

# Getting started – iLand release 1.0

Rammer, W., Seidl, R., 2016-11-16

## ***Installing iLand***

The downloaded iLand package contains:

- The executable (iland.exe) and the required library files (Qt5\*.dll, icu\*54.dll, libEGL.dll, libGLESv2.dll, msvcp140.dll, d3compiler\_47.dll, vcruntime140.dll) for running iLand on Microsoft Windows (XP, Vista, Windows 7/8/10).
- An extensive example for running the model "out of the box" in the "example" folder.
- The full source code of iLand is packed in the "sourcecode.zip" file (see also <http://iland.boku.ac.at/publicsvn>)

To install iLand, simply extract the ZIP folder to a location of your choice und start the model by double clicking the "iland.exe" file. iLand comes without a dedicated installer and does not modify the Windows registry. It is thus also possible to run iLand from an USB stick.

iLand is written in C++ and uses the cross-platform toolkit Qt. Therefore iLand works also on Linux and Mac platforms. We have successfully compiled and employed iLand on a Linux machine (e.g., Ubuntu). Macs should work as well, but we actually never tried running iLand on a Mac yet (see also the building instructions below).

## ***Running iLand***

iLand organizes its inputs and outputs in so-called "projects". A "project" is a set of folders containing different kinds of data (e.g. climate data or species parameters) and a "project-file" (<http://iland.boku.ac.at/project+file>), which is a XML-file acting as the main controller for the various model options.

A basic user interface tutorial is available at the iLand wiki (<http://iland.boku.ac.at/iLand+viewer>)

In a nutshell, to run simulations with iLand you need to (see also Figure 1):

- Start the iLand viewer (iland.exe) (see also <http://iland.boku.ac.at/iLand+Software>)
- Select a project file (1)
- Create the model (2). This sets up the model landscape and initializes the trees.
- Run the model (3). Enter the number of years you want the model to run.
- While the model is running (there is also a "Pause" button), you can view different types of data in the GUI visualization:
  - "Light influence field": This is the "LIF" (<http://iland.boku.ac.at/competition+for+light>). Red means low, blue high light levels. Spatial resolution is a 2m grid.
  - "dominance grid": Maximum tree heights at 10m resolution
  - "Individual trees" shows tree crowns from bird's eye perspective (i.e. as stylized circles with the radius indicating the size of the tree). The color indicates their competitive status as described by the light resource index (LRI, again, red means

low LRs, i.e. high competitive pressure see <http://iland.boku.ac.at/competition+for+light>)

- “resource unit” show the resource units, i.e. a 100m grid used for the calculation of e.g. the water balance. (<http://iland.boku.ac.at/resource+units>)
- Clicking on a tree or a resource unit, respectively, shows their attributes in the “Data” box in the lower right corner. You can enter mathematical expressions (<http://iland.boku.ac.at/Expression>) using these variables in the „Expression“ field to visualize various aspects (e.g. “dbh” or “dbh\*dbh/4\*3.141592” to calculate basal area, see <http://iland.boku.ac.at/tree+variables> for available variables)
- The area for log messages (4) contains messages from the model. This kind of log info is especially helpful to track configuration or model errors.
- The bulk of numeric outputs of the model are stored in an output database (<http://iland.boku.ac.at/Outputs>). See <http://iland.boku.ac.at/Software+Tools> for tools helping to access the SQLite databases iLand uses to store simulation results (it is also quite easy to read these databases into, for instance, R for further analysis).

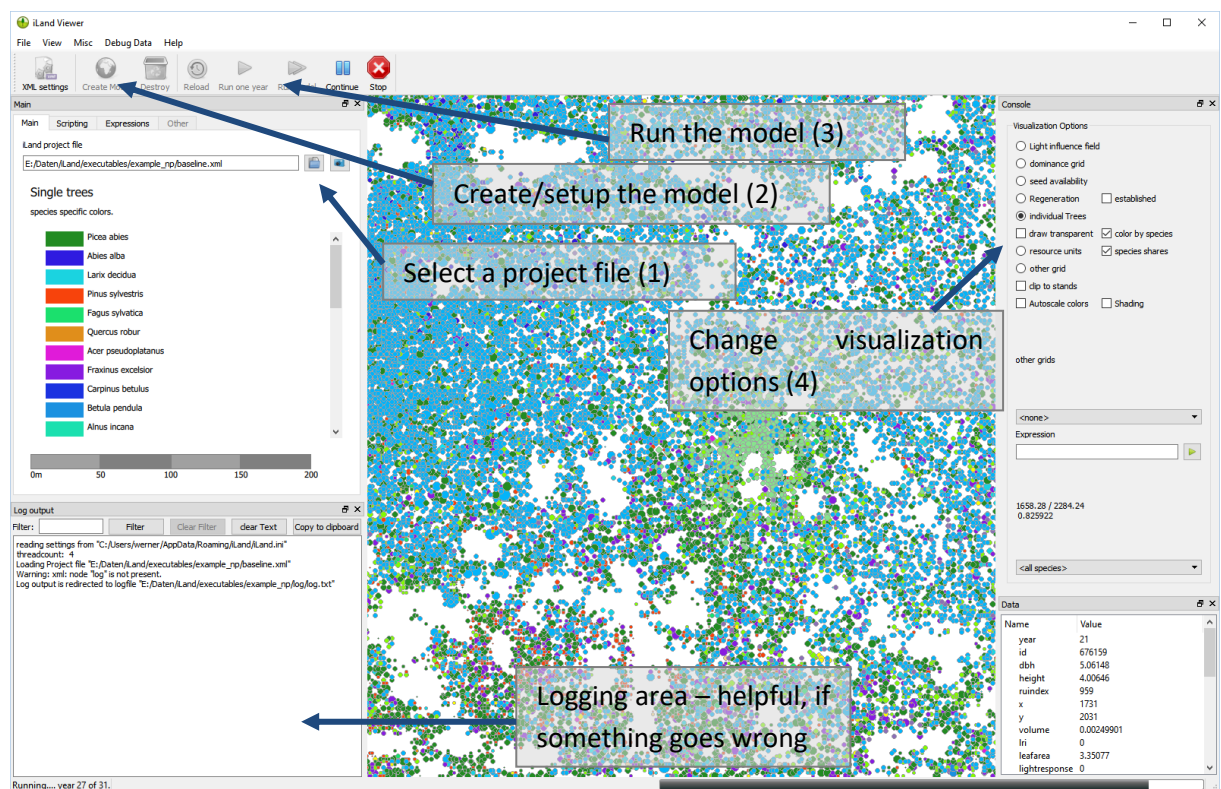


Figure 1. Main window of iLand at startup

## Building iLand from source

The download package also contains the full source code of the model as a ZIP archive “sourcecode.zip” (also available online at <http://iland.boku.ac.at/publicsvn>). In order to be able to modify and compile iLand, you will need Qt and the QtCreator IDE. Both packages are freely available

at <http://qt.io/>. The current release is compiled with Qt 5.6 for Windows (MSVC 64Bit), but iLand is known to work also with other compilers (Intel icc, msvc, gcc).

After installing the Qt Creator IDE, open the “iland.pro” project in the “src/iland” folder (be sure to preserve file paths when extracting the source code). iLand uses no external libraries (except the Qt libraries) which should simplify the building process considerably – a click to “build” should suffice (in principle). iLand is split in three projects: “plugins” (disturbance modules), “iland” (the model with graphical user interface), and “ilandc” (the console version of the model). Both iland and ilandc depend on plugins.

You might need to check the build paths in Qt Creator (Project settings, build path): use a common build-folder (e.g., “c:/iland/build”) and build the subprojects (“plugins”, “iland”, “ilandc”) to the folders “<build-folder>/plugins”, “<build-folder>/iland”, “<build-folder>/ilandc” (e.g. c:/iland/build/iland). The compiled executable file will then be e.g. c:/build/iland/release/iland.exe. Make sure to also compile all sub projects with the same compiler version.

## ***The example project***

The example supplied with the model is meant for illustrative purposes only, to allow you to conduct first runs "out of the box", to get a feel of how the model works, and have an example at hand of how the input data look like. The data supplied here are for a 5x3 km part of the Kalkalpen National Park, situated in the Northern Front Range of the Eastern Alps in Austria (i.e., not the whole National Park is covered by the example). The provided vegetation depicts the state of the forest in the year 1999, as reconstructed from terrestrial inventory data (for details see Thom et al. 2016, J. Appl. Ecol. doi: 10.1111/1365-2664.12644; but note that the vegetation data provided with the example is simplified). The climate consists of 18 distinct climate zones, corresponding to the strong environmental gradients in the mountainous landscape. The provided climate time series is for the years 1950 – 2100. Soil data were compiled from a spatially explicit analysis of soils in the park in combination with information from a large scale forest soil inventory. The provided species parameter set contains 31 Central European tree species. These parameters have been tested against independent data from forests across the Eastern Alps and have been proven to work well for the current landscape (Thom et al. 2016, J. Appl. Ecol. doi: 10.1111/1365-2664.12644). Good evaluation data were, however, not available for all species, and particularly species with current low abundance in the region were thus mainly evaluated with regard to their relative to other species.

The simulation is set up not to include management, but has wind and bark beetle disturbances enabled. Wind speed, direction, duration, and return interval have been parameterized based on past events (for details see Seidl and Rammer 2016, Landscape Ecol. doi: 10.1007/s10980-016-0396-4). Likewise, the parameters for the bark beetle module (here simulating *Ips typographus* L.) have been estimated based on empirical evidence for the species gathered in Central Europe (see the Online Supplement of Seidl and Rammer 2016, Landscape Ecol. doi: 10.1007/s10980-016-0396-4). In the simulations, a continuation of future climate is assumed by resampling years randomly from the provided climate time series.

Out of the box the model output database will only record landscape level averages per tree species, but this can be extended easily by editing the "output" section of the project file (e.g., to include stand-level information (1ha aggregates), data on the forest carbon cycle (1 ha aggregates), and more). For more details on the example project we refer to Thom et al. (2016, J. Appl. Ecol. doi: 10.1111/1365-2664.12644). Please cite this study if you should use any part of the supplied example in your work (e.g., parameters etc.), and also acknowledge the original data providers (e.g., with regard to soil, climate, etc.) as appropriate.

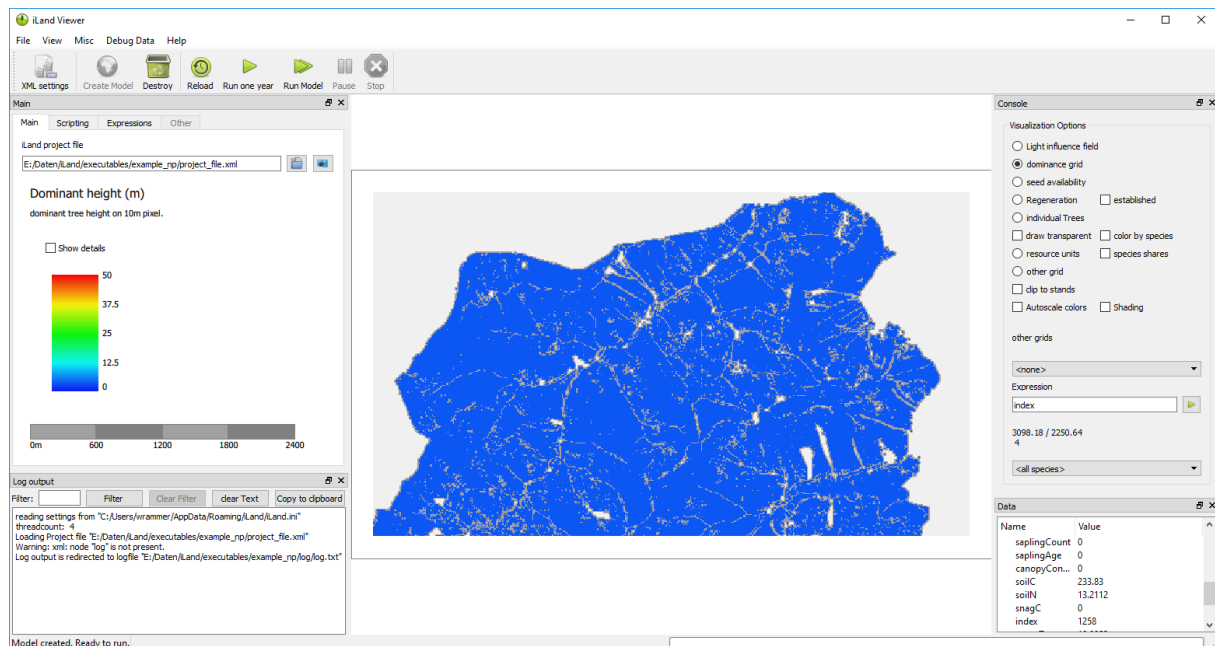


Figure 2. The iLand viewer after creating the demo project (bare ground).

By modifying (or copying and modifying) the project-file, the user can easily change the model input data and the model behavior. Some simple options are listed below.

## Current forest state or bare ground

You can initialize a “current” state of the forest, or you can start from “bare ground”. To do this, set *model.initialization.file* to either “stand\_init.txt” or “empty\_trees.csv”, and *model.initialization.saplingFile* to either “sap\_init.txt” or “empty\_saplings.csv”. In bare ground conditions, seed input comes from the area surrounding the project area (see *model.settings.seedDispersal*).

## Current climate or climate change

The “baseline.xml” project file represents a current or reference climate. The climate data (ARPEGE.sqlite) includes time series from 1951-2100 for each available climate region. A baseline climate is created by sampling repeatedly years from this time series (e.g. from 1950-2000). See *model.climate.randomSamplingList*. You can simulate climate change either by turning off the random sampling (*randomSamplingEnabled*) in conjunction with *filter*, or construct an own sampling list (see also *batchYears*). A changing CO<sub>2</sub>-concentration can be easily enabled with the time-events mechanism (see <http://iland.boku.ac.at/time+events> and the “dist\_cc\_time.txt” file). For details on the used climate data see Thom et al. 2016, J. Appl. Ecol. doi: 10.1111/1365-2664.12644. Note that this example covers only a part of the study area in Thom et al (18 instead of 200 clusters).

## Enable/disable disturbances

Switch disturbances on/off by setting *modules.barkbeetle.enabled* and *modules.wind.enabled* to either *true* or *false*.