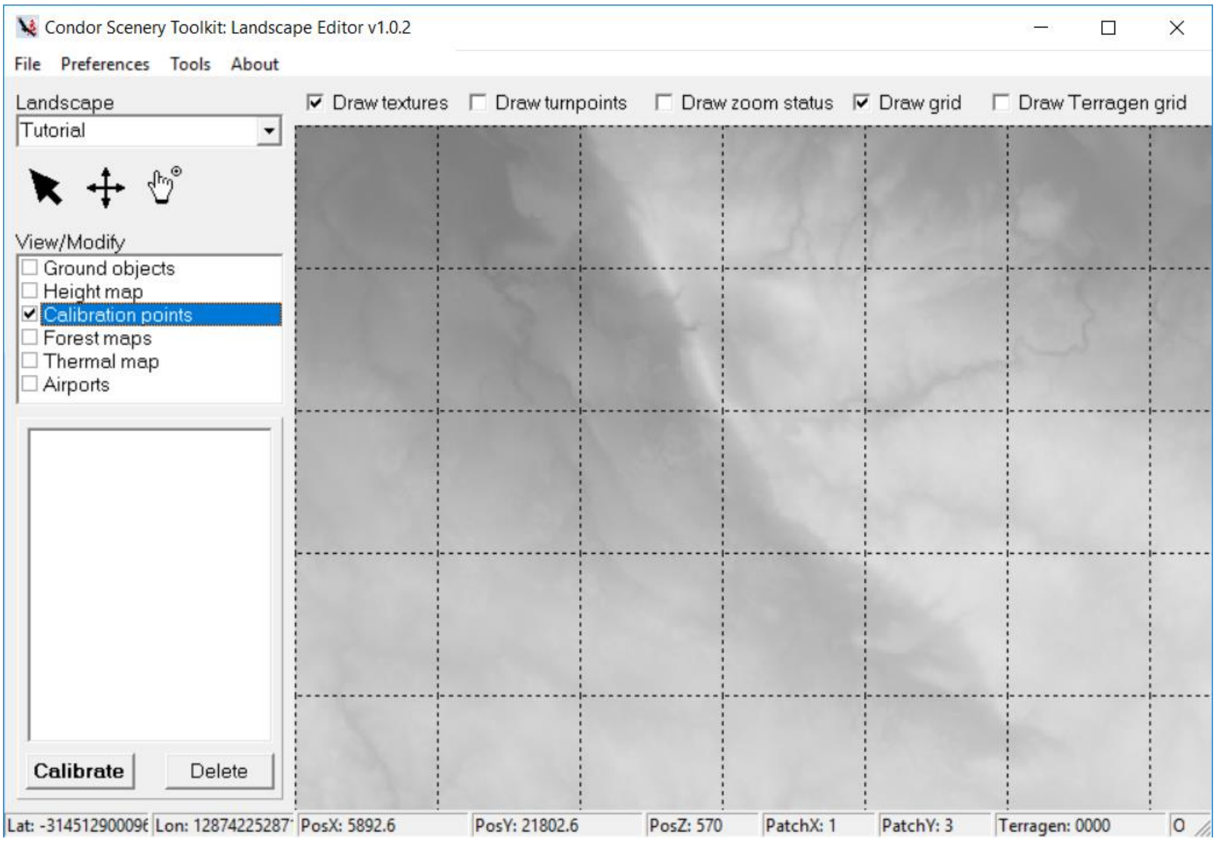
# 9. Appendix 1

## Manual terrain calibration

As the terrain is calibrated automatically during the export from **RawToTrn**, there is no need to do manual calibration in Landscape Editor anymore, unless you really need it for some reason.

If you have saved the CalibrationPoints.csv file in **UTMtools**, move this file to **Condor2\Landscapes\Tutorial\Working** folder (if the **\Working** folder is missing, create it). This will make calibration of the landscape much easier.

Tick the Calibration Points check box to display landscape calibration points. Click on the Calibration Points function to activate it. Notice that a list of calibration points appears under the toolbox. If you are using the CalibrationPoints.csv saved from UTMtools, you should now see four calibration points. If they are not there, check if the file is placed in the correct folder or if the name really is CalibrationPoints.csv. If you see the calibration points, you don't have to do the steps described in the few following paragraphs and you can skip to the point of using the Calibrate button.




If you are not using the csv file from UTMtools, you have two choices – you can either create this file yourself or calibrate the scenery manually.

For creating the CalibrationPoints.csv we need to know the lat/lon decimal coordinates of the corners and the size of the landscape in meters. When using all four scenery corners, the CalibrationPoints.csv has the following content (calibration points are placed clockwise starting from top left corner):

```
WidthMeters, HeightMeters, top left lat, top left lon
0, HeightMeters, top right lat, top left lon
0, 0, bottom right lat, bottom right lon
WidthMeters, 0, bottom left lat, bottom left lon
```

Decimal separator must be decimal point ("."). The content of the file for the tutorial example from this guide would then be as follows:

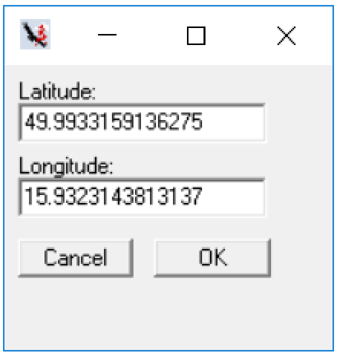
```
46080,46080,49.9967006118126,15.2894438698528
0,46080,49.9933159136275,15.9323143813137
0,0,49.5789090178296,15.924394186348
46080,0,49.5822447416054,15.2869845996996
```

Save the CalibrationPoints.csv with this content to the **\Tutorial\Working** folder and restart the Landscape Editor. After you open your Tutorial landscape again, you should see the calibration points in the list. You can then skip the following few paragraphs up to the step of using the Calibrate button.

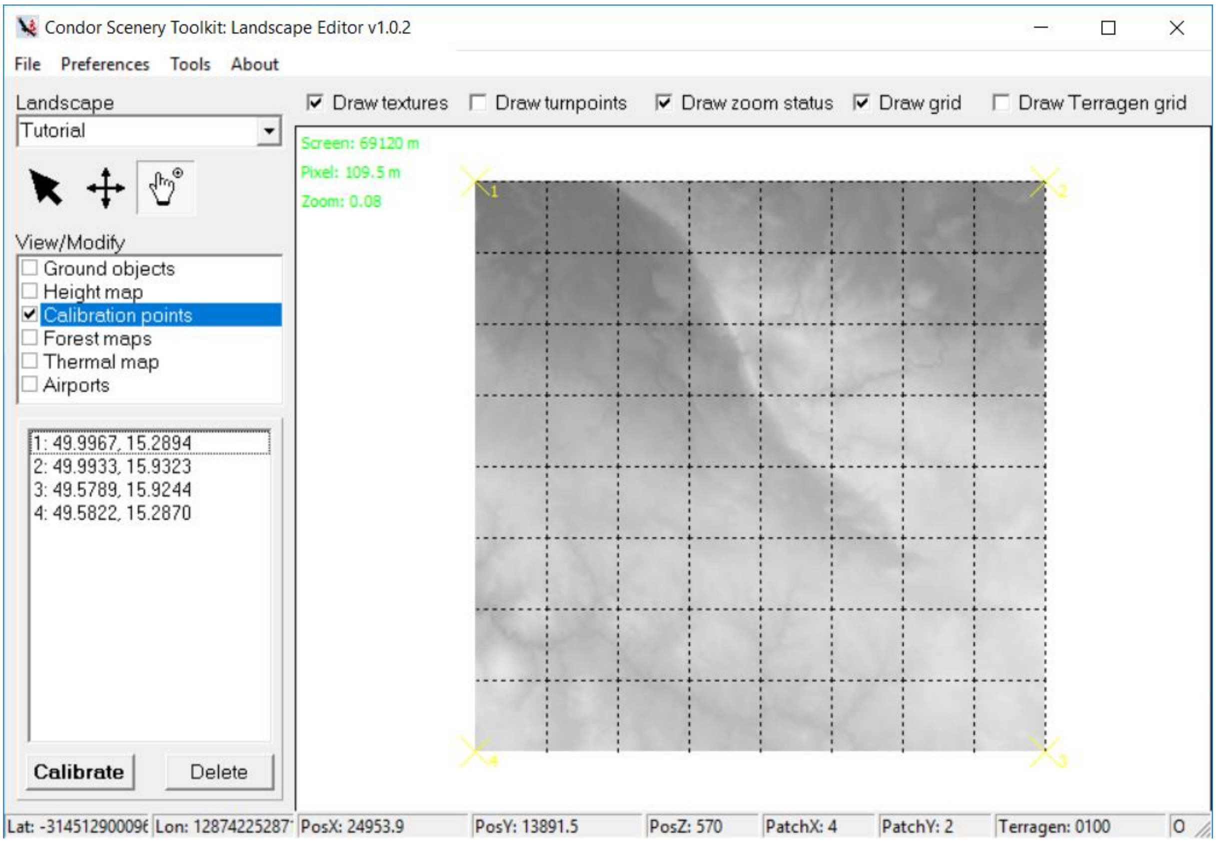
To do the calibration manually, without using CalibrationPoints.csv, we need to place calibration points as close to scenery corners as possible (ideally right to the corner). Click the hand icon to activate pan & zoom mode. We can move the terrain in the window by dragging it while holding the left mouse button. Left click zooms the terrain in and right click zooms out. Check Draw zoom status checkbox to see the current zoom level.

Let's start adding calibration points we have calculated previously. First zoom in on the left top of the terrain – keep left clicking until you see zoom level 42.67, which is the maximum zoom. Now click the arrow icon and move the mouse cursor as close to the corner of the terrain as possible. See the PosX and PosY indicators in the status bar. They tell you the current coordinates in meters from the origin point in the bottom right corner. As the size of our tutorial scenery is 46080 x 46080 meters, we aim to place the top left calibration point as close to the 46080, 46080 position as possible. When you think it is close enough (46080, 46080 in the ideal case), right click to open the context menu and then click Add calibration point here. A small dialog asking for the latitude and longitude coordinate pair opens. Either input the coordinates we have calculated earlier or copy the calibration coordinates from the UTMtools (double-clicking the value copies it to the clipboard).

Note that Landscape Editor only accepts dot as the decimal separator. If your local settings use a different decimal separator, you need to change it to dot before clicking OK in the dialog.




Using top left and bottom right corners should be enough to calibrate the terrain, but for our example we will add a calibration point in each of the corners.




Once all the calibration points are in place, we can click the **Calibrate** button. Landscape Editor will ask whether you really want to recalibrate the terrain. Click **OK** to start. We can immediately see the effect of calibration when moving the mouse cursor over the terrain – Lat and Lon indicators in the status bar now show decimal latitude and longitude of the point the cursor points at.