TREES4FUTURE - Designing Trees for the Future

Start date of project: 1 November 2011
Duration of project: 4 years
Call: FP7-INFRASTRUCTURES-2011-1
Theme: INFRA-2011.1.1.13 [Research infrastructures for forestry research]
Funding Scheme: Combination of CP & CSA

D10.4 -
Description and manual of open structure of models

Description: This documents the infrastructure of the EFISCEN, ORCHIDEE and ToSIA models, how they can be used, and results made accessible.

Due date of deliverable: M54
Actual submission date: M54
Organisation name of lead contractor for this deliverable: EFI


<table>
<thead>
<tr>
<th>Dissemination level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU Public (must be available on the website)</td>
<td>x</td>
</tr>
<tr>
<td>PP Restricted to other programme participants (including the Commission Services)</td>
<td></td>
</tr>
<tr>
<td>RE Restricted to a group specified by the consortium (including the Commission Services)</td>
<td></td>
</tr>
<tr>
<td>CO Confidential, only for members of the consortium (including the Commission Services)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes between different revision versions</th>
<th>Distribution (who have reviewed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.4.2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>29.4.2016</td>
<td>Reviewer comments addressed</td>
<td>Duncan Ray and Rob Lokers</td>
</tr>
</tbody>
</table>
# Table of Contents

**Objectives** .................................................................................................................. 4  
**Section 1: EFISCEM manual** .......................................................................................... 5  
  1. Introduction .................................................................................................................... 5  
  2. Model overview ............................................................................................................. 6  
  3. User guide for conducting simulations ......................................................................... 9  
    3.1 Introduction ................................................................................................................. 9  
    3.2 Requirements ............................................................................................................ 9  
    3.3 Files and directories ................................................................................................. 9  
    3.4 Model inputs ............................................................................................................ 10  
      3.4.1 Conventions ...................................................................................................... 10  
      3.4.2 Forest resource input files .............................................................................. 11  
      3.4.3 Scenario input files ......................................................................................... 20  
    3.5 Model simulations .................................................................................................... 26  
      3.5.1 Graphical user interface .................................................................................. 26  
      3.5.2 Command line ................................................................................................... 30  
    3.6 Model outputs .......................................................................................................... 34  
      3.6.1 Conventions ...................................................................................................... 34  
      3.6.2 Output as text files ......................................................................................... 35  
      3.6.3 Output database ............................................................................................... 39  
      3.6.4 Error logs ........................................................................................................... 40  
  4. User guide disaggregation tool ..................................................................................... 42  
    4.1 Introduction ................................................................................................................. 42  
    4.2 Requirements ............................................................................................................ 42  
      4.2.1 System requirements ....................................................................................... 42  
      4.2.2 Software requirements ..................................................................................... 42  
      5.2.2 Database drivers ............................................................................................... 43  
    4.3 Input files .................................................................................................................. 43  
      4.3.1 Output database ............................................................................................... 43  
      4.3.2 Map linkage ...................................................................................................... 44  
      4.3.3 Configuration file ............................................................................................. 45  
      4.3.4 Region raster .................................................................................................... 45  
      4.3.5 Species rasters .................................................................................................. 45  
    4.4 Tool execution ........................................................................................................... 46  
    4.5 Output ...................................................................................................................... 49  
  5 Results availability .......................................................................................................... 49  
  6 Final remarks ................................................................................................................ 50  
  7 References ..................................................................................................................... 51  
**Section 2: ORCHIDEE manual** .................................................................................... 52  
  1. Set-ups ......................................................................................................................... 52  
  2. Modelled processes ...................................................................................................... 53  
  3. Input data ...................................................................................................................... 54  
  4. Spatial conceptualization .............................................................................................. 56  
  5. Temporal conceptualization .......................................................................................... 58  
  6. ORCHIDEE for TREES4FUTURE ............................................................................... 59  
  7. Practical information for new users ............................................................................. 59  
    7.1 General information ............................................................................................... 59  
    7.2 The model code ...................................................................................................... 60  
    7.3 The run environment ............................................................................................... 60
7.4 Parameter settings ........................................................................................................ 62
7.5 GUI & support ............................................................................................................... 64
8. Result availability ............................................................................................................. 64
9. References ....................................................................................................................... 64
Section 3: ToSIA ..................................................................................................................... 65
1. Introduction ..................................................................................................................... 65
1.1 ToSIA - Tool for Sustainability Impact Assessment ................................................... 65
1.2 Existing documentation and changes ........................................................................... 66
1.2.1 General documentation ......................................................................................... 66
1.2.2 Calculation aids and partial models ...................................................................... 66
1.2.3 Handbooks and Manuals ...................................................................................... 66
1.3 New developments ....................................................................................................... 67
2. JAVE: new ToSIA Data Client ......................................................................................... 69
2.1 Login ............................................................................................................................ 69
2.2 Navigation panel ......................................................................................................... 70
2.2.1 Chain view ............................................................................................................. 70
2.2.2 Process view .......................................................................................................... 73
2.2.3 Product view ......................................................................................................... 74
2.2.4 Catalogue editor .................................................................................................... 75
2.2.5 User administration .............................................................................................. 79
3. ToSIA - EFISCEN linkage ............................................................................................... 80
4. References ....................................................................................................................... 81
Glossary

EFISCEN: empirical forest resource model
ORCHIDEE: process-based ecophysiological land surface model
ToSIA: tool for sustainability impact assessment
Objectives

The overall aim of WP10 is to integrate aspects of genetics and wood quality into large scale modelling tools, and to improve the infrastructure and compatibility of these tools to assess goods and services, sustainability, and mitigation and adaptation strategies for European forests. Sub objectives are:

- Assess requirements for compatibility and exchange of data between genetics and wood quality on the one hand and large scale assessment models on the other hand.
- Improve infrastructure and performance of three existing European models to ensure compatibility between models and the databases compiled by the current project.
- Develop an open and user-friendly structure supporting the use of these models by the wider European research community to assess goods and services, sustainability, and adaptation and mitigation capacity of European forests.

This deliverable report contains manuals for the EFISCEN, ORCHIDEE and ToSIA models describing how they can be used, and results made accessible. The report contains one section with a manual for each model. Model descriptions and manuals exist already for previous versions of all three models and are not repeated here. In this deliverable the focus is on improvements made to these models within the Trees4Future project.

This deliverable builds on three other deliverable reports in WP10, which documented the workplan to link models (D10.1), the implementation of improvements in larger scale tools ORCHIDEE, EFISCEN and ToSIA (D10.2) and how the models can be used in a coordinated and combined way to inform decision making with regards to goods and services, sustainability, adaptation and mitigation (D10.3).
Section 1: EFISCEN manual

1. Introduction

European forests are a crucial resource to supply a growing bio-economy, they provide multiple ecosystem services and they can and mitigate the effects of climate change. To assess how European forest can accommodate multiple demands, simulation models are useful tools. The European Forest Information SCENario model (EFISCEN) has been used for over two decades to provide insight in European forest resource development (e.g. Böttcher et al. 2012; Eggers et al. 2008; Hanewinkel et al. 2013; Karjalainen et al. 2003; Nabuurs et al. 2000, 2001, 2007; Schelhaas et al. 2010, 2015; Verkerk et al. 2011a,b, 2014). A full publication overview is available at: http://efiscen.efi.int.

EFISCEN is a large-scale forest model that projects forest resource development on regional to European scale. The model uses national forest inventory data as a main source of input to describe the current structure and composition of European forest resources. The model can project the development of forest resources, based on different scenarios. These scenarios are mainly determined by management actions, but the model can also consider changes in forest area and growth rates. EFISCEN provides data on basic forest inventory data (species, area, stemwood volume, increment, mortality, age-structure), but the model includes multiple indicators related to important forest ecosystem services (carbon sequestration, biodiversity, recreation, wind and fire risk), enabling the assessment of impacts of different policy and management strategies at the national and European level, thus serving forest managers and policy makers at the national and international levels.

The core of the EFISCEN model was developed in the late 1980s for Sweden by Prof. Ola Sallnäs at the Swedish Agricultural University (Sallnäs 1990). The first European application of this model was carried out by the International Institute for Applied Systems Analysis (IIASA) in the early 1990s (Nilsson et al. 1992). With help from the original developers, the model was transferred to EFI in 1996, and given the name EFISCEN. The model was developed further both by EFI and Alterra, resulting in EFISCEN 2.0 (Pussinen et al. 2001). Development of the model continued and the model was then re-programmed into C++ code and a user interface was added. This version was called EFISCEN 3.0. The EFISCEN model has been described in detail by Schelhaas et al. (2007) for EFISCEN version 3.1.3.

Within the Trees4Future project, a decision was made to develop an open and user-friendly structure supporting the use of the model by the wider European research community. For that purpose, EFISCEN was re-implemented from C++ to Java. The process to re-implement and improve EFISCEN was started by EFI in 2011 with participation by the University of Eastern Finland and Alterra. Besides re-implementation, model functionality has been extended as well. Functionality was added to improve the graphical user interface, as well as to make the model more flexible for use in various scenarios. In addition, the whole code was verified and checked, including documentation on model structure and testing.

This manual aims to describe how to initialise and apply EFISCEN 4.1. Chapter 2 contains a user guide for EFISCEN 4.1 and chapter 3 contains a user guide for the newly developed spatial disaggregation tool. For a detailed description of the model, its history and steps needed to prepare input files, we refer to Schelhaas et al. (2007) as these did not change between EFISCEN 3.1.3 and EFISCEN 4.1.
2. Model overview

**Acronym and name of the model**
EFISCEN (European Forest Information Scenario model)

**Website**
- [http://efiscen.efi.int](http://efiscen.efi.int)

**Organisation(s) involved in the development of the model**
EFISCEN has been jointly developed and applied at the European Forest Institute (EFI) and Alterra (part of Wageningen UR).

**Contact persons**
- Dr. Hans Verkerk (EFI): [hans.verkerk(at)efi.int](mailto:hans.verkerk(at)efi.int)
- Prof Dr. Gert-Jan Nabuurs (Alterra): [gert-jan.nabuurs(at)wur.nl](mailto:gert-jan.nabuurs(at)wur.nl)
- General email address: [efiscen(at)efi.int](mailto:efiscen(at)efi.int)

**Source code**
The Java source code of the model will be available on GitHub ([https://github.com/EuropeanForestInstitute/efiscen](https://github.com/EuropeanForestInstitute/efiscen)) in spring 2016 and is distributed under the GNU General Public License (version 3) conditions ([www.gnu.org/licenses/gpl-3.0.html](http://www.gnu.org/licenses/gpl-3.0.html)).

**History and development**
The core of the EFISCEN model was developed in the late 1980s for Sweden by Prof. Ola Sallnäs at the Swedish Agricultural University. The first European application of this model was carried out by the International Institute for Applied Systems Analysis in the early 1990s. With help from the original developers, the model was transferred to EFI in 1996, and given the name EFISCEN. The model was developed further both by EFI and Alterra. The current version is EFISCEN 4.1.

**Architecture and modules**
The main EFISCEN model performs the simulations. Within the EFISCEN model we can distinguish 1) the matrix simulator, 2) a carbon module to convert outputs to carbon stocks and 3) a soil module based on the YASSO soil model (Liski et al. 2005). EFISCEN 4.1 is implemented in Java. To prepare required input files, a separate program (P-2009) can be used to initialise the matrices based on the forest inventory input data. A tool is available to spatially disaggregate EFISCEN results to raster maps.

**Concept**
EFISCEN is a large-scale forest model that projects forest resource development on regional to European scale. The model uses national forest inventory data as a main source of input to describe the current structure and composition of European forest resources. The model can project the development of forest resources, based on scenarios for policy, management strategies and climate change impacts. EFISCEN provides insights in large-scale forest resource development, wood and biomass production, biodiversity and ecosystem services, thereby serving forest managers and policy makers at the national and international levels. A detailed model description is given by Schelhaas et al. (2007) and a schematic overview is shown in [Error! Reference source not found..1](#).
EFISCEN is a matrix model in which the state of the forest is described as an area distribution over age and volume classes, based on data on area, growing stock and increment by age class and forest type collected from national forest inventories. For each forest type that is distinguished in the input data (which might be according to species, region, site class and owner), a separate matrix is set up. During simulations, forest area moves between matrix cells, describing different natural processes (e.g. growth and mortality) and human actions (e.g. forest management). Aging of the forest is simulated by moving area to a higher age class, while growth is simulated by moving the area to a higher volume class. Transitions can be changed over time to simulate changes in growing conditions (e.g., due to climate change).

Management scenarios are specified at two levels in the model. A basic management regime defines the period during which thinnings can take place and a minimum age for final fellings. These regimes can be regarded as constraints on the total harvest level. The demand for wood is specified and EFISCEN may fell the demanded wood volume if available. Wood demand is the main determinant of forest resource use. Thinning in the model is simulated by moving area one volume class down. Only area that was moving to a higher volume class (increment) can be subjected to thinning. The user can specify an age range where thinnings can be carried out. If a thinning will be carried out or not depends on the actual demand for thinning. A user-defined fraction of the area that has been subjected to a thinning will be moved up one volume class extra to simulate the growth response after a thinning. Final fellings are simulated by taking the area out of a certain cell. Final felling chances can be set by the user as a function of age and volume class. The fraction that is actually harvested depends on the actual demand for wood from final fellings. Area that is taken out of the matrix is put in a separate class, the non-stocked area. Regeneration is simulated as the movement from the non-stocked area into the lowest age and volume class of the matrix. Natural mortality is simulated by moving a fraction of the area in a certain cell one volume class down. This fraction can be set by the user as a percentage of the growing stock, varying by age class. The actual fraction of the area that is moved down will then depend on the average volume before, and the difference between the volume classes. Only area that has not recently been thinned can be subjected to natural mortality.
Based on the information mentioned above, EFISCEN projects stem wood volume, increment, age-class distribution, removals, forest area, natural mortality and deadwood for every five-year time-step. With the help of biomass expansion factors, stem wood volume is converted into whole-tree biomass and subsequently to whole tree carbon stocks. Information on litterfall rates, felling residues and natural mortality is used as input into the soil module YASSO (Liski et al. 2005), which is dynamically linked to EFISCEN and delivers information on forest soil carbon stocks.

**Model input and parameters**
The forest area under study is usually separated into forest types, which can be separated based on administrative unit, ownership, tree species and site class. To initialise the model, EFISCEN needs the area, average growing stock volume and net annual increment per age class of each forest type. If the user applies gross annual increment, data about natural mortality per age class is required. Furthermore, information needs to be available on the thinning and final felling regime. To convert growing stock volume into estimates of carbon in total tree biomass, the user needs to provide the model with biomass distribution functions. Additionally, the model can simulate carbon dynamics in the soil via its soil model YASSO. This requires data on turnover rates of different biomass components, litter quality, and climate parameters.

**Output**
EFISCEN provides estimates of forest area, growing stock, increment, standing dead wood, harvest level and age class distribution over time. These are provided on different aggregation levels (e.g. per species, regions, total). Furthermore, the model can provide information on carbon stocks in biomass and soil, deadwood (an indicator for biodiversity), risk indicators and ecosystem services.

**Validation**
EFISCEN has been validated by comparing its growth functions against growth functions of other models, by comparing projections against projections of other models, and by running the model on historic data for various regions in Europe. A sensitivity analysis has been carried out.

**Strengths and limitations of the model**
EFISCEN is designed for large forest areas (e.g. provinces or countries). Application to smaller areas is possible, but there have been no studies yet to determine the minimum size and effects of scale on uncertainty of the projections. Generally, several thousand hectares is considered a safe minimum.

EFISCEN has been developed for even-aged, managed forests. Deviations from this situation (e.g. uneven-aged or unmanaged forests and shelterwood systems) make the application of EFISCEN less suitable. Furthermore, the model is currently not suited to simulate fast growing tree species with very short rotations, due to the 5-year time step. The model can handle small decreases in forest area, but is not suited to deal with large-scale deforestation issues.

As with all models, uncertainties in EFISCEN depend largely on the quality of the input data. Especially a correct estimation of the increment functions is important for the model outcomes. Initial uncertainties propagate through the model with every simulated time step, and thus the overall uncertainty increases. For 10-12 time steps (50-60 years) the model is considered to give reasonable projections. With increasing projection length, observed patterns become more important than absolute values.
3. User guide for conducting simulations

3.1 Introduction
EFISCEN version 3.1.3 has been previously described in detail by Schelhaas et al. (2007). While the general concept of the model and matrix initialisation procedure has not changed, the procedure to carry out simulations has been modified substantially in EFISCEN version 4.1. Key improvements in EFISCEN 4.1 as compared to EFISCEN 3.1.3 and new features are listed here:

- Graphical User Interface (GUI) has been redesigned with additional functionality. Total and selected data is now divided to two tabs to make the interface less crowded;
- In the GUI, the user can now select multiple regions, owners, sites or species in the selection tree and data are shown in the selected-tab accordingly;
- All simulation values that are shown in the panels can also be shown in the graphs with their history. Variables to show as graphs can now be easily toggled by clicking corresponding buttons;
- Error messages have been made more descriptive, three sets of log files are provided describing events and errors are written and fault tolerance of the program has been improved;
- Time step specification in scenario files has been changed so the last parameter values are used when last step in scenario file is reached, rather than re-running the whole scenario;
- Area scaling factors can now be applied to individual matrices through a newly added scenario file;
- Management parameters that were previously fixed through the simulation can now change. They are defined in two new scenario files. This feature allows the user to modify management regimes along the course of the simulation;
- Scenario file format has been extended to include individual matrix scaling and thinning and felling change files;
- All output files are now written in .csv format;
- It is possible for the user to specify a list of output files that need to be created;
- Outputs can directly be saved to an external database. Databases that are currently supported are MySQL, PostgreSQL and Microsoft Access.

All steps needed to carry out simulations are described in the sections below.

3.2 Requirements
The minimum requirements to run EFISCEN are Java version 8 update 40.

3.3 Files and directories
Figure 3.1 gives an overview of the file structure of EFISCEN 4.1. The files in the upper half of the figure are parameter files, while the files in the lower half of the figure are scenario files.
3.4 Model inputs

3.4.1 Conventions

In this description, the symbol *** is used to denote the country name. The naming of the files is flexible, since files are either selected by the user, or the file names to be used are listed in other files. This gives the user the opportunity to distinguish different versions of certain files by using different names (for example with scenario files).

In any input file, the hash symbol (#) can be used for inserting comments. The program will not read any lines that start with #. To separate items in a line, spaces should be used (or in many scenario files commas), but not tabs.

When several items of the same kind are listed in an input file, a number indicating how many items there are must precede the item list. This is both valid for matrices/forest types (for example the number of input matrices in an input file) as well as within lines (for example how many parameters are listed to define the final felling regime).

Forest types are always identified by four digits, representing respectively the region, owner class, site class and tree species. Many parameter values can be set separately for a forest type, or for a class of forest types. Here the same identification system is used. A zero can be used to include all forest types of a certain class. For example, 2 1 0 3 selects the matrix of the second region, first owner class, all site classes and the third tree species, as they are defined in the ***.efs file. Similarly, all matrices can be selected by listing 0 0 0 0. However, be aware that for the definition of the matrices (***.aer and ***.vcl files) every matrix should be defined separately, so here no zeros as index are allowed.
For many scenario files, input is required for each time step. If the number of time steps is longer than the length of the scenario, EFISCEN will assume start from the beginning of the scenario again. This is valid for all scenario files, so for example for the required harvest and for the soil climate file. If for example the soil climate is not defined until the end of the simulation, it will start over again with the first defined climate.

Scenario files contain lines starting with a time step number. As a convention, the values in this line are valid until (and including) the specified time step. In case simulations are continued beyond the latest time-step specified EFISCEN 3.1.3 started the loop from the beginning when no more parameters were defined. In EFISCEN 4.1 this has been changed so the last parameter values are used when no more values are defined.

### 3.4.2 Forest resource input files

#### ***.efs

The initialisation file defines the base year for the start of the simulation, i.e. the (mean) year of the forest inventory. Furthermore, the forest types are defined, in accordance with the matrix setup. For each category (region, owner class, etc.), first the number of classes that is distinguished is defined, followed by the definition of those classes (region names, owner names, etc.). For mapping purposes, a regional identification number can be defined, but this is not obligatory. The ID number is the ISO country code times 1000 plus the number of the region. ISO country codes can be for example found at [http://unstats.un.org/unsd/methods/m49/m49alpha.htm](http://unstats.un.org/unsd/methods/m49/m49alpha.htm). Furthermore, the names and locations of the parameters file, the biomass allocation file, the matrix file and the soil parameters file are defined. The initialisation file should have the ending .efs. Finally, values need to be separated using spaces, not tabs or commas. An example for Czech Republic is given in Box 3.1.

```
EFISCEN experiment file
#Experiment's initialisation file
#EFISCEN - Czech Republic
Czech Republic
#Base year (starting simulation)
2000
#Regions should be listed first, started from how many
14
1 203001 BRENENSKY
2 203002 BUDEJOVICKY
3 203003 JIHLAVSKY
4 203004 KARLOVARSKY
5 203005 KRAJLOVRADECKY
6 203006 LIBERECKY
7 203007 OLOMUICKY
8 203008 OSTRAVSKY
9 203009 PARDUBICKY
10 203010 PLZENSKY
11 203011 PRAHA
12 203012 STREDOCESKY
13 203013 USTECKY
14 203014 ZLINSKY
#Owners
1
1 ALL
#2 Private
#Sites
```
### Box 3.1: Example of a country initialisation file for the Czech Republic (czech.efs).

```plaintext
1
1 ALL
#Species
10
1 Spruce
2 Fir
3 Pine
4 Larch
5 Other_Conifers
6 Oak
7 Beech
8 Maple
9 Ash
10 Other_broadleaves
#File name for parameters
Czech.prs
#
#File name for bioparameters
biocomp.txt
#File name for matrixes
cze.aer
#
#File name for soils
soilcze.par
#END
```

The parameters file defines all tree-related parameters needed for the simulation, including age class size and number, coefficients for the growth functions, age classes for thinnings and final fellings as well as the optimal volume per age class. An example is given in Box 3.2 for Czech Republic. The time step for simulation defines the 5-year time step that is usually applied. Other time steps could be needed in different forest types (like fast growing plantations), but that has not been tested in this version yet.

The number of age classes can be taken from the input data, and should correspond with the number of age classes in the ***.aer file. The lines that define the size of the age and volume classes are not in use.

The growth function is defined by the three parameters $a_0$, $a_1$, and $a_2$ (see Equation 5 in Schelhaas et al. 2007). Optionally, confidence intervals can be added, defining age limits for the application of the growth function. In that case, the minimum and maximum age must be given in addition to the growth function coefficients (see comment lines in example below).

To define the final felling regime, the user can choose one out of two options: (1) giving the minimum age for final fellings; after reaching that age, all forest will be available for final felling or (2) to define the minimum age and the corresponding felling probability, the age at which the felling probability will reach 100%, and the felling probability for forest younger than the minimum age. In the example in Box 3.2, the felling probability for the forest type is 10% for 81-year-old forests and 100% for 120-year-old forest. Between 81 and 120 years, linear interpolation is applied. Forests aged 65 to 81 can be cut with a felling probability of 1.5% (i.e. 1.5% of the forest in the corresponding matrix cells can be submitted to final fellings). For the thinning regime, the minimum and maximum age for thinnings is defined. So for the same forest type thinnings can be carried out when the
forest is between 20 and 80 years old. In case the age range of thinnings and final fellings overlap, part of the final fellings will not be found even when there is enough removal volume available. The reason is that the model first calculates! on which fraction of the available area thinnings and final fellings should be carried out, without taking into account the overlap. An area cannot be subjected to thinning and final felling in the same time step. Since thinnings are carried out first, less area is available for final fellings. However, the fraction to be subjected to final felling is not adapted to this change in available area. It is recommended not to have overlap between the thinning and felling range.

The volume series define the optimal volume per age class. This is difficult to determine, usually the values from the input data are copied. Here, first the age class limits are defined for which the volume series are valid (AgeLims). These age class limits also define the age classes for the initial matrices, so they should also match the input data. Note that the maximum age defined here is used to define the upper limit of the age dimension in the matrix.

Mortality can be defined by forest type and age class. In the example in Box 3.2, 2% of the volume in forests up to 80 years will die due to natural mortality each time step. This 2% is converted into area transitions, depending on the average growing stock volume per volume class. The corresponding volume will move to a deadwood pool; it is assumed that the trees remain standing for a while. The deadwood volume fall rate defines the proportion of the standing deadwood pool that moves to the coarse woody litter pool of the soil sub-module each time step (4% in the example below). This rate reflects not only whole trees falling down, but also stem pieces falling off. No mortality takes place when the mortality rates are set to zero. Note that the rate of volume mortality is converted in an area transition. However, currently it is only possible to move the area one volume class down. Therefore, the highest possible mortality rate in the upper volume class is 10%, assuming equal volume classes. Even though it is possible to enter higher values, the actual mortality rate will be limited by the volume class width.

The thinning history parameter (Thhistory) defines the share of the area within the possible thinning range that is not available for thinnings, because of a recent thinning. After this area has received the growth boost, it will be available for thinnings again. Finally, values need to be separated using spaces, not tabs or commas. An example for Czech Republic is given in Box 3.2.

```
# Experiment's parameters file
# Czech Republic
# Step of simulation (how many years are in one tick)
5
# For all parameters which can depend on Reg:Own:Site:Spec
# combination - corresponding IDs could be given (0 = means for all)
# Then size of array then array itself
# For all next name_of_parameter and n_howmany
#
# Number of age classes (X axis)
AgeClassNum 1
0 0 0 0
1 16
# Size of age class (X axis)
X1 1
0 0 0 0
1 10
```
#Number of volume classes (Y axis)
VolClassNum 1
0 0 0 0
1 10
#size of volume class (Y axis)
Y1 1
0 0 0 0
1 50.
#
#Growing function's coeff.
GrFunction 10
0 0 0 1
5 -12.87316907 2113.657035 -7729.068793 10 150
0 0 0 2
5 -10.91577278 1930.812189 -7084.449633 10 150
0 0 0 3
5 -7.540788132 1677.718089 -5760.735426 10 150
(...)
#Young forest coeff
YForest 10
0 0 0 1
1 0.7
0 0 0 2
1 0.7
0 0 0 3
1 0.7
(...)#Regrow after thinnings
Gamma 1
0 0 0 0
1 0.4
#Age of Harvest
#for simplest regimes we provide only one number - age of cutting
#in other case we provide 6 values
#min_age max_age min_tresh max_tresh level_below starting_age_below
Harvest 10
0 0 0 1
6 81 120 0.1 1 0.015 0.2
0 0 0 2
6 96 150 0.1 1 0.015 0.2
0 0 0 3
6 96 120 0.1 1 0.015 0.2
(...)
#Thinnings range
Thinrange 10
0 0 0 1
2 20 80
0 0 0 2
2 20 95
0 0 0 3
2 20 95
(...)#Beta coeff
Beta 1
0 0 0 0
1 0.4
#Volume series: pair - first age classes limits; second volumes
#again IDs should be first
AgeLims 1
Box 3.2. Example of parameter input file for Czech Republic (Czech.prs). Note that only 3 forest types are shown for management regimes, young forest coefficient and optimal volume series. Data left out is indicated by (...).

Biocomp-***.txt
This file defines the parameters for carbon content, dry wood density, biomass allocation, and litter production. Each of these can be defined by region, species, owner and site class. Biomass allocation and litter production are age-specific and have to be defined for five tree compartments: stem, branches, coarse roots, fine roots and foliage. Biomass allocation values are shares of the total tree biomass and should add up to one. For example, in Box 3.3, in all spruce forest types (indicated by 0 0 0 1) the share of the stem in the total biomass is 38.52% in forests up to 30 years old. Branches account for 34.87%, coarse roots for 4.96%, fine roots for 5% and foliage for 16.65%. Litter production fractions define the proportion of the living biomass in a specific compartment that is added to the litter pool each year. In the example below, 0.43% of the stem biomass in spruce forests up to 30 years old is added to the soil as litter, 2.7% of the coarse root biomass, etcetera. Please note here that these amounts are not taken away from that compartment, since this is not a flow model. The stem litter fall rate for example does not influence the simulated standing volume. The user should be aware of the potential overlap with mortality as defined in the parameters file. The mortality as defined there really decreases the volume in the simulation. When mortality is defined in the parameter file, already most of the stem litterfall rate will be covered. Additional stem turnover as defined in the...
biocomp-***.txt would therefore only cover parts of the stem that die, for example bark. However, a stem litterfall rate of zero seems appropriate in most cases. An example for Czech Republic is given in Box 3.3.

```
#Allocation of Biomass by compartments and litter production (Czech Republic)
#Almost same as in parameters file
#first Carbon content
Carbon 1
#All All All All
0 0 0 0
1 0.5
#Then wood density Mg/m3
#after IPCC Good Practice Guidance for LULUCF
WoodDens 10
#All All All Spruce
0 0 0 1
1 0.4
#All All All Fir
0 0 0 2
1 0.4
#All All All Pine
0 0 0 3
1 0.42
(...)
#Then age classes
BioAgeLims 10
#All All All Spruce
0 0 0 1
11 20 30 40 50 60 70 80 90 100 110 1000
#All All All Fir
0 0 0 2
11 20 30 40 50 60 70 80 90 100 110 1000
#All All All Pine
0 0 0 3
8 30 40 50 60 80 100 120 1000
(...)
#Then allocations itself, number after name shows how many combinations are there
#
BioAllocations 10
#All All All spruce
0 0 0 1
#stem share
11 0.3852 0.4743 0.5622 0.6165 0.6424 0.6497 0.6443 0.6388 0.6339 0.6280 0.6188
#branches share
11 0.3487 0.2561 0.1725 0.1295 0.1043 0.1056 0.1082 0.1112 0.1153 0.1211
#coarse roots share
11 0.0496 0.0905 0.1366 0.1545 0.1630 0.1701 0.1787 0.1853 0.1901 0.1943 0.1994
#fine roots share
11 0.0500 0.0413 0.0297 0.0230 0.0194 0.0175 0.0165 0.0156 0.0150 0.0144 0.0140
#foliage share
11 0.1665 0.1378 0.0990 0.0765 0.0647 0.0584 0.0549 0.0521 0.0498 0.0480 0.0467
#All All All Fir
```
Extract from the file biocomp.txt for Czech Republic. Part of the data has been left out for clarity, indicated by (...).

In the matrix file, the initial area distribution over age and volume classes is defined per forest type. The area is given in units of 1000 ha. The columns represent age classes, the rows volume classes. The first row is reserved for the bare forest land class. The second row shows the area in the first volume class per age class. This file is usually generated by the P-2009 program (see Schelhaas et al. 2007 for details). An example for Czech Republic is given in Box 3.4.
### Model manuals

**Trees4Future (FP7 284181)**

Page 18 of 83

---

#### Figure 3.4: First part of the file containing the initial matrices for Czech Republic (cze.aer).

```
2.064 2.051 0.748 0.299 0.084 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.002 0.001
0.000 2.181 0.943 0.837 1.228 0.844 0.097 0.000 0.000 0.000 0.000 0.000
0.006 0.000 0.004 0.002
0.000 0.000 0.625 0.855 1.606 1.644 1.588 1.712 1.054 0.586 0.293 0.126
0.035 0.009 0.005 0.003
0.000 0.000 0.461 0.536 1.294 1.495 1.527 1.695 1.312 0.693 0.305 0.126
0.029 0.010 0.004 0.002
0.000 0.000 0.000 0.238 0.724 0.943 0.837 1.228 0.844 0.428 0.097 0.000
0.023 0.008 0.002 0.001
0.000 0.000 0.000 0.008 0.295 0.459 0.589 0.802 0.657 0.349 0.156 0.068
0.014 0.005 0.001 0.001
0.000 0.000 0.000 0.055 0.128 0.180 0.244 0.366 0.323 0.175 0.079 0.035
0.007 0.003 0.001 0.000
0.000 0.000 0.000 0.030 0.089 0.108 0.119 0.158 0.137 0.075 0.034 0.016
0.003 0.001 0.000 0.000
0.000 0.000 0.000 0.008 0.044 0.071 0.086 0.106 0.083 0.044 0.020 0.009
0.002 0.001 0.000 0.000
0.000 0.000 0.000 0.013 0.013 0.034 0.061 0.103 0.097 0.056 0.026 0.012
0.003 0.001 0.000 0.000

# cze : ST 1 REG 1, KAT 1, BON 1, TRSL 2
1 1 1 2
0.089 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000
0.022 0.053 0.011 0.008 0.009 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.002 0.001
0.000 0.044 0.016 0.018 0.035 0.035 0.020 0.020 0.015 0.019 0.017 0.009
0.009 0.005 0.002 0.001
0.000 0.000 0.013 0.019 0.040 0.042 0.046 0.048 0.042 0.036 0.025 0.021
0.009 0.006 0.002 0.001
0.000 0.000 0.007 0.013 0.032 0.039 0.040 0.041 0.035 0.030 0.021 0.016
0.008 0.005 0.002 0.001
0.000 0.000 0.004 0.007 0.018 0.025 0.029 0.030 0.026 0.022 0.015 0.012
0.006 0.004 0.001 0.000
0.000 0.000 0.000 0.003 0.008 0.013 0.016 0.017 0.016 0.013 0.009 0.008
0.004 0.002 0.001 0.000
0.000 0.000 0.000 0.001 0.004 0.005 0.007 0.008 0.008 0.007 0.005 0.004
0.002 0.001 0.000 0.000
0.000 0.000 0.000 0.001 0.002 0.003 0.004 0.004 0.004 0.003 0.002 0.002
0.001 0.001 0.000 0.000
0.000 0.000 0.000 0.000 0.001 0.002 0.003 0.003 0.002 0.002 0.001 0.001
0.001 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.001 0.002 0.002 0.003 0.003 0.002 0.002 0.002
0.001 0.001 0.000 0.000

# cze : ST 1 REG 1, KAT 1, BON 1, TRSL 3
1 1 1 3
(...)
```

#### ***.vcl

The volume class file (***.vcl) sets the limits of the volume classes for each forest type. In the example given in Box 3.5, the maximum volume of the first volume class in the matrix 1 1 1 1 is 130 m³ha⁻¹, the volume in the second volume class in the same matrix ranges from 131 to 260 m³ha⁻¹. This file is usually generated by P-2009 (see Schelhaas et al. 2007 for details). An example for Czech Republic is given in Box 3.5.

---
Box 3.5: First part of the file containing the limits of the volume classes for Czech Republic (cze.vcl).

***soil.par

The soil file (ending ***-soil.par) file contains all parameters needed by the soil carbon sub-module YASSO. EFISCEN requires soils to be defined for every region and species; soils for owner-classes and site-classes may be aggregated in one pool for all owner-classes and one pool for all site-classes. In the example of Czech Republic, there were 14 regions and 10 tree species; this means EFISCEN requires the specification of soil parameters for 140 combinations of regions and tree species.

There are two ways of initialising soil carbon stocks in EFISCEN. One way is to define the stocks for all litter compartments (as total carbon in the forest type, Gg C) (see soil pool 1 0 0 2 in Box 3.6); the other way is to run a spin-up. In the spin-up, the litter input of the first time step will be used as input to YASSO, and then YASSO is run repeatedly until the stocks are in balance. The spin-up will run automatically if the initial stocks are set to 0 (see soil pool 1 0 0 1 in Box 3.6). Values need to be separated using spaces, not tabs or commas. An example for Czech Republic is given in Box 3.6.
Box 3.6: First part of the soil parameter file for Czech Republic.

3.4.3 Scenario input files

**.scn

In the scenario definition file the file names of the forest growth change file (in case of environmental change scenarios), soil climate, removal demand, removal ratio, afforestation, deforestation, tree species change, matrix scaling, thinning change and final felling change scenarios are given. In case a user does not include tree species change, matrix scaling, thinning change and final felling change as scenario options, it is sufficient to provide only the statement ‘nofile’ instead of the file name. Box 3.7 gives an outline of the scenario definition file.
Box 3.7: Outline of the scenario definition file.

Forest growth scenario file (***_defgrow.csv)
In the forest growth scenario, the impact of environmental changes on tree growth can be defined. For each region, tree species, owner class, site class and time step, a ratio can be defined by age classes which will then be used to scale the increment. For example, 1.1 means an increment increase of 10%. If no changes are to be implemented, all ratios should be set to 1. Box 3.8 provides an example of the outline of the forest growth scenario file. The name of the applied scenario is “Fast Climate Change”. The number 1 in the next line defines the number of groups (i.e. combinations of regions, site classes etc.) for which separate age limits are given. The next block defines the age class limits for the growth change impacts. Then, 3 blocks of parameters are provided. The first block applies up to time step 2 (first line of block). The growth change is defined for each of the two species separately, but there is no growth change in this period. For time steps 3-6, the two species react differently. For example, in age class 41-60, species 1 has a 20% increment increase and species 2 only a 10% increase. The last block applies to time steps 7-10 and shows even more pronounced increment changes, up to 50% of the baseline increment in old forests of species 2. Note that if this simulation will be continued, time step 11 will show no increment change, since the first block will be repeated. Input is provided as comma separated text files (.csv).

```text
#Forest grow scenario file
#here we provide Name of scenario and then number of parameters i.e. for how many groups scenarios are given
Fast Climate Change
1
#Comments line 0 0 0 0 means for all: Age limits
0,0,0,0
7,20,40,60,80,100,160,300
#num Step,Gr_0000
2
2
0,0,0,1
7,1,1,1,1,1,1,1
0,0,0,2
7,1,1,1,1,1,1,1
6
2
0,0,0,1
7,1,1,1,1,1,2,1,2,1,2,1,2
```
Box 3.8: Example of the growth change file.

Soil climate scenario (**_defsoil.csv)**

In the soil climate scenario file, the climate dependency parameters are defined (see Section 3.9) and the assumed climate during the simulation. The user should note that the parameters in this file are not used when the model is executed without specifying a scenario. In that case, default values are used ($\alpha_1 = 0.0937$, $\alpha_2 = 0.00229$, $T = T_{\text{ref}} = 4$, $D = D_{\text{ref}} = -50$). The climate dependency parameters $\alpha_1$ and $\alpha_2$ depend on the method how to express the climate dependency: dependent on average annual temperature or on cumulative degree days (see Table 3.3 in Schelhaas et al. 2007). Also the reference conditions need to be listed. The required climate data are either average annual temperature or cumulative degree days (DD, with a threshold of 0°C) and the summer drought index (DI). They can be defined for each region and time step. The summer drought index is defined as precipitation (in the growing season) minus potential evaporation (in the growing season). If the drought index is positive (e.g. precipitation exceeds potential evaporation), it is set to zero assuming that drought does not limit decomposition processes. Box 3.9 shows part of the soil climate scenario file for the Czech Republic, with climate defined per region, assuming a constant climate over the simulation. Input is provided as comma separated text files (.csv).

Box 3.9: Part of the soil climate scenario file for Czech Republic.

Removal demand scenario (**_defcut.csv)**

In the harvest scenario, the amount of roundwood to be removed from the forest is specified. Removal amounts can be defined for the total country, or by region, owner and site class as well as tree species for each time step, separately for thinnings and final fellings. Units are 1000 m$^3$ overbark per 5 years. In the example in Box 3.10, 9 million m$^3$ of roundwood removals are requested from final fellings in each of the first two time steps, and 3 million m$^3$ removals from thinnings. Note that the actual felled volume in the forest will be higher, depending on the ratio removals/fellings that is specified in the removal ratio file. Input is provided as comma separated text files (.csv).
#number of combinations
1
#Comments line 0 0 0 0 means for all
0,0,0,0
#num_Step,Felling,Thinning
2,9000,3000
4,9100,3050

Box 3.10: Example of a harvest demand scenario file.

Removal ratio scenario file (**_defrem.csv)
The removal ratio scenario defines the proportion of the stems, topwood, deadwood, branches and foliage that is removed from the forest, separately for thinnings and final fellings. It is possible to define those proportions for each forest type. In the example in Box 3.11, 90% of the stem is removed in the case of final fellings, and 94% in the case of thinnings. The remaining 10% for final fellings and 6% for thinnings is considered topwood, of which an additional 50% is extracted as thinning or felling residue. The remaining 50% is assumed to be left on site to decompose, as estimated by the soil module YASSO. Furthermore, 36% of branches and foliage is removed and the remaining branches and foliage are left on site to decompose, as estimated by the soil module YASSO. Standing deadwood is not removed and will fall down following the deadwood fall rate specified in Box 3.2. The user should note that the parameters in this file are not used when the model is executed without specifying a scenario. In that case default values are used: removal rate for final fellings 0.95, removal rate for thinnings 0.9 and no removal of topwood, deadwood branches or foliage. Input is provided as comma separated text files (.csv).

#Removals scenario file
#here we Name of scenario, then provide number of parameters, i.e. for how many groups scenarios are given based on TBFRA
1
#Comments line 0 0 0 0 means for all
0,0,0,0
#num_Step,Fel_stem,fel_tops,Fel_branches,Fel_leaves,Fel_dwood,Thin_stem,Thin_tops,Thin_branches,Thin_leaves,Thin_dwood
100,0.9,0.5,0.36,0.36,0,0.94,0.5,0.36,0.36,0

Box 3.11: Example of the definition of removal ratios file

Afforestation scenario (**_affor.csv)
Areas for afforestation can be defined by region, owner class, site class and tree species. Units are 1000 ha in a 5-year time step, as a total for the combination specified. The area for afforestation will be added to the bare forest land class of the respective forest type matrix or matrices. In the example of Box 3.12, total afforestation will be 18,200 ha, of which 12,900 ha distributed over the forest types in the first region, and 5,300 ha distributed over all forest types in the second region. The area is distributed according to the area already present per forest type. Note that negative values are ignored by the software. Further, all forest types should be covered, even when no afforestation occurs in those types. If the simulated period exceeds the last time step defined in the afforestation file, the afforestation scenario will be repeated. In the case below, time step 6 and 7 will have no afforestation, time step 8 and 9 12,900 ha, etcetera. Note that there is no possibility to initialise the soil carbon pools of afforested areas. Input is provided as comma separated text files (.csv).

#Forest afforestation scenario file
#Comments (do not remove first two lines)
Afforestation scenario
2
#Comments
1,0,0,0,2,0,0,0
#Afforestation
2,0,0
4,12.9,5.3
5,4,1,0

Box 3.12: Example of the afforestation scenario file

Deforestation scenario (**_defor.csv)
Areas for deforestation can be defined by region, owner class, site class and tree species. Units are 1000 ha in a 5-year time step. The area for deforestation will be removed from the bare forest land class(es) of the concerned matrices. Thus, deforestation can only take place after a regular final harvest has occurred. If there is not enough area in the bare land class, actual deforestation will simply be equal to the area in the bare land class. In the example of Box 3.13, no deforestation takes place during the first five time steps, and in total 29,800 ha is removed from the bare forest land class of all forest types in time step 6. This total per five years is divided over the tree species according to their ratio in the total forest area. Note that negative values are ignored by the software. Further, all forest types should be covered, even when no deforestation occurs in those types. If the simulated period exceeds the last time step defined in the deforestation file, the deforestation scenario will be repeated. In the case below, time step 7 to 11 will have no deforestation, while time step 12 will have 29,800 ha deforestation. Soil carbon pools are not directly affected by deforestation, so in principle soil carbon of deforested areas is still included. Input is provided as comma separated text files (.csv).

Czech deforestation scenario
1
#Comments
0,0,0,0
#Deforestation
5,0
6,29.8

Box 3.13: Example of the deforestation scenario file

Species change file (**_sp_change.csv)
If nothing is specified ('nofile' in the ***.scn file) then it is always assumed that the same species regenerates as there was before the final harvesting. In this file it is possible to specify by region, owner class, site class and tree species if a tree species change occurs at the time of final felling, how much is regenerated as another species, and to which species. The example in Box 3.14 shows that that for the next 100 time steps, species nr 3 will be regenerated to one other species. That species number is nr 8, and this will happen to 60% of all clearcuts of species nr 3. Input is provided as comma separated text files (.csv).

#First 4 lines for explanations
#Species change scenario sample file
#first we set for how many steps scenario
#total number of steps in the scenario:
1
#number of matrices(species) - who lost the area (source)
1
Then step of simulation until the following changes are valid (as in any scenario file)
100
#
region, owner, site, species of "source" and how many different tree species are the "destination" species. On the next line: the number of the destination species, and what fraction of regenerated area will change to this new destination species.
1,1,1,3,1
8,0.6

Box 3.14: Example of the species change file

Matrix scaling file (**_matrix_scaling.csv)
The area according to the EFISCEN input data can be scaled to match another area, for example the area reported in international statistics. In EFISCEN 3.1.3 one scaling factor could be applied to scale the area in all matrices. In EFISCEN 4.1 a feature has been added to apply different scaling factor to individual matrices. The scaling of individual matrices is combined with the scaling of all matrices.

If nothing is specified (‘nofile’ in the ***.scn file) then it is always assumed that matrices are not scaled individually (i.e. a scaling factor of 1 is assumed). If input is specified, the file is formatted so that the first two lines are always reserved for comments and are not processed by the input loader. The third line has the total number of entries in the file, and after that are the entries consisting of matrix id and a scaling value. The matrix id consists of four numbers identifying region, owner, site and species. The numbers are comma separated. Any of these numbers can be zero meaning all for that category. Setting the matrix id to 0,0,0,0 means scaling is applied to all matrices. Note that scaling is applied only once, before first step of simulation applied, which is different from other scenario files. Input is provided as comma separated text files (.csv). An example is shown in Box 3.14.

Box 3.14: Example of the matrix scaling file

Thinning change file (**.tcf)
In EFISCEN 3.1.3 management parameters for thinning were set in the parameters file and were fixed for the whole simulation period. In EFISCEN 4.1 a feature has been added to define management parameters that can change during the simulation.

If nothing is specified (‘nofile’ in the ***.scn file) then the management parameters from the parameters file are used. If an input file is provided, EFISCEN will use information form that file. In the input file, the minimum and maximum age for thinnings is defined. In the example on Box 3.15, thinnings are carried out on forests between 20 and 75 years old until step 2, in steps 3 to 4 thinnings are carried out on forests between 20 and 50 years old and in steps 4-100 thinnings are carried out on forests between 20 and 70 years old.

Box 3.15: Example of the thinning change file

#Thinning parameter file
#Here we provide number of parameters for how many groups scenarios are given
**Box 3.14: Example of thinning change file**

**Final felling change file (***.tcf)**

In EFISCEN 3.1.3 management parameters for final fellings were set in the parameters file and were fixed for the whole simulation period. In EFISCEN 4.1 a feature has been added to define management parameters that can change during the simulation.

If nothing is specified (‘nofile’ in the ***.scn file) then the management parameters from the parameters file are used. If an input file is provided, EFISCEN will use information from that file. In the input file, the minimum age for final fellings is defined. In the example on Box 3.16, final fellings are carried out on forests older than 75 years old until step 2, in steps 3 to 4 final fellings are carried out in forests of at least 80 years old and in steps 4-100 final fellings are carried out in forests older than 75 years.

**Box 3.15: Example of final felling change file**

### 3.5 Model simulations

EFISCEN can be operated from a graphical user interface and using a command line.

#### 3.5.1 Graphical user interface

To open the graphical user interface (GUI) of the program in this mode the user can go to the EFISCEN folder and use an EFISCEN4 short cut (under Microsoft Windows). Alternatively the GUI can be opened from the command line by typing in the command prompt:

```bash
efiscen_gui.jar
```

EFISCEN performs by design all outputs into comma separated value files (.csv) with comma (,) as list delimiter and dot (.) as decimal symbol. In some locales the comma is a decimal symbol. To avoid the problems with outputs, two additional parameters must be added in the command line call. Those parameters will prevent EFISCEN4 from using the wrong local settings. It does not affect other applications or operating system itself. The parameters are:

- Duser.language=US
• Duser.region=US

Then our commands will look like this:

```java
java -Duser.language=US -Duser.region=US -jar Efiscen_guifx.jar
```

When the GUI is opened and no data is loaded, many of the buttons are disabled. At the top of the window at the menu bar File, Settings and About are available. When no data is loaded Open logs is disabled. By choosing settings, filepaths and usernames for database can be saved. Settings are saved in a text file “settings.txt”, located in the same folder as Efiscen.jar. When output-window is opened, settings are loaded to fields by default. Option to save settings is only available in GUI. An example of a settings file is shown in Box 3.16.

```
inputpath=C:\foldername
```

**Box 3.16: example of settings file**

Data can be loaded under the File-menu by choosing Load files (Figure 3.2). EFISCEN accepts filenames ending with .efs or .scn. It is possible to select both the experiment and scenario file to be loaded by pressing the Ctrl-button when selecting the files. Experiments can be loaded without a scenario file.

![Image](image-url)

**Figure 3.2: The interface at the program start. Before data is loaded many options are locked. Data can be loaded from File-menu.**
After loading the experiment file some data is already shown. If EFISCEN runs into problems, it reports about them in the console (Figure 3.3, number 2). More info about these errors can be found in the log files, which can be found by pressing Open logs (Figure 3.3, number 3).

If the experiment file is loaded without any scenario, thinning intensity and felling intensity can be entered (Figure 3.3, number 4). This must be done before running the simulation or the changes will not take effect. When a scenario is loaded, thinning and felling are disabled. Scaling can be entered at any time and after clicking Apply scaling will be calculated immediately.

Simulations can be run by entering number of steps (5 by default) and pressing run (Figure 3.3, number 5). You can browse graphs indicating various variables during the simulation by clicking data buttons under the graph (Figure 3.3, number 6). You can get more info about the buttons by placing the cursor on top of the button for a moment.

Figure 3.3: shows the interface after data is loaded and simulation has been run. Dead wood and national mortality have been chosen and are shown in the graph.

Under the Selected tab you can browse graphs for different datasets (Figure 3.4, number 6). When you click the tab, a selection tree opens on the left (Figure 3.4, number 7). Region, owner, site and species can be expanded to show all the items under that category by clicking on the triangle next to them. Multiple entries can be selected on the lists by pressing Ctrl-button when clicking on the items. If nothing is selected under some
category, for example owners-category, or the user has selected a category-box then data is shown for all owners. Selection only affects what is shown in the selected data-tab, and the total data always shows values for the whole dataset.

**Figure 3.4:** Selected tab has been selected. Karnten and Tirol have been selected from Regions and Oak has been selected from Species. The results shown in the chart area and the data buttons show the aggregated results for Oak in Karnten and Tirol for all owners and age-classes.

Simulation results can be saved under File-menu by choosing Output (Figure 3.2). This opens the output window (Figure 3.5). Under the tab Files you can save results to files. From the selection tree on the right side of the window you can to select which outputs to save. Under the tab Database, it is possible to save output data to a database. Currently database types Microsoft Access, MySQL and PostgreSQL are supported (see section 3.5.3 for details).
3.5.2 Command line

EFISCEN 4.1 can be used from the command line by opening the command prompt and then specifying the command:

```
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI variable1=argument1 variable2=argument2...
```

Where variable1, variable2 etc. are the argument variables and argument1, argument2 etc. are the values for these arguments.

EFISCEN 4.1 can provide outputs as comma separated value (csv) files or write results or outputs can directly be saved to an external database. In both approaches, it is possible to select which outputs should be saved. When running from the command line the user can select which outputs to save by using a simple text file. The name of the text file is given as a parameter to the command line interface. Outputs are defined with a name and either 1 or 0. 1 means the output will be saved while 0 means it is not. If an output name is not present in the file, then it will not be saved. If outputs are saved both into database and files the selections in the text file apply to both, meaning that only the selected outputs are saved. The output file must have “.txt” - extension to work properly.
Table 3.1: Output options

<table>
<thead>
<tr>
<th>Output name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>General data</td>
</tr>
<tr>
<td>carbon_country</td>
<td>Carbon country data</td>
</tr>
<tr>
<td>carbon_soil</td>
<td>Carbon soil data</td>
</tr>
<tr>
<td>gdat</td>
<td>General by regions. This data is not saved into database even if it is</td>
</tr>
<tr>
<td>gspec</td>
<td>General by species. This data is not saved into database even if it is</td>
</tr>
<tr>
<td>deadwood</td>
<td>Deadwood data. This data is not saved into a file even if it is specified</td>
</tr>
<tr>
<td>felling_matrix</td>
<td>Felling matrix data even if it is specified</td>
</tr>
<tr>
<td>felling_residues</td>
<td>Felling residue data</td>
</tr>
<tr>
<td>natmort</td>
<td>Natural mortality</td>
</tr>
<tr>
<td>thinning_matrix</td>
<td>Thinning matrix data</td>
</tr>
<tr>
<td>thinning_residues</td>
<td>Thinning residue data</td>
</tr>
<tr>
<td>treec_matrix</td>
<td>Tree carbon data</td>
</tr>
</tbody>
</table>

An example of an output selection file is given in Box 3.16.

```
base 1
carbon_country 0
carbon_soil 1
gdat 1
gspec 1
deadwood 1
felling_matrix 1
felling_residues 1
natmort 1
thinning_matrix 0
thinning_residues 1
treec_matrix
```

Box 3.16: Example of an output selection file

Arguments for running EFISCEN and saving outputs as files
When running EFISCEN from command line and saving simulation results to a file, there are
6 required arguments and 2 optional arguments (Table 3.2).
Table 3.2: command line arguments to run EFISCEN and saving results as files

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Type</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. steps</td>
<td>Number of steps to run the simulation.</td>
<td>Integer</td>
<td>Required</td>
</tr>
<tr>
<td>2. thinning</td>
<td>Intensity of the thinning applied during the simulation.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>3. felling</td>
<td>Intensity of the felling applied during the simulation.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>4. experiment</td>
<td>File path to an experiment file. Must include the file name.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>5. scaling</td>
<td>Scaling factor that is applied globally to areas of all the matrices.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>6. scenario</td>
<td>File path to a scenario file. The scenario file name must end with &quot;.scn&quot;. Path must include the file name.</td>
<td>Text</td>
<td>Optional</td>
</tr>
<tr>
<td>7. outputfile</td>
<td>File path to where the output files will be saved. Path must include a file name (for example C:\Folder\FileName would save the outputs in C:\Folder\ and the files would have names starting with FileName ).</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>8. selected</td>
<td>File path to a text file containing definitions about which outputs to save. The name must end with &quot;.txt&quot;.</td>
<td>Text</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Examples:
The following example shows how to run EFISCEN tool from the command line with only the required parameters.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.5 experiment=C:\Path\Utopia.efs scaling=1.0 outputfile=C:\OutputPath\Utopia
```

The next example has all the required and optional parameters used.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.0 experiment=C:\Path\Utopia.efs scaling=1.0 scenario=C:\Path\Utopia.scn outputfile=C:\OutputPath\Utopia selected=C:\Path\selections.txt
```

In case a file path or database address has spaces in them, quotation marks can be used to enter them (e.g. "C:\Path\name with spaces.efs").

Arguments for running EFISCEN and saving to database
The command line arguments to run EFISCEN and save results in a database are shown in Table 3.3.
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Type</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. steps</td>
<td>Number of steps to run the simulation.</td>
<td>Integer</td>
<td>Required</td>
</tr>
<tr>
<td>2. thinning</td>
<td>Intensity of the thinning applied during the simulation.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>3. felling</td>
<td>Intensity of the felling applied during the simulation.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>4. experiment</td>
<td>File path to an experiment file. File name must be included.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>5. scaling</td>
<td>Scaling factor that is applied globally to areas of all the matrices.</td>
<td>Decimal number</td>
<td>Required</td>
</tr>
<tr>
<td>6. scenario</td>
<td>File path to a scenario file. The file name must be included and the name must end with “.scn”.</td>
<td>Text</td>
<td>Optional</td>
</tr>
<tr>
<td>7. outputfile</td>
<td>File path to where the output files will be saved. Path must include a file name (for example C:\Folder\FileName would save the outputs in C:\Folder\ and the files would have names starting with FileName).</td>
<td>Text</td>
<td>Optional</td>
</tr>
<tr>
<td>8. databaseaddress</td>
<td>Address to a database where outputs will be saved. In case ODBC drivers are used to connect to database, configured ODBC data source name (see Appendix 2) can be entered as the database address. PostgreSQL and MySQL database address is entered in following form: //database_address/database_name.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>9. username</td>
<td>User name used to log into the database specified in the address.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>10. password</td>
<td>Password used to log into the database specified in the address.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>11. sid</td>
<td>Session id that will be present in all data entries saved into a database. Can be used for example to identify data from certain run of the EFISCEN tool.</td>
<td>Integer</td>
<td>Required</td>
</tr>
<tr>
<td>12. ciso</td>
<td>ISO country-code is used to identify that the output data concerns a certain country.</td>
<td>Integer</td>
<td>Required</td>
</tr>
<tr>
<td>13. selected</td>
<td>File path to a text file containing definitions about which outputs to save. The Path must include the file name and the name must end with “.txt”. The same selections are used for saving outputs to files and to a database. If this file is not used then all outputs are saved.</td>
<td>Text</td>
<td>Optional</td>
</tr>
<tr>
<td>14. pid</td>
<td>Project id that can be used to group simulations together. If this is not entered, project id will be set as 0.</td>
<td>Integer</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Notice that you can save to either files or to a database or at both of these the same time. Just specify both outputfile and databaseaddress.

In case a file path or database address has spaces in them, quotation marks can be used to enter them. For example: experiment=”C:\Path\name with spaces.efs”.
Examples:
The following example shows how to run EFISCEN tool from the command line with only the parameters required for database use.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.0 experiment=C:\Path\Utopia.efs scaling=1.0 databaseaddress=mysql:mysql.database.com/database username=user password=password sid=1 ciso=40
```

The next example runs EFISCEN and saves only the selected outputs to both database and files. All required and optional parameters are used.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.0 experiment=C:\Path\Utopia.efs scaling=1.0 scenario=C:\Path\Utopia.scn outputfile=C:\OutputPath\Utopia databaseaddress=mysql:mysql.database.com/database username=user password=password sid=1 ciso=40 selected=C:\Path\selections.txt pid=100
```

The following example demonstrates saving all outputs only to a database and a path to save output files is therefore not specified. Project id is not specified, so it will be set to 0 for this simulation.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.0 experiment=C:\Path\Utopia.efs scaling=1.0 scenario=C:\Path\Utopia.scn databaseaddress=mysql:mysql.database.com/database username=user password=password sid=1 ciso=40
```

Database type is selected by prefixing the address with the type name. Valid type names are MYSQL, odbc and PostgreSQL. This example saves data into Access (odbc) database.

```java
java -cp Efiscen_guifx.jar int_.efi.efiscen.cli.EfiscenCLI steps=2 thinning=1.0 felling=1.0 experiment=C:\Path\Utopia.efs scaling=1.0 scenario=C:\Path\Utopia.scn databaseaddress:odbc:someOdbcDatabase username=user password=password sid=1 ciso=40
```

### 3.6 Model outputs

#### 3.6.1 Conventions

The output of EFISCEN consists of a series of files, all starting with a user defined string (for example test1), here represented as x. In these files the development of growing stock, increment, age class distribution, amount of wood harvested by final felling and by thinning, area affected by final cuttings and thinning, and biomass data of stem, roots, needles/leaves, branches, litter production, slash and soil are presented. Some variables are given for the total area and some also per tree species and/or region. The output structure is the same for all countries, but the number of lines and/or columns might vary due to varying numbers of regions, owner classes, site classes and tree species.
All files are comma separated text files (.csv). They can be analysed in spreadsheet software like MS Excel, and they can be imported to database software for the management of a large number of output files and more advanced queries.

All variables concerning carbon start with a C, except for soil variables which always concern carbon. The term trees always refers to the total biomass of trees, including foliage, branches and roots.

3.6.2 Output as text files

Detailed volume output (x.csv)
The x.dat file contains detailed information on growing stock, increment and forest area per region, owner, site class, and tree species. Forest area and volume per age classes is also given in this file. Bare forest land is not included in the lowest age class (0-10), but it is included in the total area.

Table 3.4: description of variables in detailed volume output (x.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_ID</td>
<td>Matrix ID number (for internal purposes)</td>
<td>Number</td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>GrStock</td>
<td>Volume of growing stock</td>
<td>1000 m³</td>
</tr>
<tr>
<td>Area</td>
<td>Forest area (including regeneration area (= bare forest land))</td>
<td>1000 ha</td>
</tr>
<tr>
<td>DeadWood</td>
<td>Volume of standing dead wood</td>
<td>1000 m³</td>
</tr>
<tr>
<td>NatMort</td>
<td>Volume of natural mortality</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>ThinHarvest</td>
<td>Volume of removals from thinnings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelHarvest</td>
<td>Volume of removals from final fellings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelAv</td>
<td>Volume of total removals (thinnings + final fellings) per ha</td>
<td>m³ per ha</td>
</tr>
<tr>
<td>GrStockAv</td>
<td>Growing stock per ha</td>
<td>m³ per ha</td>
</tr>
<tr>
<td>IncrAv</td>
<td>Net annual increment per ha</td>
<td>m³ per ha</td>
</tr>
<tr>
<td>A_0 - 10</td>
<td>Forest area per 10 year age class</td>
<td>1000 ha</td>
</tr>
<tr>
<td>A_10 - 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_0 - 10</td>
<td>Growing stock per 10 year age class</td>
<td>1000 m³</td>
</tr>
<tr>
<td>V_10 - 20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Volume output by regions (x_gdat.csv)
Each row represents one time-step, where the first row shows the initial situation, the second row the state at the end of first time step etc. The first block of columns contains the results for the first region, characterised by the region name or its ID. The last set of columns refers to the totals at country level.
Table 3.5: description of variables in Volume output by regions (x_gdat.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>Area</td>
<td>Forest area (including regeneration area (=bare forest land))</td>
<td>1000 ha</td>
</tr>
<tr>
<td>GrStock</td>
<td>Volume of growing stock</td>
<td>1000 m³</td>
</tr>
<tr>
<td>ThinHarvest</td>
<td>Volume of removals from thinnings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelHarvest</td>
<td>Volume of removals from final fellings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelAv</td>
<td>Volume of total removals (thinnings + final fellings) per ha</td>
<td>m³ per ha per year</td>
</tr>
<tr>
<td>GrStockAv</td>
<td>Growing stock per ha</td>
<td>m³ per ha</td>
</tr>
<tr>
<td>IncrAv</td>
<td>Net annual increment per ha</td>
<td>m³ per ha per year</td>
</tr>
<tr>
<td>C_GrStock</td>
<td>Carbon in growing stocks</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_DWood</td>
<td>Carbon in standing dead wood</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_ThRem</td>
<td>Carbon removed in thinnings (including foliage and branches)</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_FelRem</td>
<td>Carbon removed in final fellings (including foliage and branches)</td>
<td>Gg C</td>
</tr>
</tbody>
</table>

Volume output by tree species (x_gspec.csv)

Each row represents one time-step, where the first row shows the initial situation, the second row the state at the end of first time step etc. The first block of columns contains the results for the first tree species, characterised by its number as defined in the efs file. The last set of columns refers to the totals at country level.

Table 3.6: description of variables in Volume output by tree species (x_gspec.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>Area</td>
<td>Forest area (including regeneration area (=bare forest land))</td>
<td>1000 ha</td>
</tr>
<tr>
<td>GrStock</td>
<td>Volume of growing stock</td>
<td>1000 m³</td>
</tr>
<tr>
<td>ThinHarvest</td>
<td>Volume of removals from thinnings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelHarvest</td>
<td>Volume of removals from final fellings</td>
<td>1000 m³ per 5 years</td>
</tr>
<tr>
<td>FelAv</td>
<td>Volume of total removals (thinnings + final fellings) per ha</td>
<td>m³ per ha per year</td>
</tr>
<tr>
<td>GrStockAv</td>
<td>Growing stock per ha</td>
<td>m³ per ha</td>
</tr>
<tr>
<td>IncrAv</td>
<td>Net annual increment per ha</td>
<td>m³ per ha per year</td>
</tr>
<tr>
<td>C_GrStock</td>
<td>Carbon in growing stocks</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_DWood</td>
<td>Carbon in standing dead wood</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_ThRem</td>
<td>Carbon removed in thinnings (including foliage and branches)</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_FelRem</td>
<td>Carbon removed in final fellings (including foliage and branches)</td>
<td>Gg C</td>
</tr>
</tbody>
</table>

Removals by age classes

The files x_fell_matr.csv and x_thin_matr.csv give output on the removals from final fellings and thinnings, respectively.
### Table 3.7: description of variables in removals by age classes (x_fell_matr.csv and x_thin_matr.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_ID</td>
<td>Matrix ID number (for internal purposes)</td>
<td>Number</td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>FelRem/ThinRem</td>
<td>Volume of removals from final fellings c.q. thinnings</td>
<td>1000 m$^3$ per 5 years</td>
</tr>
<tr>
<td>A_0 - 10</td>
<td>Forest area on which final fellings c.q. thinnings have been executed, by 10 year age class</td>
<td>1000 ha in 5 years</td>
</tr>
<tr>
<td>V_0-10, V_10-20</td>
<td>Volume of removals from final fellings c.q. thinnings by 10 year age class</td>
<td>1000 m$^3$ per 5 years</td>
</tr>
</tbody>
</table>

### Natural mortality (x_natmort.csv)

The file x_natmort.csv contains information on the amount of mortality that occurred and on the standing deadwood pool.

### Table 3.8: description of variables in Natural mortality (x_natmort.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_ID</td>
<td>Matrix ID number (for internal purposes)</td>
<td>Number</td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>Nmort</td>
<td>Total volume of mortality</td>
<td>1000 m$^3$ per 5 years</td>
</tr>
<tr>
<td>DWood</td>
<td>Total volume of standing dead wood</td>
<td>1000 m$^3$</td>
</tr>
<tr>
<td>C_DWood</td>
<td>Carbon in standing dead wood</td>
<td>Gg C</td>
</tr>
<tr>
<td>DW_0 - 10</td>
<td>Volume of standing dead wood by 10 year age class</td>
<td>1000 m$^3$</td>
</tr>
<tr>
<td>NM_0-10, NM_10-20</td>
<td>Volume of mortality by 10 year age class</td>
<td>1000 m$^3$ per 5 years</td>
</tr>
</tbody>
</table>

### Detailed soil carbon output (x_carbon_soil.csv)

This file contains more detailed information on carbon in different soil compartments per region and tree species.
**Table 3.9:** description of variables in detailed soil carbon output (x_carbon_soil.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_ID</td>
<td>Soil ID number (for internal purposes)</td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Years</td>
</tr>
<tr>
<td>C_Trees</td>
<td>Carbon in trees</td>
<td>Gg C</td>
</tr>
<tr>
<td>CWL</td>
<td>Carbon in coarse woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>FWL</td>
<td>Carbon in fine woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>NWL</td>
<td>Carbon in non-woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>SOL</td>
<td>Carbon in soluble compounds</td>
<td>Gg C</td>
</tr>
<tr>
<td>CEL</td>
<td>Carbon in holocellulose</td>
<td>Gg C</td>
</tr>
<tr>
<td>LIG</td>
<td>Carbon in lignin-like compounds</td>
<td>Gg C</td>
</tr>
<tr>
<td>HUM1</td>
<td>Carbon in first humus compartment</td>
<td>Gg C</td>
</tr>
<tr>
<td>HUM2</td>
<td>Carbon in second humus compartment</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Soil</td>
<td>Total carbon stock in soil</td>
<td>Gg C</td>
</tr>
<tr>
<td>COUT</td>
<td>Carbon released to atmosphere (gross)</td>
<td>Gg C in 5 years</td>
</tr>
<tr>
<td>LITIN</td>
<td>Litter input to soil carbon pool</td>
<td>Gg C in 5 years</td>
</tr>
</tbody>
</table>

**Detailed tree carbon output (x_treeC_matr.csv)**

This file contains more detailed information on carbon in different tree compartments per region, owner class, site class and tree species.

**Table 3.10:** description of variables in detailed tree carbon output (x_treeC_matr.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_ID</td>
<td>ID number</td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>C_Trees</td>
<td>Carbon in tree biomass</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_St_0 - 10, ...</td>
<td>Carbon in stems per 10 year age class</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Br_0 - 10, ...</td>
<td>Carbon in branches per 10 year age class</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Lv_0 - 10, ...</td>
<td>Carbon in foliage per 10 year age class</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Cr_0 - 10, ...</td>
<td>Carbon in coarse roots per 10 year age class</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Fr_0 - 10, ...</td>
<td>Carbon in fine roots per 10 year age class</td>
<td>Gg C</td>
</tr>
</tbody>
</table>

**Aggregated carbon output by country (x_carbon_country.csv)**

Time steps in rows, first row initial situation, second row state at the end of first time step etc.
Table 3.11: description of variables in aggregated carbon output by country (x_carbon_country.csv) file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>C_Trees</td>
<td>Carbon in total tree biomass</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Stem</td>
<td>Carbon in tree stems</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Leaves</td>
<td>Carbon in foliage</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Branches</td>
<td>Carbon in branches</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_CRoots</td>
<td>Carbon in coarse roots</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_FRoots</td>
<td>Carbon in fine roots</td>
<td>Gg C</td>
</tr>
<tr>
<td>CWL</td>
<td>Carbon in coarse woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>FWL</td>
<td>Carbon in fine woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>NWL</td>
<td>Carbon in non-woody litter</td>
<td>Gg C</td>
</tr>
<tr>
<td>SOL</td>
<td>Carbon in soluble compounds</td>
<td>Gg C</td>
</tr>
<tr>
<td>CEL</td>
<td>Carbon in Holocellulose</td>
<td>Gg C</td>
</tr>
<tr>
<td>LIG</td>
<td>Carbon in lignin-like compounds</td>
<td>Gg C</td>
</tr>
<tr>
<td>HUM1</td>
<td>Carbon in first humus compartment</td>
<td>Gg C</td>
</tr>
<tr>
<td>HUM2</td>
<td>Carbon in second humus compartment</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_Soil</td>
<td>Total carbon in soil</td>
<td>Gg C</td>
</tr>
<tr>
<td>COUT</td>
<td>Carbon released to atmosphere (gross)</td>
<td>Gg C in 5 years</td>
</tr>
</tbody>
</table>

Residues from management operations

The output files x_fell_residues.csv and x_thin_residues.csv contain information on carbon quantities in stem, branch and foliage residues removed or added to the soil model. Age class information is for now only available for residues from branches and foliage.

Table 3.12: description of variables in Residues from management operations file

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_ID</td>
<td>Matrix ID number (for internal purposes)</td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td>Region</td>
<td>Number</td>
</tr>
<tr>
<td>OWN</td>
<td>Owner class</td>
<td>Number</td>
</tr>
<tr>
<td>ST</td>
<td>Site class</td>
<td>Number</td>
</tr>
<tr>
<td>SP</td>
<td>Species</td>
<td>Number</td>
</tr>
<tr>
<td>Step</td>
<td>Time step (end year)</td>
<td>Year</td>
</tr>
<tr>
<td>C_TopsRes</td>
<td>Carbon in stem residues added to the soil</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_BrRes</td>
<td>Carbon in residues from branches added to the soil</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_LvRes</td>
<td>Carbon in residues from foliage added to the soil</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_TopsRem</td>
<td>Carbon removed in topwood residues</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_BrRem</td>
<td>Carbon removed in branches residues</td>
<td>Gg C</td>
</tr>
<tr>
<td>C_LvRem</td>
<td>Carbon removed in foliage residues</td>
<td>Gg C</td>
</tr>
<tr>
<td>0 - 10</td>
<td>Carbon in residues from branches and foliage</td>
<td>Gg C</td>
</tr>
<tr>
<td>10 - 20</td>
<td>Carbon in residues from branches and foliage</td>
<td>Gg C</td>
</tr>
</tbody>
</table>

3.6.3 Output database

Outputs can also directly be saved to an external database. The database needs to be created by the user, i.e. it is not created by the model. Databases that are currently supported are MySQL, PostgresSQL and Microsoft Access. A technical description of the
tables and scripts to generate the required tables are available upon request from the developers.

EFISCEN 4.1 writes results in the tables listed in Table 3.13. These tables are the equivalents of the text files described in Tables 3.4 and 3.7-3.12. Values for IDs and steps are integers and all other value types are float.

**Table 3.13: overview of tables with simulation results**

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
<th>Equivalent output file</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>Basic results by matrix.</td>
<td>x.csv (Table 3.4)</td>
</tr>
<tr>
<td>natmort</td>
<td>Natural mortality results by matrix.</td>
<td>x_fell_matr.csv and x_thin_matr.csv (Table 3.7)</td>
</tr>
<tr>
<td>carbonsoil</td>
<td>Carbon in the soils by soil ID.</td>
<td></td>
</tr>
<tr>
<td>fellingmatrix</td>
<td>Felling results by matrix.</td>
<td></td>
</tr>
<tr>
<td>thinningmatrix</td>
<td>Thinning results by matrix.</td>
<td></td>
</tr>
<tr>
<td>tree</td>
<td>Carbon in the trees results by matrix.</td>
<td></td>
</tr>
<tr>
<td>fellresidues</td>
<td>Felling residues by matrix.</td>
<td></td>
</tr>
<tr>
<td>thinresidues</td>
<td>Thinning residues by matrix.</td>
<td></td>
</tr>
<tr>
<td>carboncountry</td>
<td>Carbon results by country.</td>
<td></td>
</tr>
<tr>
<td>deadwood</td>
<td>Dead wood results by country.</td>
<td></td>
</tr>
<tr>
<td>simulation</td>
<td>Simulation metadata -table. One row contains information about single simulation.</td>
<td></td>
</tr>
</tbody>
</table>

EFISCEN prints simulation results to these tables. The table simulation contains metadata of one run. A unique identification number is given to simulation and IDs of scenario, country and project related to this simulation are saved. Description can be added later to make notes of the simulation run.

In addition to the simulation results tables, several look-up tables are part of the database (Table 3.14). Values for IDs are integers and descriptions are of the type varchar. The table ‘matrix’ is the only lookup table which EFISCEN writes to. Other lookup tables store relevant information about the simulation data, but EFISCEN does not alter these tables.

**Table 3.14: overview of look-up tables**

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix</td>
<td>Region, owner, site and species for a country.</td>
</tr>
<tr>
<td>scenario</td>
<td>Scenario names and descriptions.</td>
</tr>
<tr>
<td>country</td>
<td>Information about EFISCEN countries.</td>
</tr>
<tr>
<td>species</td>
<td>Species information for countries.</td>
</tr>
<tr>
<td>owner</td>
<td>Owners information for countries.</td>
</tr>
<tr>
<td>region</td>
<td>Regions information for countries.</td>
</tr>
<tr>
<td>site</td>
<td>Sites information for countries.</td>
</tr>
<tr>
<td>countrygroup</td>
<td>List of groupings for countries.</td>
</tr>
<tr>
<td>countryregion</td>
<td>List of country regions.</td>
</tr>
<tr>
<td>project</td>
<td>List of project names and descriptions.</td>
</tr>
</tbody>
</table>

### 3.6.4 Error logs

When loading experiment and scenario files input loading process EFISCEN 4.1 writes three log files and they are found the two folders located at “C:\Users\***\EFISCEN”. In case the folders do not exist, it is created during the first run.
In the folder “logs”, two log files start with the name of the country, followed by timestamp. Both log files are simple text files. The first file has “events” in its name and logs input loading processes events, mainly starting and finishing the reading of a file. The other has “errors” in its name and contains all the errors encountered by the input loading process. If using the graphical user interface the user is informed when errors are encountered during the loading.

In the folder “debug” a log file is created that contains information mainly useful for developers and is used for debugging purposes. Debug log is named by timestamp and inside the log can be found information about the files that caused the debug message.
4. User guide disaggregation tool

4.1 Introduction
EFISCEN is not a spatially explicit model. The level of detail in the input data determines also the level of detail of the output data. Consequently, EFISCEN estimates the future state of European forests, as well as the services they provide, based on forest information at the level of administrative regions. However, EFISCEN also projects the future state at the level of individual tree species. This feature can be used to disaggregate EFISCEN results by linking tree species in EFISCEN with spatially-explicit tree species maps. The approach has been developed and applied by Elbersen et al. (2012) and Crouzat et al. (2015) and has been formalised in the EFISCEN disaggregation tool. The program has been developed and tested using Java version 7 update 25.

4.2 Requirements

4.2.1 System requirements
The software is memory intensive. The input files require memory allocated for the Java Virtual Machine (Program that runs all Java code).
Minimum requirements:
- 3 gigabytes of RAM
- 32 bit operating system
- 32 bit Java installation
- 32 bit Office installation when using Access database

Recommended specifications:
- 4 gigabytes of RAM
- 64 bit operating system
- 64 bit Java installation
- 64 bit Office installation when using Access database

4.1.2 Software requirements
Java Virtual Machine, or JVM, is the application that executes and runs the tool. It is installed by Java installation packages. In order to function correctly with the Access database, the JVM bitness must be the same as the installed Office products. For example, 32-bit MS Office products need a 32 bit JVM in order for the tool to connect to the database.

The tool uses a translator library called GDAL (Geospatial Data Abstraction Library) to process the rasters. In order to use the tool, GDAL must be installed to the machine. The installer for Windows version of GDAL can be found at http://www.gisinternals.com/sdk/PackageList.aspx?file=release-1600-gdal-1-10-mapserver-6-2.zip.
Use the Generic installer for the GDAL core components. Visual C++ runtimes may also have to be installed. If the program does not work, putting “C:\Program Files\GDAL” to the Path - environment variable may help.
5.2.2 Database drivers

Database drivers are used for accessing databases. There are currently three types of databases supported: Microsoft Access, MySQL and PostgreSQL. In order to make connections to selected database, appropriate drivers must be installed. The driver type depends on the type of the database and used operating system. Each database has its own driver, called JDBC driver, for interacting with the database. Database driver is selected via command line interface by entering the type of the database before database address. For windows, there is a driver called JDBC-ODBC bridge, that can be used for all of the three database types. Also a specific JDBC driver for each database type exists. In order to use the JDBC-ODBC bridge to connect to a database, ODBC datasource must first be defined. Instructions for doing this are in appendix 2. The tool uses JDBC-ODBC bridge by default, so the connection is made just by entering an existing ODBC datasource name.

Linux does not have this kind of JDBC-ODBC bridge, so driver type is selected according to the database. To select the driver type, enter the either “PostgreSQL:” or “MySQL:” before the database address.

4.3 Input files

This section lists the details of all input files used by the program. Required input files are database, configuration file, region raster and species rasters.

4.3.1 Output database

The EFISCEN output database (see section 3.5.3) contains the values that are to be disaggregated and serves as input to the disaggregation tool. The table from which disaggregated values are read must include columns “Step”, “SP”, “REG” and “SID”, which are used to specify the right rows to fetch variables from. Any table included in the database can be used, as long as it contains these columns. Notice that the columns to select values do not include “Owner” and “Site”. Values to be disaggregated are summed up by owner and site classes. This is why values to be disaggregated should be totals (and not e.g. per ha).
Example: Base-table contains data by region, owner, site and species and contains SID scenario id and Step. Growing stock results can be disaggregated by specifying the name of the column (“GrStock”) in the configuration file after the name of the table. For more information about configuration file see section 4.3.3.

4.3.2 Map linkage

Two tables need to be included in the EFISCEN output database (section 3.5.3) that link the tree species in the EFISCEN with tree species in raster maps. These tables are called “GISMapping” and “GISSpeciesMap”. The table “GISMapping” must contain at least four database columns (Table 4.1) and the “GISSpeciesMap” has two columns (Table 4.2).

Table 4.1: description of the GISMmapping table stored in the EFISCEN output database

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFISCEN_Species_ID</td>
<td>Integer</td>
<td>Id of the species in Efiscen results</td>
</tr>
<tr>
<td>Species_map_ID</td>
<td>Integer</td>
<td>Id of the species map</td>
</tr>
<tr>
<td>ISOREG</td>
<td>Integer</td>
<td>Region id in the region raster (ISO code*1000+region number)</td>
</tr>
<tr>
<td>Mapping_ID</td>
<td>Integer</td>
<td>Id which groups different mappings together</td>
</tr>
</tbody>
</table>

Table 4.2: description of the GISSpeciesMap table stored in the EFISCEN output database

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Integer</td>
<td>Unique id for this raster (primary key)</td>
</tr>
<tr>
<td>Raster_filename</td>
<td>Text</td>
<td>Filename of species raster</td>
</tr>
</tbody>
</table>

Mapping_ID is used to group mappings together. The mappings that will be used are specified in the configuration file. Mappings with other IDs are ignored during that run.

Species_map_ID is a unique identifier for individual maps. Mappings link species IDs from Efiscen results into these identifiers with the Species_map_ID column of the GISMapping-table.

Example tables:
Tables 4.3 and 4.4 show the GISMapping and GISSpeciesMap tables that contain the required name and the required columns. GISMapping-table also has some other columns that are not used by the program but may be useful information for users of the database.

Table 4.3 defines three relations for country 80, region 1. For mapping ID 2 EFISCEN species with ID 5 is linked to species with ID 1 from the tree species map and EFISCEN species with ID 6 is linked to two tree species IDs 2 and 3 from the tree species map.

Table 4.3: example of the GISMmapping table stored in the EFISCEN output database

<table>
<thead>
<tr>
<th>EFISCEN_Species_ID</th>
<th>Species_map_ID</th>
<th>ISOREG</th>
<th>Mapping_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>8001</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>8001</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>8001</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.4: example of the GISSpeciesMap table stored in the EFISCEN output database

<table>
<thead>
<tr>
<th>ID</th>
<th>Raster_filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FagusSpp.tif</td>
</tr>
<tr>
<td>2</td>
<td>QuercusMisc.tif</td>
</tr>
<tr>
<td>3</td>
<td>QuercusRoburPetraea.tif</td>
</tr>
</tbody>
</table>
4.3.3 Configuration file
A configuration file is used to specify where the variables to be disaggregated are. Values are fetched from the database stated in the command line arguments. The configuration file is used to define the mapping id, SID, step and also tables and columns where data are stored. These define the values that are fetched from the database. The contents of file are as follows: first line contains mapping id, SID and step. After this table and column are entered. Fields are separated by comma (,). Lines starting with number sign (#) are ignored. Only one table - variable pair is supported.

The configuration file has the format as shown in Box 4.1

<table>
<thead>
<tr>
<th>Mapping_id, SID, step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table, column</td>
</tr>
</tbody>
</table>

Box 4.1: format of the configuration file for the disaggregation tool

Table and column specify where the values are in the database. Disaggregation works with data that is by region and species and has scenario id and step. If the data in the table is further divided in other ways, it does not affect the functioning of the tool.

An example of the configuration file is shown in Box 4.2. The example will make the program use mappings with Mapping_ID of 2, and the results to disaggregate are stored in the GrStock columns of table Base in the database. The results must have SID of 104 and Step 2010.

<table>
<thead>
<tr>
<th>#comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 104, 2010</td>
</tr>
<tr>
<td>Base, GrStock</td>
</tr>
</tbody>
</table>

Box 4.2: example of the configuration file for the disaggregation tool

4.3.4 Region raster
A region raster should be provided in raster format. The supported file format is *tiff*. Region raster must contain a spatial reference but it does not have to be the same as the species rasters. The pixel resolution and pixel size can be different and the tool will automatically convert it into the format of the species maps (see section 3.4.). Accuracy can be lower this way. The regions in the raster are identified with an ID that is a combination of country ISO-code and region id. IDs must be numbered the following way: country ISO-code * 1000 + region number) in order to correctly match with the region in the EFISCEN output database.

4.3.5 Species rasters
Tree species rasters are supposed to be in a specific format and have certain spatial references:

The tree species maps should be provided in raster format. The supported file format is *tiff*. Raster are required to be projected as follows: ETRS89 Lambert Azimuthal Equal Area projection.

Parameters:
- False_Easting: 4321000.0
- False_Northing: 3210000.0
- Central_Meridian: 10.0
- Latitude_Of_Origin: 52.0
- Linear Unit: Meter (1.000000)
- Angular Unit: Degree (0.017453292519943299)
- Datum: ETRS89 (European Terrestrial Reference System 1989)
- Pixel size is 1000,-1000.

The tool can use any tree species map if specified in the correct format. For example, species rasters for use with the tool are available at: [http://www.efi.int/portal/virtual_library/information_services/mapping_services/tree_species_maps_for_european_forests/](http://www.efi.int/portal/virtual_library/information_services/mapping_services/tree_species_maps_for_european_forests/).

### 4.4 Tool execution

Arguments to run the tool are provided through command line interface. This includes database address, input file paths, configuration file path, output file path and some optional arguments. This section lists all the arguments that can be used to affect program behaviour. Some arguments for the Java Virtual Machine (JVM) may be necessary for the program to operate correctly.

Different arguments are separated by space. If filepaths have spaces in them, quotation marks can be used to enter them (example: in:“C:\input\input file.tif”).

To run the tool, there are 7 arguments that can be provided. All of these are necessary. There are also 3 optional switches. All are described in Tables 4.5-4.7.
### Table 4.5: overview of arguments to execute the disaggregation tool

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Required/optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database address</td>
<td>Address to the database. Either a path to a file, URL to external database or on Windows operating system a name of a datasource. Database name can contain a prefix to specify the type of the database. Valid prefixes are “odbc:”, ”mysql:” and “postgresql:”. If no prefix is specified then odbc-driver is assumed.</td>
<td>Required</td>
</tr>
<tr>
<td>Path to configuration file</td>
<td>File path to configuration file.</td>
<td>Required</td>
</tr>
<tr>
<td>Region raster file path</td>
<td>Path to the region raster file.</td>
<td>Required</td>
</tr>
<tr>
<td>Species folder or shares file path</td>
<td>Path to the species raster folder or to the raster file containing shares. In case shares are used as an input, prefix “s:”, followed by the share file name can be entered instead. If prefix is not set it is assumed to be species rasters. Prefix for species rasters is “r:”.</td>
<td>Required</td>
</tr>
<tr>
<td>Output file</td>
<td>File path including the name of the output file that will be produced by the program. Aggregated values or shares will be saved to this file. In case shares are saved as an output prefix “s:” must be provided. If no prefix is entered, it is assumed to be result raster. Prefix for result raster is “r:”.</td>
<td>Required</td>
</tr>
<tr>
<td>Database username</td>
<td>User name for the provided database. If the database has a default user this may be omitted.</td>
<td>Optional</td>
</tr>
<tr>
<td>Database password</td>
<td>Password for the database. If no password is set for the database, then this can be omitted.</td>
<td>Optional</td>
</tr>
<tr>
<td>Debug switch</td>
<td>This is an optional argument that can be entered after all the other arguments. If “-debug” is entered results will be aggregated back and printed so their validity can be checked. Shares will also be calculated to be accurate and memory consumption will be printed. These operations will take some time, so this argument can be left out, in which case the checks are not made.</td>
<td>Optional</td>
</tr>
<tr>
<td>Compression switch</td>
<td>In case the system runs out of memory while loading the input raster files to program, this switch can be used to reduce the memory consumption. In case “-compress” is entered as one of the arguments, rasters are processed in a compressed form, rather than trying to process the entire raster at the same time. This method provides the same results as without the compression, but this may reduce the memory consumption. In case the system runs out of memory when loading the input files, this switch can be used. See also below how to tell the java virtual machine to allocate more memory for the program.</td>
<td>Optional</td>
</tr>
<tr>
<td>Memory information</td>
<td>If entered, then the program will display information about memory consumption when it is running. This will not affect performance of the tool.</td>
<td>Optional</td>
</tr>
</tbody>
</table>

In case ODBC drivers are used to connect to database, a configured ODBC data source name can be entered as the database address. PostgreSQL and MySQL database address is entered in following form: //database_address/database_name.
Table 4.6: details on required arguments to execute the disaggregation tool

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>Value</th>
<th>Default</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database address</td>
<td>db</td>
<td>Database_type: String</td>
<td>odbc</td>
<td>db=mysql:efidb</td>
</tr>
<tr>
<td>Configuration file</td>
<td>config</td>
<td>Path to file</td>
<td>--</td>
<td>config=C:\conf.txt</td>
</tr>
<tr>
<td>Region raster file</td>
<td>regionraster</td>
<td>Path to folder</td>
<td>--</td>
<td>regionraster=C:\regionraster.tif</td>
</tr>
<tr>
<td>Species raster folder or shares</td>
<td>in</td>
<td>Path to file or folder</td>
<td>Species rasters</td>
<td>in=C:\speciesrasters</td>
</tr>
<tr>
<td>Output file</td>
<td>out</td>
<td>Path to file</td>
<td>Result raster</td>
<td>out=C:\output.tif</td>
</tr>
</tbody>
</table>

Table 4.7: details on optional arguments to execute the disaggregation tool

<table>
<thead>
<tr>
<th>Description</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database username</td>
<td>user</td>
</tr>
<tr>
<td>Database password</td>
<td>pass</td>
</tr>
<tr>
<td>Debug switch</td>
<td>-debug</td>
</tr>
<tr>
<td>Use compression</td>
<td>-compress</td>
</tr>
<tr>
<td>Display memory information</td>
<td>-meminfo</td>
</tr>
</tbody>
</table>

Example usage:

These parameters run the program with only the required parameters. Species rasters are used as input and disaggregated results are printed to “C:\output\” with a name “out.tif”. This approach does not tell Java to allocate more memory which may result in running out of memory on some computers. In the example database does not have username and password set so they are omitted from the arguments.

```java
Java -jar Efiscen_tools.jar  
  db=postgresql://localhost/EFISCEN_db 
  config=C:\configuration.txt 
  in=r:C:\speciesrasters\ 
  out=C:\output\out.tif 
  regionraster=C:\rasters\regions.tif
```

These parameters run the program with species raster inputs and outputs disaggregated result to a raster file. Java is told to allocate 1.2 gigabytes (1200 megabytes) of memory for the program, this may help to alleviate running out of memory on a 32 bit Operating system. Database type is not specified, so odbc-driver is used. Compression is on to save memory. Information about memory consumption is printed.

```java
Java -Xms1200m -jar Efiscen_tools.jar  
  db=exampledb user=db_user pass=db_password 
  config=C:\configuration.txt 
  regionraster=c:\rasters\regions.tif 
  in=r:C:\speciesrasters\ 
  out=r:C:\output\out -compress -meminfo
```

These parameters run the program with shares as input and disaggregated result outputs. This approach does not tell Java to allocate more memory which may result in running out of memory on some computers. In the example database does not have username and password set so they are omitted from the arguments.
Java -jar Efiscen_tools.jar db=postgresql://localhost/EFISCEN_db
cfg=C:\configuration.txt in=s:\sharesRaster.tif out=C:\output\out
regionRaster=C:\rasters\regions.tif

On some systems it may be necessary to tell the virtual machine to allocate extra memory
for the program. This can be done by entering “-Xmx” after Java, followed by the needed
memory in megabytes. For example, the command -Xmx1200m tells the JVM to allocate
1200 megabytes of memory. On 32bit computers 1200-1500 megabytes is the maximum
possible amount of memory. If this is not enough, see compression switch below.

- -Xms switch can be specified to make JVM allocate a certain amount of memory
when the program starts. The format is -XmsNumberUnit, where number is an
integer and unit is one of $m$ for megabytes, $g$ for gigabytes. NOTICE: there are no
spaces between. If the program runs out of memory during execution this may help.

- -Xmx switch specifies the maximum amount of memory allocated during the
execution of the program. The format is -XmxNumberUnit, where number is an
integer and unit is one of $m$ for megabytes, $g$ for gigabytes. NOTICE: there are no
spaces between. If the program runs out of memory during execution this may help.

4.5 Output

The result raster will contain disaggregated Efiscen results for the data specified in the
configuration file. The resulting GeoTiff raster will have the same spatial reference that
species rasters are expected to have (see section 4.3.4). The extents will be the same as
those in the region raster that was used.

Shares will be saved to a file and can be loaded again during another run of the program.
Shares will be written for regions covered by the region raster used and which have
mappings for at least one species. When using previously saved shares it is important that
the same region raster is used as when the shares were written. Shares are stored in
GeoTiff-format in a single raster file containing multiple bands. The spatial reference and
extends will be the same as the species rasters.

5 Results availability

When conducting simulations, results are immediately available to the user as text files or
stored within a database. Simulation results from previous applications are described in a
range of publications. For a complete overview, see the EFISCEN webpage
(http://efiscen.efi.int). A database for results from previous model simulations does not
exist, but results are mostly available upon request. Results from the European Forest
sector Outlook Study II are available at the national level for individual countries
(http://www.unece.org/efsos2).
6 Final remarks

The EFISCEN model has been used for over two decades to provide insight in European forest resource development, woody biomass availability and ecosystem service provisioning. The model has been validated by (i) comparing its growth functions against growth functions of other models (Sterba 2003), (ii) comparing its projections against other projections carried out for the same forests (e.g. Nilsson et al. 1992; Böttcher et al. 2012), and (iii) running the model on historic data and comparing the output to present day forest state. Separate validation studies have been conducted for Finland with EFISCEN 1.0 (Nabuurs et al. 2000) and for Switzerland with EFISCEN 2.2 (Thürig and Schelhaas 2006).

EFISCEN 1.0 was applied to Finland for the period 1921-1990. The initial situation was based on the results of the first national forest inventory, carried out in the years 1921-1924. Simulation results were then compared to corresponding later inventories. This validation study showed that projected forest resource development was able to capture observed forest resource development for 50-60 years. Differences in increment level were found after 30-40 years, with visible consequences in growing stock levels after 50-60 years. Another conclusion of the validation was that the results at a national level were fairly close to observed levels, but that results at a lower level (per species and region) showed much more deviation. An important reason for this seemed to be differences in management intensity between regions and between tree species.

EFISCEN 2.2 was validated for Switzerland. Firstly, the matrix as initialised by EFISCEN was compared with the original plot data from the second Swiss National Forest Inventory. The result of this comparison was in general satisfying. The largest deviations occurred in poor sites in the Alps region. This was attributed to the fact that forests on such sites usually have a protective rather than a productive function. Such stands are generally managed in an uneven-aged way, leading to a different distribution of growing stocks over age than in case of forests that are managed in a truly even-aged way. Secondly, the forest resource development was projected for the canton Bern for different sets and time spans of inventory plots where repeated measurements were available. On the aggregated (national) level the model produced results comparable to the observed values. However, at a more detailed (forest type) level results deviated sometimes considerably, due to differences in management intensity between regions and between tree species. Moreover, it was concluded that the current structure of the model is not suitable to simulate the uneven-aged, selective management as practised in the Alpine region.

EFISCEN has been designed to address managed, even-aged forests. Deviations from this situation will lead to less reliable results. Further, results on a regional level are less reliable than those on the country level. This is due to inaccuracies in the definition of the management regimes, leading to differences in the regional allocation of the harvest. If reliable results are required on the regional level, more attention should be given to an exact representation of harvest practices. Alternatively, required harvest levels should be specified at the regional level. Probably the same is valid for different owner categories, tree species and site classes. One outcome of the Finnish validation (Nabuurs et al., 2000) was a decrease in increment levels after 30-40 years in EFISCEN 1.0. In later versions a mechanism has been introduced to counteract this. Increment levels seem to be more realistic for longer timeframes now. However, a new validation would be needed to determine appropriate time horizons for reliable projections. Despite this, the current practice of 50-60 years projection horizons seems to be well defendable.
7 References


Section 2: ORCHIDEE manual

1. Set-ups

ORCHIDEE is the land surface model of the IPSL (Institut Pierre Simon Laplace) Earth System Model. Hence, by conception, the ORCHIDEE model can be run coupled to a global circulation model (Fig 1.1a). In a coupled set-up, the atmospheric conditions affect the land surface and the land surface, in turn, affects the atmospheric conditions. Coupled land-atmosphere models thus offer the possibility to quantify both the climate effects of changes in the land surface and the effects of climate change on the land surface. However, when a study focuses on changes in the land surface rather than on the interaction with climate, ORCHIDEE can be run off line as a stand-alone land surface model (Fig 1.1b). The stand-alone configuration receives the atmospheric conditions such as temperature, humidity and wind, to mention a few, from the so-called ‘forcing files’. Unlike the coupled set-up, which needs to run at the global scale (but with the possibility of a regional zoom), the stand alone configuration can cover any area ranging from the global domain to a single grid point.

Figure 1.1: Conceptual differences between (A) a coupled simulation and (B) an off-line simulation. Note the same model is used and the difference is in the interface and the source of the forcing data. In the coupled set-up the interface supports a two-way interaction and the climate conditions are calculated by a global circulation model. In the off-line set-up a one-way interface is used and the climate conditions are read from forcing files.
Although the ORCHIDEE code is the same for the coupled and the off-line set-up, the coupled set-up includes the code of the atmospheric model. Hence, the installation and configuration of the model are different. In the remainder of this document, the instructions refer to the off-line set-up. It is highly recommended to gain experience with the off-line set-up before moving towards the coupled set-up. Nevertheless, detailed instructions to install the coupled model can be found at:

A manual to configure a pre-industrial coupled set-up can be found at:

A manual to configure a present day coupled set-up can be found at:

The settings for automated post-processing of the coupled simulation results are described at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide/LaunchATLAS

2. Modelled processes

The processes included in the early versions of ORCHIDEE were aimed at quantifying the terrestrial water (Fig. 2.1a) and the energy balance (Fig. 2.1b). Later versions of the model were extended with biogeochemical processes (Fig 2.1c), and the current version simulates the interference of anthropogenic activities with natural biogeochemical processes (Fig 2.1d). The biophysical process includes latent (denoted 'evaporation and transpiration' in Fig 2.1a or LE in Fig 2.1b), sensible (denoted H in Fig 2.1b), and kinetic energy exchanges at the surface of soils (G in Fig 2.1b). Heat dissipation and water fluxes are vertically distributed in the soil ('drainage' in Fig. 2.1a and G in 2.1b) and the runoff (Fig 2.1a) is collected in rivers and lakes. The simulated processes that affect the global carbon cycle (Fig 2.1c) include photosynthesis, carbon allocation, litter decomposition, soil carbon decomposition, maintenance and growth respiration and vegetation dynamics. The anthropogenic interference (Fig 2.1d) includes land cover changes, fire, crop irrigation, and forest and grassland management.

The structure of the code including a detailed call of the different modules and subroutines is documented in a pdf and can be downloaded from (895 pages):
http://dods.ipsl.jussieu.fr/orchidee/DOXYGEN/documentation_reduced_2425.pdf
Figure 2.1: The main processes simulated in ORCHIDEE. The processes are grouped in terms of the water balance (a), the energy balance (b), the biogeochemical processes (c) and anthropogenic processes (d). Although the water and energy budget were separated in this presentation, both need to be run at the same time. These processes are coded in a group of modules called ‘Sechiba’ and are the backbone of ORCHIDEE. In addition to Sechiba, the biogeochemical code and the anthropogenic code, grouped in modules called ‘Stomate’, can be activated. Several individual processes can be switched on or off, so supporting a wide range of model set-ups.

3. Input data

ORCHIDEE calculates its prognostic variables (i.e. a multitude of C, H\(_2\)O and energy fluxes) from the following environmental drivers: air temperature, wind speed, solar radiation, air humidity, precipitation and atmospheric CO\(_2\) concentration. In off-line mode, the user should provide these drivers, and ORCHIDEE simulates their impact on ecosystem production, ecosystem respiration, the energy budget of ecosystems, the soil water budget and the surface run-off at a wide range of temporal and spatial scales (Fig. 3.1). When coupled to an atmospheric model, ORCHIDEE follows an implicit approach. The prognostic variables of ORCHIDEE at each time-step are therefore simultaneously calculated with the atmospheric drivers in the planetary boundary layer (Fig. 1.1a). For both setups, the user needs to provide files describing the boundary conditions, namely the (initial) vegetation distribution (that may change with time if the dynamic vegetation module is activated) and a soil map. When the biogeochemical processes are not activated, the leaf area index
(LAI) of the vegetation is read from a map, which needs to be provided by the user. When the runoff is routed through rivers and lakes, a basin and a floodplain map are required. Finally when irrigation is applied, it needs to be prescribed by an irrigation map.

When working on one of the default servers, the input data are accessible to the user. Stand-alone installations do not come with the input data because those data come with a variety of use policies. The spatial and temporal resolution of the different climate drivers is detailed at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/Forcings#a1.WeatherData.

A description of the other input data, vegetation map, soil map, slope and irrigation maps is given at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/Forcings

**Mandatory maps**

- Vegetation
- Soil types and color
- Long term temperature

**Optional maps**

- LAI
- Basins
- Irrigation
- Floodplains

**Figure 3.1:** Mandatory (top row) and optional (bottom row) boundary conditions that need to be provided for ORCHIDEE simulations. The vegetation map specifies the location and share of the different meta-classes (13 in total), the soil types and colour map is required for soil water and energy computations as well as the calculation of the bare soil albedo. The long term temperature is required for phenological initializations. The optional maps are needed only when specific modules are to be used. The LAI map is only required if ORCHIDEE is run without the carbon cycle.
4. Spatial conceptualization

Although ORCHIDEE does not enforce a spatial or temporal resolution, the model does use a spatial grid and equidistant time steps. The spatial resolution is an implicit user setting that is determined by the coarsest resolution of the forcing data (Fig. 1.1b) and the boundary conditions (Fig. 3.1): the vegetation distribution (default 5km×5km), climatological forcing data (default 1°×1°) and the soil map (default 0.5°×0.5°). If higher resolution drivers are available, the model can then be run at that scale. If site-level drivers are available, then simulations at the site scale are feasible. When ORCHIDEE is used in a coupled set-up the grid applied by the atmospheric model determines the spatial resolution of the surface layer model (Fig 1.1a). For global simulations a typical grid is 2.5° latitude by 3.0° longitude but zoomed grids with a 0.5° latitude by 0.5° longitude grid over a region of interest is supported.

The standard version of ORCHIDEE builds on the concept of meta-classes to describe vegetation distribution. By default, it distinguishes 13 such meta-classes (one for bare soil, eight for forests, two for grasslands and two for croplands). Each meta-class can be subdivided in an unlimited number of Plant functional types (PFTs). By default, each meta-class has a single PFT. Biogeochemical and biophysical variables are calculated for each PFT (Fig 4.1a), where most of the biogeochemical variables are reported at the PFT level, the biophysical variables are aggregated at the pixel level because the atmospheric model does not distinguish PFTs and hence its spatial resolution is limited to the pixel scale. For the water and heat balance of the soil, three soil columns are distinguished: one containing each meta-class that includes forest, one for every meta-class with grass and crops and, finally, one for bare soils. Water and heat related soil variables are calculated separately for each column.
Figure 4.1: Spatial conceptualization of ORCHIDEE. ORCHIDEE is run at a regular grid, hence, its basic spatial unit is a grid cell, a grid cell having a known longitude and latitude. (a) The vegetation within a grid cell can be composed of 13 different meta-classes and an unlimited number of PFTs (Plant Functional Types) within each meta-class. Meta-class and PFT have no known location within a grid cell - they simply take up a share of the grid cell. The biogeochemical soil processes such as litter decomposition follow the meta-class based subdivision. (b) A similar concept is used to model the biophysical soil processes such as soil water and heat storage, however, three columns cutting across several meta-class are now used. One column is saved for the soil under tall vegetation (i.e. all meta-classes containing forest), one for short vegetation (i.e. meta-classes that contain grass and crops) and one for soil without vegetation. The three different soil columns do not have a known location within the grid cell.
5. Temporal conceptualization

ORCHIDEE can run on any temporal resolution, however, this apparent flexibility is rather restricted as the processes are formalized at given time steps: half-hourly (i.e. photosynthesis and energy budget), daily (i.e. net primary production) and annual time step (i.e. vegetation dynamics) (Fig. 5.1). Hence, meaningful simulations have a temporal resolution of 15 minutes to one hour for the energy balance, water balance and photosynthesis calculations.

The temporal conceptualization can be set in the run.def file which is the file that contains all none-default settings for ORCHIDEE. Although large time steps result in a substantial gain in model speed, many processes are non-linear (for example, photosynthesis, leaf temperature, groundwater content, snow smelt) and therefore their temporal resolution has important consequences for the numerical stability of the model.
6. ORCHIDEE for TREES4FUTURE

As a contribution to Trees4Future the ORCHIDEE-CAN version (revision 2566) of the global land surface model ORCHIDEE was recently developed, parameterized and validated to simulate the biogeochemical and biophysical effects of forest management over Europe (K Naudts et al., 2015). The model calculates plant photosynthesis, plant growth, mortality, stand structure, forest management, soil carbon dynamics, soil hydrology, soil thermodynamics and biophysical interactions between the land surface and the atmosphere. The model has been parameterized and validated over Europe for Betula sp., Fagus sylvatica, Pinus sylvestris, Picea sp., Pinus pinaster, Quercus ilex and a group combining Quercus robur and Quercus petraea. For Pinus sylvestris, Picea sp. and Betula sp. An additional distinction between boreal and temperate forest was made for the species map and parametrisation: trees located in Norway, Sweden and Finland were considered boreal, while trees growing at lower latitudes were categorised as temperate. Given the potential role of tree species of the Salicacea genus in short rotation coppice management, a separate PFT was parametrised for Populus sp. Furthermore, to improve the parametrisation of the MTC of boreal needleaved deciduous forest, observations from Larix sp. were included when possible.

The default vegetation distribution map in ORCHIDEE, i.e. Olson et al. (1983), was replaced by an up-to-date global MTC map which has been produced using the ESA CCI ECV Land Cover map (http://www.esa-landcover-cci.org/) (Poulter et al., 2015). For the European domain, the global MTC distribution was overlaid by a tree species distribution map (Brus et al., 2012). For the purpose of the TREES4FUTURE project, ORCHIDEE-CAN can account for different species, provenances and genotypes to project the evolution of the forest resources, forest structure, its water use, and its impact on the climate as a function of changing atmospheric CO2 concentrations, changing climate and changing growing conditions such as droughts and heat waves.

7. Practical information for new users

7.1 General information

The model is intentionally distributed without a graphical user interface. Potential users need to reserve adequate time to learn the run environment and the model code. ORCHIDEE is a research tool and is distributed without warranty and user support. User support is only provided through scientific collaborations. For all users a FAQ page with practical tips and solutions is available from: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide.

The instruction provided below assume that the user obtained an account on one of the default servers. Such an account can be obtained through collaboration with the ORCHIDEE-team. If the model is to be used outside such a collaboration, a stand-alone set-up will be required. On a different operating system, the instructions will no longer be accurate but can still serve as a guideline.
7.2 The model code

The model code is distributed under the CeCILL license (https://en.wikipedia.org/wiki/CeCILL) and model use if free for academic research. A password to download the code can be obtained from: http://forge.ipsl.jussieu.fr/orchidee.

The principles underlying the model code are described in (Krinner, 2005; K Naudts et al., 2015; Kim Naudts et al., 2016). The details of the implementation are documented in the code itself. Over 30% of the content of the scientific modules of the ORCHIDEE-CAN code is documentation. Integrating the documentation into the code has the advantage that the documentation is up-to-date with the code.

7.3 The run environment

The official releases of ORCHIDEE are distributed with a run environment called libIGCM. The run environment contains all the files to set-up both coupled and off-line simulations as well as the compilation files for the servers on which ORCHIDEE is routinely run (http://forge.ipsl.jussieu.fr/libigcm). Permission to make use of these servers can typically be obtained (but not guaranteed) through collaboration with ORCHIDEE developers. The model can run outside this run environment but the user then needs to compile the model him/herself and will need to write customized scripts to guide the simulation. Single-pixel off-line simulation can be run on a desktop (1/12 hour per decade). Global one degree off-line simulations require a small cluster (400 cpu hours per decade), coupled simulations require a large cluster (4000 cpu hours per decade)

Download first modipsl and explore what is inside. modipsl contains some tools in the directory util. In util, scripts are found for: extraction (model, mod.def), creation of make_les (ins make, AA make.gdef), creation of job (ins job) and some more. modipsl is also an empty file tree that will receive the models and tools.

Start by extracting modipsl in a new directory:

mkdir MYFIRSTTEST ; cd MYFIRSTTEST
svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
cd modipsl/util
ls

The script model is used to download a specific predefined configuration with the model sources and tools needed. The script uses the file mod.def that contains specifications for each configuration predefined. Use ./model -h to see all existing configurations and ./model -h config name for information about a specific configuration. Same information can be found in the file mod.def.

Download a configuration using ./model config name.

In this example use the configuration ORCHIDEE trunk which is an offline set up using the latest version of the trunk ORCHIDEE. Open mod.def and look at lines begging with ORCHIDEE trunk and then extract as follow:

vi mod.def # Explore lines beggining with ORCHIDEE_trunk
Now explore the directories in modipsl. You will find all source code for ORCHIDEE in directory modipsl/modeles. You also find the directory IOIPSL and XIOS which are fortran and C libraries linked to ORCHIDEE for input/output issues. In directory modipsl/config/ORCHIDEE OL you find scripts to run ORCHIDEE using libIGCM. libIGCM is a tool developed at IPSL to run coupled and off-line simulations. Specific training session about libIGCM are given by the Platform group at IPSL.

Go into each directory and check which versions have been extracted. You're supposed to find the same information as you can see in mod.def. Use svn to check the version and revision number.

```
cd ../modeles/ORCHIDEE
svn info
cd ../modeles/IOIPSL
svn info
cd ../../libIGCM
svn info
```

The makefile was created automatically in the end of the script model, done by the script `ins_make`. Ins_make will detect at which machine you are working on and create adjusted makefiles. By default ins_make recognize the following machines: curie at TGCC, ada at IDRIS and obelix at LSCE. Ins_make can also be re-launched manually. For example this is needed if you move the modipsl directory or if you want to create makefiles for another target machine. The main makefile is found in modipsl/config/ORCHIDEE OL directory.

Now compile the model:

```
cd ../config/ORCHIDEE_Ol
make
```

When the compilation is finished you will find the executables orchidee ol, teststomate and forcesoil in modipsl/bin.

It is possible to install ORCHIDEE at a local linux PC or laptop. We recommend to use the compiler gfortran (or ifort). First you need to install netcdf and compile it with gfortran. Netcdf must be compiled with the same compiler as you will use to compile the model. The file modipsl/util/AA make.gdef contains compile options for different target platforms. Modify the section begging with gfortran at least for the path to your netcdf library. Also change the path to your netcdf library in ORCHIDEE/arch/arch-gfortran.path. Re-generate the main makefile using `./ins make -t gfortran` and compile as normal.

More instructions on the installation of the model can be found at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide#InstallationFirstrunBranch

The basics of svn are detailed at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide#svn

Examples of how to use the libIGCM run environment are given at:
7.4 Parameter settings

The model is distributed without the driver and boundary files because these files are subject to the user policies imposed by the different data owners. When working on one of the default servers, the driver and boundary files are accessible. The model is distributed with default parameter settings for the 13 MTC covering the main tropical, temperate and boreal forest, grassland and cropland biomes. Species-specific parameter settings are not included in the standard distribution but can be obtained by contacting (K Naudts et al., 2015). Users can expand the capacity of the model by parameterizing their own PFTs, species, provenances or genotypes for which ecophysiological parameters are available.

To run the model you need at least the following files in the run directory:

- orchidee_ol: ORCHIDEE executable
- run.def: parameter text file
- forcing_file.nc: climate forcing variables (this file can have another name, in that case it should be indicated in run.def file)
- PFTmap.nc: vegetation map
- soils param.nc: initialization of soil parameters

Create a new directory outside modipsl to run the model and copy or link the ORCHIDEE executable:

```bash
cd MYFIRSTTEST; mkdir RUN1; cd RUN1
ln -s ../modipsl/bin/orchidee_ol .
```

Create the parameter file by saving the following lines into a _l file named run.def:

```bash
TIME_LENGTH=31D
STOMATE_OK_STOMATE= y
LIMIT_WEST = -10.
LIMIT_EAST = 30.
LIMIT_NORTH = 70.
LIMIT_SOUTH = 30.
```

Copy or link the netcdf files from the shared repository IGCM into your run directory. The location of the shared repository IGCM depends on the machine but the content is synchronized between the different repositories. You can use export if your shell is bash (for shell tcsh: replace export by set):

```bash
# At obelix :
export R_IN=/home/orchideeshare/igcmg/IGCM/
# At Ada:
export R_IN=/workgpfs/rech/psl/rpsl035/IGCM
# At curie:
export R_IN=/ccc/work/cont003/dsm/p86ipsl/IGCM
ln -s $R_IN/SRF/METEO/CRU-NCEP/v5.3.2/twodeg/cruncep_twodeg_1901.nc forcing_file.nc
```
You can use ncdump to see what is in the netcdf files. For example:

ncdump -h forcing_file.nc

Now launch the model:

./orchidee_ol       # or ./orchidee_ol > out_exec

When the model finished correctly, following log message is found in the output text file out orchidee 0000:

END of dim2_driver

The file run.def contains parameters to run the model. A line beginning with a # is a comment. Default values will be used for all parameters not set in run.def. You can find the list of all parameters and their default values in modipsl/modeles/ORCHIDEE/orchidee.default. The simulation length in each execution is given by the parameter TIME LENGTH. In this test case TIME LENGTH is 31 days. It is possible to run one year by putting TIME LENGTH=1Y. It is not possible to run less than one day.

LIMIT_EAST, LIMIT_WEST, LIMIT_NORTH and LIMIT_SOUTH are borders (in degrees) for the horizontal domain to be modeled. The default values correspond to the domain of the forcing file. The difference between EAST-WEST and NORTH-SOUTH must be at least one degree. The model will stop if the domain does not cover any land points with error message:

FATAL ERROR FROM ROUTINE dim2_driver
---> number of land points error.
---> is zero!
---> stop driver

Here is a small example to visualize sechiba_history.nc using ferret.

> ferret
use sechiba_history_0000.nc # read file
sh d # list content in file
shade CONTFRAC # 2D plot of a variable
go land # add contour of continents
shade TEMP_SOL[l=1] # 2D plot of TEMP_SOL for first time step
shade TEMP_SOL[l=@ave] # 2D plot of TEMP_SOL average over all time steps
shade SWDOWN[i=@ave] # zonal plot
plot SWDOWN[i=@ave,j=@ave] # plot mean value over time
quit

Different options to set the variables contained in the output files are given at:
Some tips to use the netcdf format in which all output files are written can be found at: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide#netcdf


The content of the two day training course which is organised for free for all ORCHIDEE users can be accessed through: https://forge.ipsl.jussieu.fr/orchidee/wiki/GroupActivities/Training/ORCHIDEEtrainingcourses

7.5 GUI & support

The model is intentionally distributed without a graphical user interface. Potential users need to reserve adequate time to learn the run environment and the model code. ORCHIDEE is a research tool and is distributed without warranty and user support. User support is only provided through scientific collaborations. For all users a FAQ page with practical tips and solutions is available from: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide.

8. Result availability

Since 2013, the performance of ORCHIDEE, both offline and coupled to the atmospheric model, is systematically evaluated for each milestone in its development. The results of these simulations can be checked at: https://forge.ipsl.jussieu.fr/orchidee/wiki/ReferenceSimulations

Typically, international model inter-comparisons establish an archive in which the simulations of the contributing models are stored. Over the years, ORCHIDEE has contributed to numerous model inter-comparisons including those for the 4th and 5th Assessment Report of the IPCC as well as TRENDY.

9. References

Naudts, K. et al., 2015. A vertically discretised canopy description for ORCHIDEE (SVN r2290) and the modifications to the energy, water and carbon fluxes. Geoscientific Model Development, 8, pp.2035-2065.
Section 3: ToSIA

1. Introduction

1.1 ToSIA - Tool for Sustainability Impact Assessment

The Tool for Sustainability Impact Assessment (ToSIA) is designed to support decision making in relation to the forest-based sector. When ToSIA is utilized in forest-related business and industry, policy makers and researchers are able to analyze impacts of different scenarios compared to the status quo within regional, national, and international levels.

ToSIA analyses environmental, economic and social impacts of changes in value chains. It allows users to analyze various sustainability effects in a balanced and unbiased way. Sustainability impacts are objectively quantifiable by comparing changes between a status quo (=reference) and an alternative (=reference future and/or scenario(s)). ToSIA compares alternative process chains, such as forest value chains or value chains. Impacts are assessed by calculating changes in material flows and indicators of environmental, economic and social sustainability within each chain. Studies can range from local to international assessments, from detailed “real” company applications to a more generic, aggregated level. The amount of detail can be independently chosen according to the requirements of the user.

Scenarios are no predictions, but are used to create a consistent image of a possible future. Each one assumes a distinctly different direction for future developments, including specified drivers, and does not necessarily aim to be realistic. In order to create a scenario in ToSIA, either processes, material flow or indicator values - or a combination thereof - differ from the baseline.

ToSIA tracks material flows from the initialization process(es) throughout the entire value chain, including imports, exports, and losses as defined by the user. Data integrity is ensured by checking that calculated material inputs and outputs for each process are balanced. Material flows are captured in organic carbon content (i.e. tons of carbon) both within the material (e.g. wood) and in area (e.g. hectares), and in the product unit itself (e.g. m³ or tons).

The ToSIA software consists of two separate entities: ToSIA Data Client and ToSIA Engine.

- **ToSIA Data Client (TDC)**: for building chain topologies and feeding indicator values and material flow information into ToSIA database, as well as to create chain and process xml for manual saving and loading into ToSIA Engine. The TDC is implemented in Delphi.
- **ToSIA Engine**: for loading value chain xmls into ToSIA to perform material flow and variant calculations, visualise impact assessments of chain runs in a graphical and numerical presentation, run comparisons between scenarios, run MCA and CBA analysis tools, and export reports on calculated data. The engine is implemented in Java.
The ToSIA Management and User Group (TMUG) has been set up as a mechanism to ensure the continued development of ToSIA and support for its users. Any organisation interested in using and further improving ToSIA is welcome to join (http://tosia.efi.int/tmug.html).

1.2 Existing documentation and changes

1.2.1 General documentation
Scientific papers regarding ToSIA (http://tosia.efi.int/material/publications.html) are listed online and can be sorted into papers describing the functioning of the ToSIA tool Edwards et al., 2011; Lindner et al., 2010, 2012; Palosuo et al., 2010; Pützl et al., 2012; Päivinen et al., 2012; Rosén et al., 2012; Werhahn-Mees et al., 2011) and papers using the tool or discussion its application (Berg et al., 2012, 2016; den Herder et al., 2012; Haatanen et al., 2014; Martire et al., 2015; Tuomasjukka et al., 2013, 2014; Wolfslehner et al., 2012). Materials containing a general introduction to ToSIA and brochures include:

- A short film: http://tosia.efi.int/material/film.html
- EFI Making a difference: http://www.efi.int/files/attachments/reports/efi_mad-esite_sustainability_impact_net.pdf
- EFORWOOD final Report, detailing the functioning and design of ToISA: http://tosia.efi.int/uploads/ToSIAdocuments/Final_print%20version_23205_Eforw-ood.pdf
- Fifty-five deliverables from the EFORWOOD project have been published in EFI's Technical Report numbers 31-86: http://www.efi.int/portal/virtual_library/publications/technical_reports/

1.2.2 Calculation aids and partial models
ToSIA is entirely data-driven, which makes the application of the tool very flexible and at the same time requires the user to supply all data: indicator values and material flow related data (conversion factors for product to process unit and to tons of carbon, product shares, split ratios, initiation values). To simplify this data provision, the following calculation aids and partial models were developed, which are available only to TMUG users, along with the ToSIA software as specified in the TMUG MoU (http://tosia.efi.int/uploads/TMUG%20MOU_Version%20121011.pdf):

- ToSIA Case template_extended. A template to define a case study, scenarios and select indicators
- ToSIA indicator _Sept2015 for pre-selection of indicators
- Formulas for indicator calculation_extended
- Golden rules of flow calculation

1.2.3 Handbooks and Manuals
Manuals and handbooks for the use of ToSIA have been developed as several EFORWOOD deliverables, and complemented and expanded as necessary. They are accessible to TMUG members. The following EFORWOOD deliverables have been combined into the standard “Integrated ToSIA Handbook”:
• D1.4.10: ToSIA functionality and use
• D1.2.5 Attachment: EFORWOOD Database Client User Manual
• D 1.5.7: Documentation of concept, implementation and use of the Multi-Criteria Analysis Software component (ToSIA-MCA) in EFORWOOD
• D1.5.6. Monetary values of environmental and social externalities for the purpose of cost-benefit analysis in the EFORWOOD project
• D 1.1.5: An updated and further elaborated policy database; and
• PD 1.1.9: A tested prototype of policy analysis interface for ToSIA

Indicator calculation is described in detail for all indicators in the “Data Collectioon Protocol (DCP) for Indicators” (http://www.efi.int/files/attachments/publications/eforwood/efi_tr_36.pdf).

1.3 New developments

The TDC and ToSIA Engine are two entities. TDC was developed in Delphi and ToSIA Engine in Java due to historical software development reasons. To improve the compatibility and facilitate future integration of the two tools, the TDC was reimplemented into a Java-based data client software (hereafter referred to as JAVE). The reimplementation also includes a new procedure to directly load data from the ToSIA database into ToSIA. The advantages of direct loading are a faster connection, reduction of data disruption potential, the possibility to load in also other data sources such as the ToSIA-EFISCEN linkage and the possibility to merge ToSIA Client (JAVE) and ToSIA engine into one single software. The old procedure of manually exporting chain and process xmls from the TDC (Figure 1.1) and new direct loading procedure (Figure 1.2) are both in use.

The “Integrated ToSIA Handbook” does not yet include developments that took place during Trees4Future, namely the new Java-based Data Client (JAVE) (Chapter 2) and the “ToSIA-EFISCEN linkage” (Chapter 3). All three developments are described below.
Figure 1.1: Old loading of manually saved xml files

Figure 1.2: New loading by directly connecting to database
2. JAVE: new ToSIA Data Client

As described in Chapter 1 the reprogramming of the TDC into JAVE is an essential pre-requirement to creating one piece of software, and in the long run to maintain the possibility to update and further develop ToSIA without restrictions by commercial software packages or integrated, hidden libraries, as well as to maintain compatibility with the latest software standards such as Java and MySQL databases. The new client JAVE also drew on experiences in using the TDC over the last nine years and so was streamlined and simplified for the benefit of user-friendliness, while maintaining the logic and elements that are recognizable for veteran TDC users.

2.1 Login

All old TDC usernames and passwords are migrated and remain unchanged. However, the addition of a user connection directly to the database server in question requires a new login account (see Figure 2.1 and Table 2.1). The same login information is used as for the ToSIA Engine database connection (as shown in Figure 1.2).

![Figure 2.1: New login to ToSIA Client (JAVE)](image)

Table 2.1: Login specifications to for TDC (old client) and JAVE (new client)

<table>
<thead>
<tr>
<th></th>
<th>TDC</th>
<th>JAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>server ID</td>
<td>193.185.149.19 (EFI external)</td>
<td>193.185.149.20 (EFI external)</td>
</tr>
<tr>
<td></td>
<td>10.100.61.28 (EFI internal)</td>
<td>10.100.61.31 (EFI internal)</td>
</tr>
<tr>
<td></td>
<td>212.96.167.24 (IFER)</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>8080</td>
<td>3307</td>
</tr>
<tr>
<td>server login needed</td>
<td>not required</td>
<td>required</td>
</tr>
<tr>
<td>available databases</td>
<td>drop-down list. All available databases are visible</td>
<td>need to type out database name. List of databases invisible</td>
</tr>
</tbody>
</table>
2.2 Navigation panel

The navigation panel has five sheets with assigned functions (See Figure 2.2, left side top):
- **Chain view**: building of chain topologies, linking of processes with products, and linking products together; indicator data entry.
- **Process view**: define new processes and add process related info such as ID, name, stage, process unit, region and process definition
- **Product view**: define new products and add product related info such as product unit, product definition and conversion factors. This information is alternative sensitive, i.e. each alternative has its own set of conversion factor, while values may vary.
- **Catalog editor**: The catalogue has advanced user functionality. It is here where new alternatives, indicator values, indicators, units and conversion factors, regions and countries, stages and modules are defined. These combinations appear as “pre-defined” options for chain users.
- **User administration**: user log-ins, passwords and rights are set here. User rights are at chain editing (basic), catalog editing (advanced) and admin level.

Some sheets are blocked depending on the user rights.

<table>
<thead>
<tr>
<th>Table 2.2: Specifications of user rights</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chain editing rights</strong></td>
</tr>
<tr>
<td>Chain view</td>
</tr>
<tr>
<td>Process view</td>
</tr>
<tr>
<td>Product view</td>
</tr>
<tr>
<td>Catalog editor</td>
</tr>
<tr>
<td>User administration</td>
</tr>
</tbody>
</table>

2.2.1 Chain view

A value chain, process chain or more specifically a forest-wood-chain (FWC) represents a set of processes by which resources from e.g. forests are converted into services and products. In ToSIA, value chains are dealt with at various levels. The highest level used hitherto is the European FWC which is defined as EU 25 plus Norway and Switzerland (EFTA countries). There are many kinds of value chains at the more detailed levels. They can be geographically defined or linked to the main processing chains (paper, wood-products, bio-energy etc.) or to the topic of the chain (e.g. Reindeer husbandry in Malå).

Chain view is the graphical building of chain topologies. Here processes are linked with products, and products are linked together (left panel). Chain related data is displayed in the right panel and has individual sections for:
Figure 2.2: Chain view in JAVE

**Process properties:** Process related information such as process ID, name, initialisation value, process unit, Country, country group and region, contact and name of person who entered process, module and stage, product shares.

*Modules are organisational entities at the sub-chain level. Modules combine stages and processes together in logical groups (see also Processes and Stages in online glossary http://tosia.efi.int/glossary.html). Modules present the highest hierarchical level of a value chain: Chain > Module > Stage > Process*

From the ToSIA database point of view, the module is just one of the classifiers for the processes. From a ToSIA run point of view, indicator values can be aggregated and compared at chain, module and process level.

**Conversion factors:** display info from product view per activated process

**Indicator data entry:** Indicator data is entered per process and indicator, along with the indicator value (“Mean value”) and metadata like distribution type, representativeness of data quality (3 = high, 2 = medium, 1 = low), Data source (Data from experiments or scientific measures, Expert judgement, branch statistics, Official statistics, Modelling - process models, “will be corrected”, weighting or scaling factors relevant for adoption of generic data for the actual case, modelling - time extrapolation), free form entry for “indicator note” (references and assumptions for calculation) and “algorithm” (formulas used for calculation).
Figure 2.3: Specifications of Indicators in Chain View

Indicator: Indicators have been defined as a parameter which points to or provides information about the state of a phenomenon, environment, area with a significance extending beyond that directly associated with a parameter value (OECD, 1993). It has been seen as a means devised to reduce the large quantity of data down to its simplest form.

Within ToSIA, indicator values per process and per material flow are taken from the database client, where a set of indicators has been selected and filled with values. In ToSIA, the calculated absolute process indicator values are determined based on the material flow through the process multiplied with the relative indicator value per process unit as defined in the database. Calculated module and chain indicator values are then determined by aggregating the calculated process indicator values along the chain taking into account the system boundaries selected by the user.

Flows: flows describe “factors” (formerly known as “split ratios”) when one product (e.g. wood ash) is linked to the next process(es). The standard value is 1, when it is a direct link. If a product goes to several places the sum of the factor values needs to add up to 1.

A factor or split ratio is used to divide one product to many (output) or merge many products to one (input) in subsequent process(es). For more info see "Links".
**Links:** As soon as two processes with their input and output products are defined, the processes are linked with each other. A link symbolizes where the output material flow of one process is going to.

There can be 3 types of links:
- simple 1-to-1 links
- many-to-one (e.g. pulpwood logs from several sources enter one mill)
- one-to-many (e.g. one pulp product is delivered to different types for further usage).

**Chain properties:** describes the chain ID and name, as well as products and processes in use. Here the baseline and associated alternatives, as well as base unit of chain (usually tons of Carbon) can be set.

![Chain properties](image)

**Figure 2.4:** Specifications of Chain Properties in Chain View

In order to edit any data “edit chain” needs to be activated first. This locks the chain so that only one person at a time can alter data, to avoid conflicting data entry from different users. Similarly, once editing is finished, the session needs to be saved and chain editing exited.

**2.2.2 Process view**

**Process (in a FWC); production process:** The smallest and most important element of a FWC is a process. Transformation of energy and materials takes place in a process. In a process (wood) material will change its appearance and/or move to another location. Every process requires inputs and produces outputs. Inputs for each Process in a chain are supplied by outputs of previous Processes. Therefore, in case of the FWC we call inputs and outputs simply Products. Processes include e.g. planting trees, stand treatments,
harvesting, transport, sawing, pulping, papermaking, printing, packaging, recycling, and energy production - or when needed subsets thereof.

Figure 2.5: Specifications of Process View

In the process view new processes are added. Process related info such as ID, name, stage, process unit, region and process definition can be added. For successful ToSIA runs at minimum, stage and process unit need to be defined. Process unit is usually the unit in which reports or statistics are detailed. For example, for harvesting operations it may be costs in EUR per m3, or employment in hours per m3; making m3 the process unit. All further information helps users such as process assumptions, references and specifications.

2.2.3 Product view

Product: Products are the Carbon mass-based inputs and outputs of processes, such as spruce logs or finished wood furniture. The functional purpose of products is to link together processes to form chain structures. Products are expressed in mass units and for each product the conversion factor, for converting it to different units (e.g. tons of Carbon, m3, ha) needs to be stated. Processes can also receive input products from outside of the value chain’s system boundaries (e.g. imported materials or non-wood material used in furniture manufacturing).

In the product view panel new products are defined and product related info is added such as product unit, product definition and conversion factors. This information is alternative
sensitive, i.e. each alternative has its own conversion factor set, while values may vary.

Figure 2.6: Specifications of Product View

**Conversion factor:** Mass in tons of Carbon is used as the information carrier for value chains in ToSIA. Tons of Carbon is used in internal calculations in ToSIA to ensure that all information is comparable, and consistent. Each individual product needs a conversion factor from product unit to process unit, and from product unit to tons of Carbon (=original mass to mass of contained pure Carbon). As a minimum, conversion factors between product to product unit (=1), product to process unit of connected processes, and conversion factor to tons of Carbon are necessary.

### 2.2.4 Catalogue editor

The catalogue panel has advanced user functionality and can only be accessed by users with catalogue editing and admin rights. Here the following combinations can be set that appear as “pre-defined” for chain users:

And **alternative** or **variant** is an alternative value chain in terms of indicator values, volume flow and/or slight changes in topology, and only exist in connection to a baseline case study (= basic chain). Variants exist in **reference futures** and **scenarios**.

**Reference futures** can be calculated and modelled as possible prolongations over time. Reference futures are the basic change over time on top of which scenarios can be built. These reference futures are user-defined, as it suits the requirements of the case study.

A **scenario** is a combination of internal or external drivers and their impacts to the chain. The scenarios result in alternative value chains with different sustainability impacts compared to the current value chains. Scenario impacts can further be evaluated with analysis tools, such as MCA and CBA evaluation methods.

Scenarios can be also combined with reference futures. In that case the reference future describes impacts as they are assumed for future development (compare “Reference future”), the future scenario is calculated as affecting on top of these changes. Scenarios may differ against the background of different reference futures.
Alternatives: The catalogue editor is the place where new years, new reference futures and new scenarios can be added; and where they are combined to alternatives consisting of year + reference future + scenario. For cases when only a year and no reference future and/or scenario is valid, an “empty” reference future and/or scenario needs to be selected.

Figure 2.7: Specifications of Alternatives in Catalog Editor View

Indicator values: Indicator values are numerical expressions of indicators for different processes. How reliable values are and how they were calculated is crucial for the interpretation of the assessment results. This type of information about the value is called metadata. Metadata describing the indicator values is given as pre-set drop-down value alternatives. They are visible in the Chain view panel > Indicators and can be edited there. Currently metadata like distribution type, representativeness of data quality (3 = high, 2 = medium, 1 = low), data source (Data from experiments or scientific measures, Expert judgement, branch statistics, Official statistics, Modelling - process models, “will be corrected”, weighting or scaling factors relevant for adoption of generic data for the actual case, modelling - time extrapolation) are available.
Figure 2.8: Specifications of Indicator values in Catalog Editor View

**Indicators:** Existing indicators can be edited here and new indicators or sub-indicators added. The structure is such that each indicator (top panel) has a main indicator class (e.g. Employment) and below it sub-indicators, complete with a unique indicator ID, name, indicator unit, indicator type and aggregation.

The lower panel allows specifying information for ToSIA calculation, on how to treat the indicator value when aggregating several processes per module or chain.

- **Type:** relative or absolute. Most indicators are relative as an indicator value per process unit. There are exceptions, such as 3.1.1. Import volume states the absolute amount of volume in product unit of the imported product per process. If an indicator value is set as an absolute value it is not multiplied with the material flow in the process, like the relative values are.

- **Category:** Social, environmental, economic. These sustainability categories are used in selecting and narrowing down the presentation of calculated indicator results in ToSIA. The same categories are also used in the MCA tool for weighing user preferences per indicator.

- **Aggregation:** sum, weighted mean, none. Aggregation specifies the calculation routines of how ToSIA shall treat the process-wise indicator values for aggregating to module or chain level. E.g for values expressed as a percentage (e.g. 10.2.2 - Employment male - % of total employment) only a volume-weighted means can give a correct result as opposed to summing up.

- **Unit:** Indicator unit is specific to each subindicator, and can be anything quantifiable. Examples of indicator units (non-exhaustive list) are EUR, fulltime equivalents (FTE), kg, tons, ha, m3, kg of CO2 equivalents, MJ, kWh, kg BOD5, cmol per kg soil, qualitative ratings as oints in a Lipert scale, etc.
Figure 2.9: Specifications of Indicators in Catalog Editor View

Units and conversion factors: ToSIA needs conversion factors between used product units to process units (here “reporting unit”), and to ToSIA’s internal calculation unit “tons of Carbon”. Here is the place to add new process units and conversion factor templates, that will be visible in Product view. It is recommended to keep the number of process units to a minimum.

Figure 2.10: Specifications of Units and Conversion factors in Catalog Editor View

Regions and countries: regional specifications are valuable additional process information. European countries and regions are pre-defined in the ToSIA Client according to NUTS3 level. A generic “anywhere” also exists. In addition to these regions, any other global country-region classification system can be added here (top panel).
Country groups are a ToSIA and user-specific classification of several countries into a countrygroup which can be used to aggregate processes in ToSIA. This is particularly useful for very large chains involving several countries, e.g., like the EU FWC (http://tosia.efi.int/forest-wood-chains/european-fwc.html).

Figure 2.11: Specifications of Regions and countries in Catalog Editor View

**Stages and modules:** Modules are linked with stages in fixed combinations, which are again associated to processes. Modules present the highest hierarchical level of a value chain: Chain > Module > Stage > Process

New modules and stages can be added here.

Figure 2.12: Specifications of Stages and modules in Catalog Editor View

2.2.5 **User administration**

User administration can only be accessed by admin users, as it is here where user log-ins, passwords, and rights are set. User rights are at chain editing (basic), catalog editing (advanced) and admin level.
3. ToSIA - EFISCEN linkage

ToSIA is data-driven, and most forest wood chains (FWC) start with the amount of available forest resources in volume and species on a specific area, or with the amount of timber available for cutting in a specific area. This information is available in EFISCEN, and for more complex chains and multiple scenarios a direct loading of this type of EFISCEN data into ToSIA is very beneficial. Therefore a model linkage between ToSIA and EFISCEN was developed as part of the Trees4Future project.

EFISCEN-ToSIA Mapping Tool is used for linking results from the EFISCEN model to the input data of the ToSIA model. In 2014 it was initially made as a stand-alone pilot version, but its functionality has now been integrated into JAVE. The tool is a graphical user interface-driven program where the user manually makes links between the data. The tool connects to an EFISCEN database and through a tool that is accessed in a pop up window of JAVE and allows for importing and aggregating EFISCEN results to be used as process initialization values in FWCs. A screenshot from the user interface is shown in Figure 3.1.
The EFISCEN-ToSIA mapping allows users to initialize FWC processes from the EFISCEN results. Searches into the results are made by specifying a country, scenario, simulation, year and result table. EFISCEN results are stored in a matrix consisting of region, owner, site and species. The results can be freely aggregated within the matrices of a single country. When a fitting value has been found it can be set as the process initialization value and the process unit can also be set.

The tool is integrated into the new ToSIA Database Client that is used to manipulate ToSIA databases. The user needs to first establish a connection to a database containing EFISCEN results. After that they can initialize any process in the chain that they are editing from any of the results in the database.

4. References


