

IMAGE User Manual

User guide to run the IMAGE program

Background Studies

IMAGE User Manual

User guide to run the IMAGE program

PBL (E.Stehfest, L. de Waal, R. Oostenrijk)
VORtech



IMAGE User Manual

© Netherlands Environmental Assessment Agency (PBL), June 2010
PBL publication number 500110006

Corresponding Author: Elke.Stehfest@pbl.nl, IMAGE-info@pbl.nl

Parts of this publication may be reproduced, providing the source is stated, in the form:
Netherlands Environmental Assessment Agency: IMAGE User Manual.

This publication can be downloaded from our website: www.pbl.nl/en. A hard copy may be ordered from: reports@pbl.nl, citing the PBL publication number.

The Netherlands Environmental Assessment Agency (PBL) is the national institute for strategic policy analysis in the field of environment, nature and spatial planning. We contribute to improving the quality of political and administrative decision-making by conducting outlook studies, analyses and evaluations in which an integrated approach is considered paramount. Policy relevance is the prime concern in all our studies. We conduct solicited and unsolicited research that is both independent and always scientifically sound.

Office Bilthoven
PO Box 303
3720 AH Bilthoven
The Netherlands
Telephone: +31 (0) 30 274 274 5
Fax: +31 (0) 30 274 44 79

Office The Hague
PO Box 30314
2500 GH The Hague
The Netherlands
Telephone: +31 (0) 70 328 8700
Fax: +31 (0) 70 328 8799

Rapport in het kort

IMAGE Gebruikershandleiding

Deze gebruikershandleiding bevat de basis-informatie om het simulatiemodel IMAGE (“Integrated Model to Assess the Global Environment”) van het PBL te kunnen gebruiken. De aanleiding was onder andere een fundamentele herstructurering van de programmatuur, voor IMAGE versie 2.5. Dit document geeft beknopte inhoudelijke informatie over de submodellen, vertelt de gebruiker hoe het programma geïnstalleerd moet worden, beschrijft de directory-structuur van de run-omgeving, geeft aan hoe scenario’s voorbereid en gedraaid worden, en geeft inzicht in de restart-functionaliteit.

Trefwoorden:

IMAGE, gebruiker; handleiding; scenario; restart.

Contents

- Rapport in het kort 5
- Summary 9
- 1 Introduction 11
 - 1.1 Submodel and code structure 11
 - 1.2 Other relevant documents 12
- 2 Brief description of IMAGE submodels 13
- 3 Getting started 15
 - 3.1 System requirements 15
 - 3.2 Installation 15
 - 3.3 How to run an IMAGE scenario 15
- 4 Run environment 17
 - 4.1 Directory structure of run environment 17
 - 4.2 Directory descriptions 17
- 5 Working with IMAGE 21
 - 5.1 Creating a new project 21
 - 5.2 Creating a new scenario 21
 - 5.3 Updating input data 21
 - 5.4 Using model options 22
 - 5.5 The zoom version 22
 - 5.6 Adding new or modified models 22
- 6 Pre-processing and post-processing facilities 23
 - 6.1 The IMAGE macro 23
 - 6.2 Format conversions 23
 - 6.3 The viewer USS (User Support System) 25
- 7 Coupling to other models 27
 - 7.1 IMAGE-TIMER-FAIR 27
 - 7.2 Providing data to GLOBIO 29
 - 7.3 Coupling with agro-economic models 29
 - 7.4 Coupling to LPJ 29
- 8 Restart functionality 31
 - 8.1 Functions 31
 - 8.2 Preparations 31
 - 8.3 Restart run 32
 - 8.4 Nocompute 32
 - 8.5 Readfromfile 33
- Appendix A File structures 34
- Appendix B Calibration 35
- Colophon 37

Summary

This user manual contains the basic information for running the simulation model IMAGE (“Integrated Model to Assess the Global Environment”) of PBL. The motivation for this report was a substantial restructuring of the source code for IMAGE version 2.5. The document gives concise content information about the submodels, tells the user how to install the program, describes the directory structure of the run environment, shows how scenarios have to be prepared and run, and gives insight in the restart functionality.

Keywords:

IMAGE; User Manual; scenario; restart.

Introduction



This user manual contains the basic information to run the simulation model IMAGE.

The name IMAGE means “Integrated Model to Assess the Global Environment”. IMAGE is an ecological-environmental modelling framework that simulates the environmental consequences of human activities worldwide. It represents interactions between society, the biosphere and the climate system to assess sustainability issues, such as climate change, biodiversity and human well-being. The objective of the IMAGE framework is to explore the long-term dynamics of global change as the result of interacting demographic, technological, economic, social, cultural and political factors.

A central part of the IMAGE framework is the IMAGE program (executable), that contains all land and climate related models. This report deals with the IMAGE program, which will be shortened to IMAGE in the rest of this report. One of the reasons for this report was the substantial restructuring of the source code for IMAGE version 2.5. Therefore, the sections on the structure of data and source code are applicable for IMAGE version 2.5 onwards. Also in general, the report reflects the situation in early 2010, and will be updated as required.

1.1 Submodel and code structure

The current version of IMAGE contains a number of submodels, cooperating within a hierarchical structure. This structure is shown in the scheme below; every submodel is called by its owner (parent-child relationship). Chapter 1.2 gives a brief description of each of the submodels.

The const files mentioned with each submodel contain physical constants related to the submodels.

IMAGE.F – main program unit

- AOS – atmosphere ocean system
(const file *aos.const*)
 - ATMOCHEM - atmospheric chemistry model
(const file *chem.const*)
 - ADDEMISS - adds emissions from energy-industrial sources, emissions that are land-use related, and natural emissions.
(const file *emiss.const*)
 - CLIMATE - climate model
(const file *climate.const*)
- OCEAN - ocean model
(const file *ocean.const*)
- RADIAT - radiative forcings
(const file *radiat.const*)
- READEIS - emission scenarios from the energy-industry system
(no const file)
- SLR - sea level rise
(const file *slr.const*)
- WORLDTOZOOM - translates global atmospheric and ocean data to regional data. This submodel is an alternative for all other aos child submodels except readeis.
(no const file)
- TES - terrestrial environment system
(const file *tes.const*; dat file *coptions.dat*)
 - ADMLCM – adm/lcm cluster
(const file *admlcm.const*)
 - ADM - agricultural demand model
(const file *adm.const*)
 - LCM - land-cover model
(const file *lcm.const*)
 - WOOD - land-cover changes due to demand for wood
(const file *wood.const*)
 - DRIVFORCE - driving forces from an external source
(no const file)
 - BIOFUEL - biofuel data from an external model
(const file *biofuel.const*)
 - CCM - carbon cycle model
(const file *ccm.const*)
 - CARBON - carbon model
(const file *carbon.const*; dat file *carbonregrowth.dat*)
 - LUEM - land-use emissions model
(const file *luem.const*)
 - SOIL - land-degradation model
(const file *soil.const*)
 - TVM - terrestrial vegetation model
(const file *tvm.const*)
 - WAT - water model
(no const file)
 - WRITETES - supporting routine for tes
(no const file)
- POSTPR – post-processing
(const file *postpr.const*)

1.2 Other relevant documents

In this report, we also refer to the following reports where additional information can be found:

- IMAGE model website, providing detailed description of the model, publications and references for the entire IMAGE modelling framework (<http://www.pbl.nl/image>).
- IMAGE Programming Guidelines (PBL, 2010)
- Description of the calibration functionality (IMAGEcalibration.doc)
- Description of the excel macro (Excel macro handleiding_NL.doc, and Excel macro handleiding_NL.doc)
- Description on how to create scenario projections (IMAGEscenarioCooker.doc)
- Description of the coupling to GTAP (Description GTAP to IMAGE_*.doc, and (Description IMAGE to GTAP_*.doc)
- Description of the coupling to IMPACT (IMPACT_to_IMAGE_manual.doc).
- Description of the coupling to GLOBIO (IMAGE_to_GLOBIO.doc)
- Description of the coupling to LPJ (IMAGE_LPJ*.doc)
- Description of the IMAGE zoom versions (IMAGE_zoom.doc)
- Description of the IMAGE version control (IMAGE_versionControl.doc)

Links to these documents can be found at the IMAGE project directory, under \Image_Intro\RelevantLinks, or they can be obtained, if relevant, from image-info@pbl.nl.

The IMAGE project directory is 'project\M481508_IMAGEaanpassingEnBeheer', at the moment.

2

Brief description of IMAGE submodels

This chapter gives a very brief overview of the submodels in IMAGE. The submodels are ordered alphabetically. More details about most of the submodels can be found at <http://www.pbl.nl/en/themes/sites/image/index.html>. At this location, also the references and scientific model descriptions can be found.

ADDEMISS

The ADDEMISS model adds emissions of the most important greenhouse gases and other reactive gases to the current atmospheric concentrations. Emissions come from energy and industrial sources as well as land-use related and natural sources. The ADDEMISS model is a submodel of the ATMOCHEM model.

ADM

The ADM model is the Agricultural Demand Model. It is part of the coupled ADM/LCM model combination in the TES model. The ADM computes the regional demand for crops and animal products. The required production is determined by the sum of domestic regional demand and net trade.

AOS

The AOS model describes the atmosphere-ocean system. It contains models on, for example, atmospheric chemistry, climate changes and the ocean carbon balance. The AOS model is one of the two top-level models of IMAGE (the other is the TES model).

ATMOCHEM

The Atmospheric Chemistry Model calculates the concentrations of the most important greenhouse gases and other reactive gases. It is part of the AOS model. In turn, it uses the ADDEMISS submodel to describe the addition of gases through emissions. The ATMOCHEM model computes atmospheric concentrations for carbon dioxide (CO₂), methane (CH₄), carbon oxide (CO), NO_x, SO_x, VOC, nitrous oxide (N₂O), the CFCs, CCs, HCFCs, bromocarbons, PFCs, SF₆ and HFCs. The change in concentrations depends on the change in emissions and on atmospheric removal, as determined by its atmospheric lifetime.

BIOFUEL

The BIOFUEL model determines the demand for modern and traditional biofuels. Traditional biofuels include fuel wood and charcoal. The demand is currently obtained from input files.

CARBON

The CARBON model is the terrestrial carbon model, which computes changes in biomass and fluxes from/to the biosphere due to changes in land cover, CO₂ concentration and climate. In addition, the Surplus Potential Productivity (SPP) for carbon plantations is calculated. The CARBON model is part of the CCM model.

CCM

The Carbon Cycle Model (CCM) simulates the carbon (C) flux between the atmosphere and biosphere, as altered by atmospheric CO₂ concentrations, climate change and different land-cover conversions. The spatial resolution of the calculations is 0.5 degrees latitude by 0.5 degrees longitude, as in most calculations in the Terrestrial Environment System (TES). The CCM model contains the CARBON model and a model for land-cover changes due to the implementation of carbon plantations.

CLIMATE

The Climate Model represents the core model of the Atmospheric Ocean System (AOS). The CLIMATE model converts the radiative forcings as determined by the RADIAT model into temperature changes of the global mean surface and the ocean.

DRIVFORCE

The DRIVFORCE model provides the driving forces for global changes. It is part of the ADM/LCM model combination. The main driving forces are the changes in (urban) population, use of fertilisers, the grazing intensity and the desired self-sufficiency ratios of crops and animal products. The DRIVFORCE model currently obtains the driving forces from input files.

LCM

The objective of the Land Cover Model (LCM) is to simulate global land-use and land-cover changes due to agricultural, livestock and wood demand, by reconciling the land-use demand with the land potential. The basic idea of the model

is to keep changing gridded land cover within different world regions until the total demand for each region is satisfied.

LUEM

The LUEM model combines the Agricultural Waste Burning Emission model (AGWBEM), the Land Use Emissions Model for ANimal and animal waste emissions (LUEMAN), and the actual Land Use Emissions Model (LUEM). It computes the emissions of gaseous pollutants (greenhouse gases, gas species involved in ozone chemistry, aerosol formation and acidifying compounds), stemming from natural and land-use related sources. The terrestrial flux of CO₂ is computed by the Carbon Cycle Model (CCM). Therefore, the land-use emissions model focuses on emissions of other important gases, including: greenhouse gases (CH₄, N₂O); ozone precursors (NO_x, CO, VOC); and acidifying compounds (SO₂).

OCEAN

The Oceanic Carbon Model (OCEAN) simulates the carbon flux between the atmosphere and ocean with the Bern Carbon Cycle model (Bern-CC). This model is a box-diffusion type oceanic carbon model. The model is based on a mixed-layer-pulse-response function, which allows for describing time-dependent non-linear effects of seawater chemistry resulting from changes in the atmospheric CO₂ concentration. This non-linearity occurs in the chemical equilibria involving the buffering system formed by undissociated CO₂ and HCO₃⁻ and CO₃²⁻ ions. The analytical representation of the mixed layer response function of the Princeton 3D model is used.

POSTPR

POSTPR is not really a submodel, but rather a post-processing step for the IMAGE program. It writes copies of grid files to the working directory. There are two categories of files:

- Files that are written unconditionally for the year in which the simulation starts (timestart) and for 1990.
- Files that are written only for years that are specified in the file FILEHAND.DAT.

RADIAT

The RADIAT model sets the radiative forcings due to greenhouse gases, tropospheric ozone, sulphate aerosols, fossil-fuel black carbon and organic carbon aerosols, and biomass burning aerosols. The RADIAT model is called from the AOS model and provides input for the CLIMATE model.

READEIS

The main objective of the READEIS model is to read the regional emissions from energy use and industrial processes of greenhouse gases (CO₂, CH₄, N₂O), ozone precursors (NO_x, CO, VOC) and acidifying compounds (SO₂).

SLR

The impacts of global warming on global sea levels are calculated by the Sea-Level Rise model (SLR). In this model, the total sea level rise is influenced by thermal expansion of the oceans and by the changing net mass balance of glaciers and ice sheets. The most important ice sheets of Greenland and Antarctica are taken into account in the SLR model.

SOIL

The SOIL model computes the land degradation due to erosion. The SOIL model is part of the TES model.

TES

The TES model describes changes in the terrestrial environment. It contains separate models on, for example, agricultural demand, exchange of moisture with the atmosphere, soil degradation, land-cover changes, natural emissions, emissions from agriculture and carbon balances. The TES model is one of the two top-level models of IMAGE (the other is the AOS model).

TVM

The Terrestrial Vegetation Model (TVM) simulates the occurrence and productivity of crops and the distribution of natural vegetation on the basis of climate conditions and soil characteristics on a spatial resolution of 0.5 degrees latitude by 0.5 degrees longitude.

WAT

The WAT model describes the exchange of water between the earth surface and the atmosphere through evaporation and precipitation. The WAT model is part of the TES model.

WOOD

The WOOD model simulates land-cover changes due to the demand for wood. The WOOD model is part of the LCM model.

WORLDTOZOOM

The WORLDTOZOOM model is an alternative implementation of the AOS model. It does not determine all the variables that are computed by the AOS model. The WORLDTOZOOM model only reads the temperature changes since 1990 and the atmospheric CO₂ concentration. The WORLDTOZOOM model is called from within the AOS model, which then omits most of the other submodels. Thereby, a 'Zoom Version' of IMAGE, which does not cover the entire globe, can be run, using climate change information from an earlier, global simulation (see IMAGE_zoom.doc, Section 1.2).

WRITETES

WRITETES is not really a submodel, but rather a post-processing step for the TES model. It performs some additional calculations and writes results to regional output files.

Getting started

3

3.1 System requirements

For an efficient execution, IMAGE needs a platform that satisfies the following requirements:

- Intel PC with Windows XP or Vista.
- Recommended memory size at least 512 MB
- Recommended free disk space at least 3 GB
- Recommended processor speed at least 2 GHz

3.2 Installation

This section describes the installation of the IMAGE run environment for end users, and the installation of source code for programming users.

3.2.1 The IMAGE run environment

To run IMAGE, the user must install the IMAGE run environment on a local workstation. This run environment contains the IMAGE executable and data files to run a simulation example. The example contains the SRES_A2 scenario

The IMAGE run environment is available in the IMAGE version control system. In order to obtain the desired version of IMAGE, a programming user of the IMAGE model has to export the entire directory structure from the repository (see next section), and provide it to the user. The user can then store this exported run environment at a directory of his choice. In this manual, this latter directory is referred to as *user_directory*.

The directory structure of the run environment is described in Chapter 4.

3.2.2 The IMAGE program source code

Programming users have the opportunity to create their own (experimental) version of IMAGE, and contribute to model development. For this purpose, the source code is available for specialists.

The IMAGE program source code is stored in the Subversion-based source control system TortoiseSVN, on a central network drive at MNPBL To use subversion, first install Subversion and Tortoise. To obtain the IMAGE source code, the user can check out the entire source code files from the IMAGE repository (see IMAGE_versionControl.doc, Section 1.2).

The IMAGE source code files must be organised in a directory structure that matches the IMAGE submodel structure. To edit the code and generate a new IMAGE version, the program is advised to use Compaq Visual Fortran Standard Edition vs 6.6C. For more information about IMAGE programming, see the IMAGE Programming Standard.

3.3 How to run an IMAGE scenario

The IMAGE run environment consists of a project directory (<project_name>), which contains all scenarios of this project (<scenario_name>), and all information and data to run these scenarios. For details on the structure of the IMAGE run environment see Chapter 4.

In the IMAGE directory structure, the folder <project_name>\bat (see Section 4.2.2) contains three bat files:

| | |
|-------------------------------|-------------------------------|
| <i>run_image.bat</i> | : to run the IMAGE executable |
| <i>empty_image_output.bat</i> | : to delete old output files |
| <i>empty_image_state.bat</i> | : to delete old restart files |

These three bat files have at least *ScenarioName* as argument. This argument must be equal to the name of one of the available scenario directories in the project folder; for scenarios, see Chapter. When the argument is omitted or is incorrect, the bat file will give an error message and stop.

The file *run_image.bat* has an optional second argument: *ExecutableName*. When the user wants to run a scenario with an alternative version of IMAGE, he must create the appropriate executable, give it a special name, and place it in the *bin* directory next to the standard executable. When calling *run_image.bat*, the user must pass the special executable name as the second argument. When omitted, the bat file will choose the standard executable. When the name is incorrect, the bat file will give an error message and stop.

To run a scenario, the user must prepare control and input files, as described in Chapter 4, and then call the runsript *run_image.bat*. When there has been a previous run, the old output and restart files may be erased beforehand by calling the scripts *empty_image_output.bat* and *empty_image_restart.bat*.

When running a scenario, IMAGE places the output files in the appropriate output directories, see Chapter 4. The run messages will be stored in the file *log_image.txt*; this file

contains details about the progress of the simulation run. It is stored in the folder *docu*. Sometimes, the subroutine *lintpol* (for linear interpolations) produces special error messages; these are stored in a file called *lintpol_error.log*, which is also placed in the *docu* folder.

To identify the exact program version with which the scenario has been run, version information is stored in the logfile. This version information consists of three components:

- The SVN version number of the program.
- The SVN version number of every called subroutine.
- The date and time stamp of the IMAGE executable.

Note: When a programmer creates his own version of IMAGE by making adaptations to the source code, and does not commit this code back into the TortoiseSVN repository, the SVN version information will not be up-to-date. However, the date and time stamp will be up-to-date.

Run environment

4

4.1 Directory structure of run environment

The installation of IMAGE creates an environment consisting of a structure of directories that contain all files that are related to IMAGE runs. These files are control files, input files, intermediate files, output files, and restart files.

This directory structure is as follows:

```
<user_directory>
  <project_name>
    bat
    bin
    data
      const
      GCM
      unf
    <scenario_name>
      CPCC (optional)
      dat
      input
      map
      output
      outputcell (optional)
      outputrest
      state (optional)
        {structure of subfolders}
      work
    start
```

A description of each directory is found in the following section.

For almost every subdirectory a keyword is defined with the appropriate pathname in the file *datapaths.dat*, see Section 4.2.11. These paths are relative to the directory *work*, in which the IMAGE executable runs.

4.2 Directory descriptions

4.2.1 <user_directory>

This directory is the user-defined folder under which the IMAGE run environment is installed. Locally at PBL, this is “\data\Image\Scenario_lib”

4.2.2 <project_name>

This directory encompasses a run environment for one particular project. A project comprises a series of scenarios

that are related to each other and use identical information for the data provided in *data* and *start* directories. Mostly, they should use identical executables, and should be based on the same calibration and, therefore, on historical input data. The directory name should identify the project involved.

After installation, the example project folder name is SRES (IPCC - Special Report on Emissions Scenarios).

This directory contains folders with files that are permanent for the project, and the scenario folders.

4.2.3 bat

This directory contains scripts that are used to run IMAGE.

The most important ones are:

run_image.bat : to run the IMAGE executable
empty_image_output.bat : to delete old output files
empty_image_state.bat : to delete old restart files

For more information about these scripts, see Section 3.3.

4.2.4 bin

This directory contains the IMAGE executable *IMAGE_PC.exe*.

4.2.5 data

This directory contains the subdirectories *const*, *GCM* and *unf*, which contain files that are the same for all scenarios within the project.

4.2.6 const

This directory contains *const* control files.

These files contain physical – or ‘natural’ – constants. Generally, the values of these quantities will not be changed by the user. They may be altered by specialists, from new insights, or for experiments. A short description of each constant is given in the *const* files themselves.

For the relations between *const* files and submodels, see Chapter 1.

The path name keyword in *datapaths.dat* is *defDataConstDir*.

File name extensions are: *.const*.

4.2.7 GCM

This directory contains GCM (general circulation model) input files from 4 different climate models, identified by the first two letters:

| | | |
|----|---|-----------|
| HH | The UK Hadley Centre for Climate Prediction and Research | HadCM2 |
| EE | The German Climate Research Centre | ECHAM4 |
| CC | The Canadian Centre for Climate Modelling and Analysis | CGCM1 |
| AA | The Australian Commonwealth Scientific and Industrial Research Organisation | CSIRO-Mk2 |

The data that is used from these GCM's are:

- gridded monthly mean temperature change (GTMP_*.UNFo)
- gridded monthly mean change in precipitation (GPRES_*.UNFo)
- the global annual temperature change (*.DAT)

In addition, this directory contains several other files.

The files AAGGA150.DAT, EEGGA150.DAT, CCGGA150.DAT and HHGGA150.DAT, which are data obtained from GCM runs that determine the climate change due to an increase in greenhouse gases. AGS04T is data obtained from GCM runs that determine the climate change due to an increase in sulfate, keeping the remaining greenhouse gases constant. Finally, the following files are UIUC-GCM patterns constructed with increased forcing in the six UIUC regions:

| | |
|--------|---------------------|
| AGAMER | North America |
| AGAFRI | North Africa |
| AGEURO | Europe |
| AGSIBE | Siberia |
| AGASIA | Asia |
| AGSOUT | Southern Hemisphere |

Recently, additional GCM patterns have been added based on results from IPCC's Fourth Assessment report. The *readme* file in the GCM directory describes these additional files. At the same time, also the climate model of IMAGE has been updated with Magicc 6.0, which is described in the PBL report 'The climate subsystem in IMAGE updated to Magicc 6.0'.

The path name keyword of the GCM directory in *datapaths.dat* is *defDataGCMDir*.

File name extensions are: *.dat* and *.unfo*.

4.2.8 unf

This directory contains unformatted grid input files.

The path name keyword in *datapaths.dat* is *defDataUnfDir*.

File name extensions are: *.unfo*, *.unf1*, *.unf2*, *.unf4*.

4.2.9 <scenario_name>

This directory contains the run environment for one particular scenario. The folder name should identify the scenario involved. With the run environment, normally one recent scenario is provided.

4.2.10 CPCC

This directory contains output files with carbon plantation cost curves.

The path name keyword in *datapaths.dat* is *defCpccDir*.

File name extensions are: *.out*.

4.2.11 dat

This directory contains the *dat* control files.

These *dat* files define values for variables which control the simulation run.

In alphabetical order:

| | |
|--------------------|---|
| calibration.dat | This file contains a single keyword to denote the number of the calibration run, in case the IMAGE executable is used for calibration. For production use of IMAGE, the keyword <i>calibrun</i> is set to -1. |
| carbonregrowth.dat | This file contains the keyword <i>cregyear</i> which means number of years after which regrowth carbon fluxes are reconverted to natural fluxes. For the actual contents of this file see <i>carbonregrowth.dat</i> . |

The file is used in the submodel carbon.

| | |
|---------------------|--|
| const_filenames.dat | This file contains the names of the const files. In general, the user is advised <i>not to change</i> those names. When alternative constants should be used, it is advised to create alternative *.const files, and to use them adjusting the reference in const_filenames.dat (see Section 5.3). |
| cpoptions.dat | This file contains a number of parameters that influence the carbon plantation computations. For a detailed description of the parameters see <i>cpoptions.dat</i> . |

These options are used in submodel tes.

filehand. dat &

| | | | | | | | | | | | | | | | | | | | | |
|---|--|--|-------|----------|--|-----------------|--------|-----------|--|-------------|--|------------|--|------------|--|----------|--|---------|--|------------|
| filehandbf.dat | <p>The files <i>filehand.dat</i> and <i>filehandbf.dat</i> in this directory deserve special attention.</p> <p>The first of these two files is only used if <code>rfa4bfopt=0</code> (in the file <i>options.dat</i>), the other one is only used if <code>rfa4bfopt=1</code>. Both files contain a list of indicators for which output can be generated. For each indicator, flags can be set to indicate the years for which this indicator must be written to file. The years that can be selected have intervals of <code>RUN_TVM_INT</code>, which is an internal constant of IMAGE typically set to 5.</p> <p>If, for a particular indicator, this flag is 1 for any year, then IMAGE generates unformatted output files containing this indicator in the directory 'work' for the years that are marked by a 1. The names of these output files are constructed from the name of the indicator and the year for which the file has been written.</p> <p>For example, if the years for which output can be generated are 1970, 1975, 1980 and so on, and <i>filehand.dat</i> contains a line, similar to:</p> <pre>'GANNTMP '0 0 1 0 0 0 0 0 0 ...</pre> | <p>be produced by the message system. Levels equal to or higher than the threshold will pass, others are blocked. For details about the types and levels, see the file <i>message.dat</i>.</p> | | | | | | | | | | | | | | | | | | |
| | <p>options.dat</p> <p>This file contains a number of switches with which certain parts of the simulation process can be controlled. For a detailed description of the switches, see <i>options.dat</i>.</p> | | | | | | | | | | | | | | | | | | | |
| | <p>time.dat</p> <p>This file contains the variables which determine the time frame of the simulation:</p> <p><i>timestart</i> Starting year of the simulation; <i>by default: 1970. It is advised not to change this number</i>, as the code of the current IMAGE version is strongly related to this particular starting year.</p> <p><i>Note:</i> Preliminary calculations are performed from the year 1765 until 1970.</p> <p><i>timescen</i> Scenario year. That is the last year of the calibration (or historical) period, after which the scenario period starts. As some files are only provided for the historical period, the value of <i>timescen</i> and the starting and end dates of the respective files need to match.</p> | | | | | | | | | | | | | | | | | | | |
| | | <p><i>timestop</i> End year of the simulation. The length of the simulation period (<i>timestop-timestart</i>) must be a multiple of <i>outfreq</i>.</p> | | | | | | | | | | | | | | | | | | |
| | | <p><i>outfreq</i> Output interval, in years. For every <i>outfreq</i> year, IMAGE writes variables and special results to output files.</p> <p><i>Note:</i> <i>outfreq</i> must be a multiple of 5.</p> | | | | | | | | | | | | | | | | | | |
| then a file named GANNTMP_1980.UNFo is generated in the directory <i>work</i> . | | | | | | | | | | | | | | | | | | | | |
| GCMoption.dat | <p>This file contains information on which GCM pattern should be used in <i>climscale.f</i>. An overview of available patterns and abbreviations is provided at the end of the file.</p> | | | | | | | | | | | | | | | | | | | |
| imexport.dat | <p>This file contains all setting variables necessary to control the restart mechanism. For a description of the variables, see <i>imexport.dat</i>. See also Chapter 8.</p> | | | | | | | | | | | | | | | | | | | |
| message.dat | <p>This file contains two options to control the type and levels of messages that the program produces: <i>typemask</i> and <i>threshold</i>.</p> <p>The message system recognises several types of messages. With the keyword <i>typemask</i>, the user may specify the types of messages that will be produced by the system. Furthermore, the system assigns different levels to each type of message. With the keyword <i>threshold</i>, the user may define the levels that will</p> | <p>4.2.12 input</p> <p>This directory contains the scenario input files, and all other input files.</p> <p>Several files are produced by other models (see also Section 7), or by pre-processing, or in a combination of both:</p> <table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">FAIR:</td> <td>AB* .OUT</td> </tr> <tr> <td></td> <td>Sinks2image.scn</td> </tr> <tr> <td>TIMER:</td> <td>BFDEM.SCN</td> </tr> <tr> <td></td> <td>BIOTRAD.SCN</td> </tr> <tr> <td></td> <td>GDP_PC.OUT</td> </tr> <tr> <td></td> <td>IVA_PC.OUT</td> </tr> <tr> <td></td> <td>MFBF.SCN</td> </tr> <tr> <td></td> <td>POP.OUT</td> </tr> <tr> <td></td> <td>EMIS* .OUT</td> </tr> </table> | FAIR: | AB* .OUT | | Sinks2image.scn | TIMER: | BFDEM.SCN | | BIOTRAD.SCN | | GDP_PC.OUT | | IVA_PC.OUT | | MFBF.SCN | | POP.OUT | | EMIS* .OUT |
| FAIR: | AB* .OUT | | | | | | | | | | | | | | | | | | | |
| | Sinks2image.scn | | | | | | | | | | | | | | | | | | | |
| TIMER: | BFDEM.SCN | | | | | | | | | | | | | | | | | | | |
| | BIOTRAD.SCN | | | | | | | | | | | | | | | | | | | |
| | GDP_PC.OUT | | | | | | | | | | | | | | | | | | | |
| | IVA_PC.OUT | | | | | | | | | | | | | | | | | | | |
| | MFBF.SCN | | | | | | | | | | | | | | | | | | | |
| | POP.OUT | | | | | | | | | | | | | | | | | | | |
| | EMIS* .OUT | | | | | | | | | | | | | | | | | | | |

ENEM*.OUT
INEM*.OUT

pre-processing: ANIMALSTOCKCH.SCN
CROPSTOCKCH.SCN

Some of the files are only used for a particular setting of *modelopt* (see *options.dat* under the *dat*-directory):

For example, *modelopt* determines, whether the following files are used:

modelopt=0: ELAARGDP.DAT
FOODUTIL.DAT
FRINTB11.SCN
FRINTB12.SCN

modelopt=1: AGRCONSA.SCN
AGRPRODA.SCN
AGRPRODC.SCN

The path name keyword of this directory (i.e. *input*) in *datapaths.dat* is *defInputDir*.

File name extensions are: *.scn*, *.out* and *.dat*.

All files that provide scenario information for the array of regions, and over the entire scenario period (timestart – timescen – timestop) have the extension *.scn* (as they describe the scenario). However, for historical reasons, some files that are produced by TIMER or FAIR still have the extension *.out* (as they are output of these models).

Other files that give regional information, but not for a time series, have the extension *.dat*. And also a few other files that were not placed in *data/const* for practical reasons (*landalb.dat*, *soil.dat*, *soilwhc.dat*) are included in this folder and also have the extension *.dat*.

4.2.13 map

The control file *filehand.dat* determines which files are written out, later to be converted to M files and placed in this directory. The map output is first written to the directory 'work' and can later be converted to M format in the directorylder 'map' (see also Section 6.2). During the calibration runs, however, some maps are directly written to the directory 'map'.

The path name keyword in *datapaths.dat* is *defMapDir*.

File name extensions: none.

4.2.14 output

This directory contains regional scenario output files.

The path name keyword in *datapaths.dat* is *defOutputDir*.

File name extensions are: *.out*.

4.2.15 outputcell

This directory contains files grid cell specific output that is produced by the routines *followgrid.f* and *followforest.f*.

The path name keyword in *datapaths.dat* is *defOutcellDir*.

4.2.16 outputrest

This directory contains files with logs from the adm and lcm models.

The path name keyword in *datapaths.dat* is *defOutputrestDir*.

There are actually only two files in this directory: *adm.out* and *lcm.out*.

4.2.17 state

This is the main directory for the structure of subfolders containing the *restart* files.

There are two types of *restart* files:

- Regular *restart* files.
- Unformatted grid *restart* files.

For more information about restart functionality and *restart* files, see Chapter 8, Restart functionality, and Appendix A, File structures.

4.2.18 work

The IMAGE executable runs in this directory. Therefore, this directory also contains the file *datapaths.dat*, where keywords determine the directory paths of all necessary control, input and output files. Note that all paths are relative to the directory 'work', so that this file does not have to be adapted when copying the project run environment.

After an IMAGE run, this directory contains unformatted intermediate grid files. As the IMAGE executable runs in the directory *work*, pathless output files are automatically created in this directory. The intermediate files exist only during the simulation run. They are deleted at the end of the simulation, unless the run is prematurely ended, for instance, when the program stops after a fatal error occurs.

There is no path name keyword for this directory in *datapaths.dat* as this is the starting point of all the relative paths in *datapaths.dat*!

File name extensions are: *.unfo*, *.unfi*, *.unfz*.

4.2.19 start

This directory contains terrestrial vegetation input files for the *start* year. Most of these files are created by the IMAGE calibration run4.

The path name keyword in *datapaths.dat* is *defStartDir*.

File name extensions are: *.unfo* and *.unfi*.

Working with IMAGE

5

The IMAGE program is under constant development. New models are added, existing models are modified on the basis of new insights, and new or improved data is used as it becomes available. All this makes working with IMAGE a highly dynamic activity. This chapter gives a first outline of the way in which IMAGE is used, structured around common activities.

5.1 Creating a new project

A new project typically arises when a series of related scenarios have to be developed and evaluated for a specific purpose. In most cases, such a project is related to a specific project, either within the PBL itself, or in cooperation (Eurualis, EU, IPCC, etc.). All projects and scenarios are centrally stored at `\data\image\Scenario_lib`.

To start working on a whole new series of scenarios, the new 'project', the user has to set up a new run environment. This can be done by copying the example project directory with its complete contents, and subsequently adapting the files in this new environment. For a description of the run environment, see Chapter 4.

At project level, the user may want to change or replace the scenario-independent *const* and/or other files in the *data* or *start* directories outside the scenario directory. These changes will remain valid for all scenarios within the project.

Note: The file *datapaths.dat* need not be changed: all paths are relative.

The user may wish to create alternative versions of the IMAGE executable within the project environment. Typically, the same executable is used throughout an entire project.

5.2 Creating a new scenario

A new scenario has to be created for every new IMAGE run or experiment, if no old run should be overwritten. For this purpose, a new scenario directory will be created in an existing project directory or in a new project directory. To create a new scenario within the project environment, the user can make a copy of the existing scenario directory with all its subdirectories.

Additionally, the user may want to adapt or replace the *dat* and/or *input* files in the directories within the scenario directory. The appropriate scenario files have to be set according to the specifications of the scenario. This is typically a task of one of the model experts who knows which scenario files have to be modified and how. The origin of the scenario data in the input folder have to be documented in the excel file 'INPUT' or in a README.txt file in the *docu* folder.

To clean up the result directories before the simulation, the user can run the batfiles for clearing the output and restart folders. For a description of these batfiles, see Section 4.2.2. Finally, the user can start the scenario simulation by calling the runsript *run_image.bat*.

In principle, all input data can be manually changed by using a text editor. However, an easy way of manipulating input data, especially in the scenario period after *timescen*, is by importing the input file into excel, manipulating the data there, and exporting them again into the textfile structure. Both import and export are done by a special Excel Macro 'image.xls', which is described in more detail in Section 6.1

5.3 Updating input data

Much of the input data for IMAGE is obtained from other sources and models, either within or outside PBL. When new versions of these data become available, then it may be important to use this new version for further computations with IMAGE. This section describes the steps to be taken to insert the new data in the IMAGE environment.

First, the data that is supplied must be reformatted to the IMAGE input format (see Appendix A for concise information on the IMAGE file formats or the programmers manual, for details). In principle, all new input data for the period before *timescen* make a new calibration of IMAGE necessary (see Appendix B).

The new data must then be inserted in the IMAGE production environment. In principle, data in completed projects is not updated, but all new projects will make use of the updated data. This means that the data must be copied to the proper place in the first project and scenario that is going to use the new data.

Obviously, IMAGE needs to be calibrated with the new data (see Appendix B). Once this has been done, the information for the historical period (*timestart* – *timescen*) is updated, and IMAGE is ready for use with the new data. Make sure to check-in the new version of the data in the version control system with appropriate description of the modifications.

For the period after *timescen*, most data updates do not provide new data, but often trends from old scenarios are also applied to the new starting point (see also Appendix B).

5.4 Using model options

As already mentioned in Section 4.2.11, there are several ways to use alternative settings for running a scenario. Here, we describe the model options that are most relevant to using IMAGE.

Options.dat: In this file, many model options can be set. Above each option, a short explanation is given.

Cpoptions.dat: In this file, setting for the carbon plantation module can be set. Above each option, a short explanation is given.

GCMoption.dat: In this file, the GCM pattern to be used in *climscale.f* can be selected. An overview of available patterns and abbreviations is provided at the end of the file. For standard scenarios, AOS settings (see below, *aos_*.const*) and GCM options should be consistent and use information from the same GCM. For uncertainty analysis, however, GCM pattern and AOS settings from different GCM may be combined.

Const_filename.dat: In principle, the **.const* file should not be changed. However, in some cases alternative **const* files can be used to do sensitivity analysis by adjusting the setting, for example, to test alternative carbon cycle parameterisation. Also, the use of different AOGCM calibrations (as available with the Magicc 6.0 update) is done by choosing the respective *aos-const* file in *const_filenames.dat*. For every AOGCM that has been calibrated, a specific *aos_*.const* file is available.

5.5 The zoom version

In order to better couple IMAGE to the LEITAP model, a zoom version of the IMAGE model has been developed. That means, most importantly, that the number of regions has been increased from 24 to 45, with, additionally, most European countries being represented in a separate region. All details of the zoom version, and especially the preparations necessary to prepare import data, are described in the document *IMAGE_Zoom.doc* (see Section 1.2).

In order to use the IMAGE code to run in the zoom mode, the files *image.fip* and *soxinhis.const* need to be changed to the 45 regions version, and the code to be compiled again. Additionally, all input files and the DATA and START information need to be used from the zoom calibration. In

order to make use of the ZOOM version's specific DATA and START directories, the file *datapaths.dat* in the directory 'work' need to be adjusted accordingly.

5.6 Adding new or modified models

New or modified models are constantly added to the IMAGE program, as it tries to stay in line with the most recent insights. The development of these modifications is typically done in a separate environment, up to the point where it has been sufficiently tested to be included in projects.

At that point, a quality check should be performed to determine that no serious errors have been made, and to see whether the new program code conforms to the programming guidelines.

Preferably, the check on the programming guidelines should be done first. If modifications have to be done to bring the code in line with the guidelines, then these modifications are at least checked by the person who inspects the code from a modelling point of view. The check to see if no serious modelling errors are included is best done by a fellow developer. If the quality check detects problems at any point, then the developer should correct the problems and the check should be performed again.

After the quality check, the new program code must be checked in in the version control system for the software, so that it gets a unique version identifier. Decisions have to be made as to which projects/scenarios will be run with the new or modified model. This must be properly administrated, so that the version with which results have been obtained can be unambiguously determined afterwards.

In many cases, new or modified data will have to be included (see Section 5.1 above) and the IMAGE model will have to be recalibrated (see Appendix B).

6

Pre-processing and post-processing facilities

In this section, we describe a number of utilities that are available around the IMAGE.exe.

These utilities are not part of the IMAGE program, but available in separate locations on the central drive, mostly at `\project\M481508_IMAGEaanpassingEnBeheer\Tools\`, and at `\data\image\Utils\`.

6.1 The IMAGE macro

There is a set of macros in Excel to support importing and exporting IMAGE data in M-format. A database with dimension data, identifier code and items, is maintained to provide the imported data with labels. The Excel workbook `image.xls` contains the macros as well as the database. Map data cannot be imported in Excel due to the size of grid data. This data can be converted to ascii-grid format with the `m2gis` tool (see Format conversions).

6.1.1 Installation of the macros

`Image.xls` can be found at `\project\M481508_IMAGEaanpassingEnBeheer\Tools\xl_macro`.

1. Copy `image.xls` to your local drive, for instance `c:\apps\image`.
2. Open this file once and accept macro enabling (when prompted).
3. The menu bar will show a new toolbar called IMAGE with many buttons.
4. Close the file

From now on the IMAGE buttons will be visible in Excel and can be used.

6.1.2 The IMAGE macro manual

There is an extensive manual on the macros in Dutch ('Excel macro handleiding.doc') and a short version in English ('short English version.doc'), see Section 1.2. Only the import and export macros are described here.

6.1.3 Use of Excel macros: import data

On the menu bar you'll find the IMAGE toolbar with the buttons:



- `IMAGE/M=>Long`: import data using long format.
- `IMAGE/M=>Matrix`: import data using matrix format.

Clicking a button will open a file dialogue. Browse to your data file and open it. The macro reads the data, uses the IMAGE header (first line in data file) to identify the dimensions, interprets these and renders the labeled data to a new, blank sheet.

IMAGE header: `! type variable [dimension1,...,dimensionn] (t); unit=F unit name; label=F descr. var.`

Long format means that, after import, every row will contain one data item and its identifying dimension classes. This format is suitable for the Excel Pivot table Report function.

Matrix format: After opening the data file, the data is presented in a matrix format on a new sheet. The last dimension is used to create the columns of the matrix.

If there is no IMAGE header or the dimension code from the header is not in the database, the dimension items will get a generic label. In case of the latter, please contact Rineke Oostenrijk, or `image-info@pbl.nl` for an update of the database.

6.1.4 Use of Excel macros: export data

Export buttons on the menu bar:

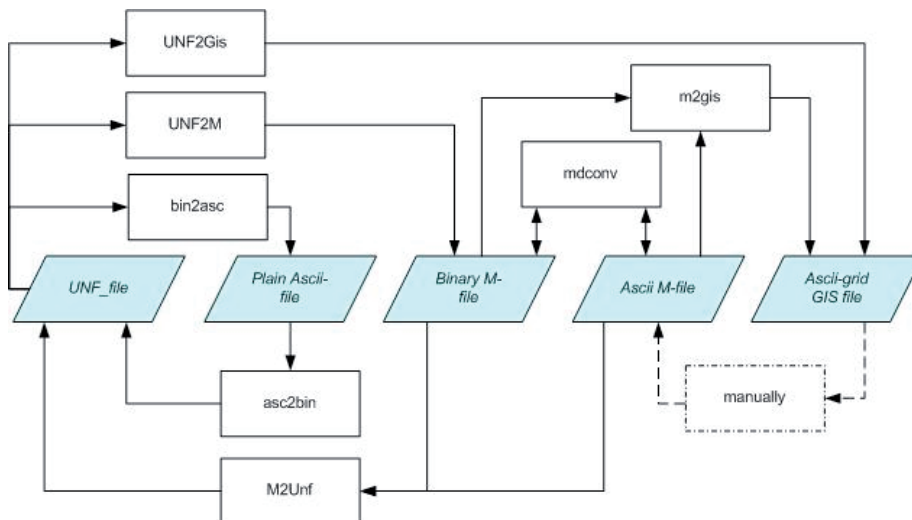


- `Excel=>M/IMAGE`
- `Excel=>M/IMAGE UNIX`

You can adjust data in a worksheet and then export the sheet to M format again. The UNIX button is made to generate data in the M format that can be read as a data file by IMAGE. This is a stricter format based on agreement. So be sure to use the second button if you create an input file for IMAGE!

6.2 Format conversions

Image uses data in UNF format for grid data and ascii M format for the regional data. Conversions are needed to exchange data with other models and/or programs. The IMAGE viewers use data in M format. Grid data (UNF files)



must be converted to M format for the viewers or to ascii-grid format that can be used in GIS software. UNF-format data can be converted to plain ascii files to be viewed in a text editor. Some conversion programs depend on the IMAGE naming convention for the UNF files: <var>[_<year>][.<dimension size>].UNF<datatype=F 0|1|2|4>. 0,1,2,4 represent the data types float, byte (0 or 1), short, and int, respectively.

A scheme, (Figure 6.1) visualising from-to ways to convert the above mentioned formats.

6.2.1 Format descriptions

UNF format : Unformatted binary data. No header, no footer, just a sequence of numbers. UNF files have a number extension indicating the type of data : UNFo =F real (4) , UNF1 =F byte, UNF2 =F short, and UNF4 =F Integer(4)

Two types of the M format can be distinguished: binary and ascii.

Binary M format: contains a header with information on the data structure, year, followed by data of year, ..., and is closed with a footer. See the M documentation for a description of the binary format.

Ascii M format: includes an (optional) IMAGE header. The format is composed of the following components:

1. the IMAGE header followed by zero to an arbitrary number of comment lines.
! type variable [dimension₁,...,dimension_n] (t); unit=unit name; label=F descr. var.
2. the M header
type variable [dimension₁,...,dimension_n] (t) = [(time dependent)
or
type variable [dimension₁,...,dimension_n] =
3. Data part: If time dependent, then year, followed data for respective year. If not time dependent, just data. Numbers are separated by spaces, commas, new lines, or tabs
- 4.

5. Footer. If time dependent, then]; else ;

Ascii-grid GISformat is a standard ascii format for GIS raster files. It is composed of:

1. a header containing something similar to:

| | |
|--------------|----------|
| ncols | 720 |
| nrows | 360 |
| xllcorner | -180 |
| yllcorner | -90 |
| cellsize | 0.500000 |
| NODATA_value | -9999 |
2. number of 'latitude' lines with number of 'longitude' numbers. No data values are usually represented by -9999.

Plain Asciiformat is a sequence of numbers in ascii. No header, no footer.

6.2.2 Conversion programs

All conversion programs can be found in the IMAGE project directory (see Section 1.2), under 'Tools\FormatConversions'.

unf2M

Location: subdirectory unf2M

Usage: dir_unf2m <scenario directory> [-<outdir>] [-f <filehand.dat>] [-p <variablename>]

Defaults: output directory =<scenario directory>\MAP and the 'filehand.dat' in the UNF2M directory is used.

Options: The -f option defines a specific 'filehand.dat'. Use the absolute path name to specify 'filehand.dat'. The -p <variablename> option converts all the <variablename>... <year>.UNF files of this variable in the <scenario directory>

UnfToGis

Location: subdirectory unf2Gis

Usage: The application appUnf2gis.exe is the graphical interface to unf2Gis.exe. This program converts the binary unformatted .UNF x ($x = F 0, 1, 2, 4$) output files from the IMAGE model into ascii-grid files, which can be read by GIS software.

Defaults: Cell size =F 0.5, ncols =F 720, nrows =F 360 numbers, XLL corner =F -180, YLL corner =F -90, and NODATA is represented by -9999.

Remark 1: There is an unf2gis.ini file with defaults for the input and output directory. Defaults can be modified if desired.

Remark 2: You can use the conversion program unf2gis.exe directly from the dos prompt or in a batch file with the following command:

```
unf2gis -f <filename>[-d <input directory>] [-o <output-
directory>] [-g <grid data directory>] [-s Cell size] [-n
nodata] [-c xllcorner yllcorner] [-r nrows ncols]
```

bin2asc

Location: subdirectory bin_asc

Usage: bin2asc 0|1|2|4 [-u] [-a coeff] [-b coeff] [-d decim] [-o outputpath] <inputfile> ... Bin2asc reads a sequence of binary files (of reals, bytes, shorts or ints) and writes to <inputfile>.asc, their ascii representation separated by newlines.

Defaults: signed data, transformation =F $1.0 * x + 0.0$, print no decimals, output path is same as for <inputfile>

Options: -u for unsigned data type, -a and -b take floats that specify the transformation $a * x + b$, -d is the number of decimals to print, -o is the output path

asc2bin

Location: subdirectory bin_asc

Usage: asc2bin 0|1|2|4 <inputfile> <outputfile>. Asc2bin reads an input file containing 'whitespace' separated values and converts these values to a sequence of binary values of data type 0|1|2|4 =F float, byte, short, and int

M2Unf

Location: subdirectory m2unf

Usage: m2Unf <type (0|1|2|4)> <m-file> <outputdir>. M2Unf reads a file in M format (binary or ascii) and converts it to one or more UNF files of type 0|1|2|4 =F float, byte, short, or int, in the output directory

M2Gis

Location: subdirectory m2gis

Usage: Start m2Gis.exe. This is a graphical interface to the conversion function. Choose between the options 'one file' or 'whole directory'. Browse to the <scenario> map folder and select the file (option one file) or select

the map folder (option whole directory). Browse to an output directory on your system and activate convert.

Help: an extensive help is available in m2gis.html

Mdconv

Location: subdirectory mdconv

Usage: mdconv.exe [options] <inputfile> <outputfile>. Mdconv is an M tool and converts binary M data to ascii M data and vice versa.

Defaults: conversion to ascii format

Options: -h displays help info, -a generates ASCII output, -b generates binary output, -t generates tabular ASCII output, -r converts to floats, -i converts to integers, -byte converts to bytes.

Ascii-grid GIS to Ascii M.

This conversion needs to be done manually, by removing header and stop sign, and by removing all no-data values (mostly -999).

6.2.3 Automatic conversion after IMAGE run

In order to view the grid data of IMAGE runs in USS, post-processing of the UNF data from IMAGE to M binary format is necessary (see also above, format conversion). A batch file has been made to automate the post-processing and is provided at \data\Image\Utils\modellentrein\IMAGE_postprocessing.

6.3 The viewer USS (User Support System)

Here, we provide a very short description of the USS; more information can be found at \data\Image\documentation\USS_howto.

6.3.1 Preparatory steps

Unf2M: In order to view the grid data of IMAGE runs in USS, post-processing of the UNF data from IMAGE to M binary format is necessary. A batch has been made to automate the post-processing and is provided at \data\Image\Utils\modellentrein\IMAGE_postprocessing (see also above, format conversion).

MSD entry: To view a scenario in USS there must be an entry in the <theme>.msd for the scenario. You can manually enter the scenario link and meta data or use the tool \data\Image\Utils\USSMSD\msd.exe to add scenarios and/or edit the msd file.

Creating new theme: When starting a project, you may need to create a new theme for its scenarios.

1. edit the Theme.ini file in the uss directory:
Add a line with <themecode>, <theme label> , [< themedocumentation>.doc].
2. Open a DOS command window with 'run cmd' and go to directory \data\Image\utils\uss24. Enter: CopyUss24.bat <themeCode>. This will copy all the necessary files (.mdl, .sce, .mcm, .mop) to the USS directory.

3. Create `uss24_<themecode>.msd` with `\data\image\utils\USSMSD\MSD.exe`

6.3.2 Short description of User Support System (USS)

USS is a User Support System to view and analyse the output of TIMER and IMAGE. USS is created with the MyM software. This software provides USS with all kinds of graph types and graph functions. IMAGE results are stored in `\data\image\Scenario_lib\<project name>\<scenario name>`. In USS you can load a scenario. Some 200 variables or indicators belonging to this scenario are displayed in a hierarchical order. Variables can be grid data and then displayed as maps which may be animated in time. Non-grid variables can be defined over more dimensions, that is, over regions, sectors, and energy carriers. In USS you can view any cross section of these variables with the MyM functionality. It is also possible to load more scenarios and compare the data of this scenario within the graph of a specific variable.

6.3.3 Theme structure for scenarios

In the course of time, many scenarios have been developed by the IMAGE team. These scenarios are organised in themes to keep USS manageable. A theme consists of all the scenarios summed in the `<theme name>.msd` and there is an identical USS for each theme. A theme selector tool, the so-called scenario viewer (`theme.exe`), facilitates the choice and launch of the USS for a specific theme. Each version of IMAGE, for 17, 24, and 45 regions, has its own scenario viewer.

6.3.4 How to start and use

Shortcuts to the scenario viewer can be found in the directory `\data\image\`.

- `Uss24 (\data\image\USS24\theme.exe)` starts USS for the current IMAGE 2.4 24-region version.
- `themeUSS45 (\data\image\USS45\theme.exe)` starts USS for the 45 Region Zoom version
- `IMAGE2.2ScenarioViewer (\data\image\USS\theme.exe)` starts USS for the IMAGE 2.2 17-region version.
- `ImageDocumentation (\data\image\documentation\IMAGE2.2\index.html)` launches the html based documentation for IMAGE 2.2. Here, you can find documentation for each indicator, for all the IMAGE submodels, scenario concepts and so on. Though this documentation has been made for the USS for IMAGE 2.2 version, most of it is still valid for the current version.

Coupling to other models

7

Several of the IMAGE input files can be provided by different models linked to the IMAGE model described here.

7.1 IMAGE-TIMER-FAIR

The communication between the TIMER, IMAGE and FAIR models consists of three different parts.

- Providing information on biofuel potentials to the TIMER model (7.1.1)
- Providing baseline information from IMAGE to the FAIR model (7.1.2)
- Converting and using FAIR and TIMER data as input for an IMAGE run (7.1.3)
- Providing Cost curves for carbon plantations to the FAIR model (7.2.4)

The entire communication will soon be handled via the so-called ‘Model train’, which will be described in a separate document. It will provide an easy-to-use tool to execute the

batch files and manual actions described in the following. The figure below shows the main data flow between the IMAGE, TIMER and FAIR models. In this document, the flows to and from IMAGE are described, not those between the FAIR and TIMER models.

7.1.1 IMAGE to TIMER

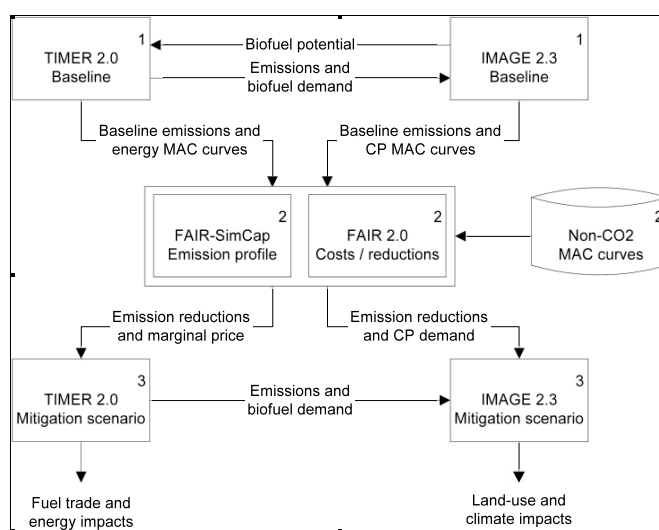
The TIMER model uses information from the IMAGE model on the availability of land to produce biofuels. If the biofuel option `rfa4bfopt` is set to 1 (=F Demand for biofuels zero; RFA becomes available for BF), then the maps indicating the potential for the crops sugar, maize and woody biofuels (`GRMPPC_MAIZE`, `GRMPPC`, `GRMPPC_WB`), plus a summary map (`GLCTRFA4BF`) are written out. These maps need to be converted to M format (see Section 6.2) and can then be used by the TIMER model.

7.1.2 IMAGE to FAIR

The FAIR model uses cost curves for carbon plantations (see below) and activity data from the IMAGE model. All land-use

Linkage and information flows between the FAIR, TIMER and IMAGE models

Figure 7.1



Linkage and information flows between the FAIR, TIMER and IMAGE models (note CP = Carbon plantations). For more detail see mitigation papers.

related GHG emissions in the baseline scenario, which can partly be mitigated, are communicated to FAIR. Resulting from a climate policy simulation in FAIR, IMAGE then receives abatement levels for all gases (see below).

7.1.3 FAIR/TIMER to IMAGE

This procedure needs to be followed to provide input from the FAIR and TIMER models to the IMAGE model (it overlaps partly with the general instructions for running IMAGE)

- 1) Update IMAGE data in the input folder, for example, based on a similar old scenario.
- 2) Put (new) TIMER data in the folders <scenarioName>\econ, emis, energy and pop
- 3) Put all files from the FAIR model to <scenarioName>\mitig, and also copy the CPCC folder to the scenario library.
- 4) Copy sinks2image from the FAIR model to the input directory and delete all years before 2005.
- 5) Put a command prompt at \data\Image\Utils\modellentrein\IMAGE_TIMER_FAIR and run 'image_timer_fair_24.exe <projectName>\<scenarioName>'
- 6) Make sure the implementation fraction (CPOPTIONS.DAT) and all options (OPTIONS.DAT) are in order.
- 8) Run IMAGE

7.1.4 Cost curves for carbon plantations

This procedure describes the generation Cost Curves for Carbon Plantations, as used in the FAIR model.

Step 1

Run a baseline scenario where the 'Indicator for taking carbon plantations into account' in the OPTIONS.DAT file should have the value 1 (=Potential carbon plantations).

Step 2

The baseline scenario run produces, among other things, the following output files:

- GCPSPPAVG.UNF0 contains information on the annual average net potential carbon sequestration of possible carbon plantations, compared to the baseline. Note that, in the potential run, all grid cells where trees can grow are considered as possible locations for carbon plantations. In the end, only those grid cells that have a net positive carbon sequestration are taken into account.
- GLCT_CPORG.UNF1 contains the original land-cover types of all potential carbon plantations.
- GCPYRS.3.UNF2 contains (i) the earliest possible planting year of a carbon plantation, (ii) the first year where the cumulative carbon sequestration amount (compared to the baseline) is positive and (iii) the end year after which no further sequestration takes place (only applicable to permanent plantations).
- GCPTREE.UNF1 contains the tree species which are used in the carbon plantations.

These files are needed as an input to the program SPPCURVE.EXE. This program generates text files containing the supply curves for all 26 regions with a timestep of 5 years, starting in the year 2000. The names of the output files are: SPP<region>_<year>.OUT. Output files for a certain year are generated only if there is a potential in the year considered.

Each output file contains the following nine columns:

1. Grid number
2. Average net carbon sequestration per year, compared to the baseline (tC/ha/yr)
3. Surface area (Mha)
4. Original land-cover type
5. Tree species (1... 6)
6. Starting year of the carbon plantation (2000... 2100)
7. Starting year of sequestration (2000... 2100)
8. Total net carbon sequestration per year, for this grid (MtC/yr)
9. End year (2000... 2100 or -9000, where the latter means: somewhere after 2100)

If the program SPPCURVE.EXE is run, it asks for a scenario name.

Then, it expects the following input directory containing the input files mentioned above:

```
D:\sinks\<scenario_name>
```

Currently, the first part of the directory name is hard coded, but should be read from an input file. The output directory currently is:

```
D:\sinks\curves\<scenario_name>
```

Again, the part D:\sinks\curves is hard coded.

Step 3

The supply curves as generated in the previous step are input to an excel macro called sinks26_macro.xls. This macro contains a sheet containing the 'Costs Parameters', which should be self-explanatory. More information on the values can be found in Strengers *et al.* 2008, 'The role of carbon plantations in mitigating climate change: Potentials and costs'.

If the macro is activated, the following window is shown in Figure 7.2:

Input directory: Enter the name of the directory in which the supply curves, generated in the previous step, are to be found.

Output directory: Enter the name of the directory in which the cost curves are to be placed by the macro, plus the name of the output spreadsheet that is to contain the cost curves. Besides this spreadsheet, spreadsheets for each time step will be generated. Also, a ASCII file called *costcurve.dat* is generated. *This file must be used as input for the FAIR model.*

Stepsize for taxclasses: Stepsize used in the cost curves, in this case 5 dollars.

Vegetation types: In principle, carbon plantations will be planted on abandoned agricultural land only. However, the supply curves also include information on other vegetation types in the cost curves. This can be selected in this field. REMARK: currently, other vegetation types are not included in the supply curves (to minimise their size). If necessary, this can be done easily. Please, contact Bart Strengers or image-info@pbl.nl for more details.

Select GDP/capita: Enter the name of the file containing the GDP scenarios for all regions.

Select Population: Enter the name of the file containing the Population scenarios for all regions.

Finally, click on the 'Go' button!

If any problems occur when running this macro, Rineke Oostenrijk will be able to solve it!

7.2 Providing data to GLOBIO

IMAGE data are used as an input to the global biodiversity model Globio. The coupling and the datasets are described in a separate document (IMAGE_to_GLOBIO.doc, see Section 1.2).

7.3 Coupling with agro-economic models

The regional demand, trade and production of agricultural commodities can either be calculated by the Agricultural

Economy Model (AEM) of IMAGE, or by other, more specialized agro-economic models. The recent year, most applications of IMAGE have been carried out in cooperation with the agro-economic models LEITAP (derived from GTAP) or IMPACT. From these models, IMAGE receives regional production of crops and livestock products, and the change in agricultural intensity. The coupling to both models is described in more detail in other documents (see Section 1.2).

7.4 Coupling to LPJ

In order to update the dynamic vegetation and carbon cycle modules in the IMAGE model, to add hydrological module, and to improve the crop modelling, a cooperation with the LPJ model has been established, and the LPJ model is currently being coupled to the IMAGE model. This coupling is described in detail in the document IMAGE_LPJ*.doc (see Section 1.2).



Restart functionality

8.1 Functions

The restart functionality comprises three main functions:

1. *Restart run*: the possibility to run a simulation that starts off with data resulting from a previous run.
2. *Nocompute*: the possibility to switch off the functionality of certain selected submodels in a simulation run.
3. *Readfromfile*: the possibility for certain selected submodels, that have the *nocompute* option set, to read their results from files produced by a previous run.

To perform a *restart run* or a run with the *readfromfile* option, restart files with data from a previous run have to be available. The action of producing these restart files in a (previous) simulation run is called 'exporting'; a run can 'export' files; the created restart files are sometimes called 'export files'. The opposite action: reading the restart files, is called 'importing'; a run can 'import' files.

Restart files are located in one or more subdirectories under the main restart directory *state*, see also Section 4.2.17.

The user can control the functions of the restart functionality by setting the values of the keywords in the control file *imexport.dat*, which can be found in the folder *dat* under the scenario directory. Keywords associated with a function will be discussed in the section about that function, below.

General keywords are:

| | |
|---------------|--|
| exporttextBin | The value of this keyword is a string that denotes the name extension of binary restart files; see also the keyword <i>textbinary</i> . Standard value: .bin |
| exporttextTxt | The value of this keyword is a string that denotes the name extension of textual (ascii) restart files; see also the keyword <i>textbinary</i> . Standard value: .txt |
| textbinary | The value of this keyword determines whether the restart files are in binary or in textual (ascii) format. Binary format is efficient, textual format is useful for inspection with a text editor. |

Possible values:

1 =textual format

2 =binary format

Standard value: 2, binary format

appendoverwrite The value of this keyword determines whether new restart data have to be appended to old data in existing restart files, or that the old data have to be overwritten by new data.

Possible values:

1 = append the new restart data.

2 = overwrite the old restart data.

Standard value: 2, overwrite.

image_syspath The value of this keyword is a string denoting the main folder, located under directory *state*, for the directory tree of restart files. This directory tree has a structure that matches the submodel structure, as depicted in Chapter 1.
Standard value: /.

<subsys>_syspath The value of this keyword is a string denoting the path, under <image_syspath>, to the submodel folder for restart files.

restart_dir The value of this keyword is a string denoting the folder, located under directory *state*, in which restart files are stored when the value of keyword *exportgoal* equals 'export for restart'. See the description of *exportgoal* in Section 8.2.
Standard value: restart/.

8.2 Preparations

Preparations have to be made for restart runs and for readfromfile runs. These preparations are virtually identical for both types, but there are specific differences.

To make a run produce restart files, the following keywords of *imexport.dat* have to be considered.

nexpyears The value of this keyword denotes the number of simulation times (years) at which restart files have to be exported, see keyword *expyears*.
Possible values:
0 means: no files are exported.
-1 means: files are exported for every simulation time.
>0 means: number of times (years) given in keyword *expyears*.
Standard value: -1.

Possible values:
-1 =no restart
<yr> = the year from which the simulation has to run
Note: <yr> must lie within the simulation period.
Standard value: -1, no restart.

Also, nexpyears will often be set to zero, indicating that no restart files have to be generated (as this run already starts from restart files).

expyears This keyword contains the simulation times (years) at which restart files are exported. The actual value is a series of numbers separated by blanks.
Note: The number of specified times (years) must equal the value of keyword *nexpyears* when this is greater than zero. When *nexpyears* is 0 or -1, the values in *expyears* are ignored.
Note: Restart files are only created for times actually occurring in the simulation period.
Standard values: 1970 1971 1972 1973 1974 1975

Step 2: Then copy the necessary data from the backup to the *state* directory. For example, to restart in 2007:

```
<user_directory>\project_name>\bat>prepare_restart.bat
<scenario_name>
```

This copies the information from the export year and the years 1970, 1990 and 200 (if appropriate) to the folder *state*\restart.

Step 3: If the restart year is not '1 plus a multiple of five' then the last special year before the restart year must also be copied:

```
<user_directory>\project_name>\bat>copy_stateyyyy.bat
<scenario_name>
```

exportgoal The value of this keyword determines the purpose of the restart files.
Possible values:
1 = export for restart.
2 = export for readfromfile.
Standard value: 1, export for restart.

This also applies if the restart year is 2008, 2009 or 2010.

Step 4: Now the restart run can be started:

```
<user_directory>\project_name>\bat>run2.bat
<scenario_name>
```

When the run is finished, the results in the *state* and *output* directories have to be copied to the directory *state_backup*. This can be done with scripts available in the *bat* directory:

```
<user_directory>\project_name>\bat> backup_state.bat
<scenario_name>

<user_directory>\project_name>\bat> backup_output.bat
<scenario_name>
```

With these backup data, a run can be restarted as many times as needed by following the procedure described in Section 7.3.

8.3 Restart run

A restart run starts the simulation with data generated by a previous run. The settings for that run are described in Section 8.2, Preparations. A restart run requires four steps to be taken:

Step 1: Before starting the restart run, the keyword *restartYear* in the control file *imexport.dat* must be set:

restartYear The value of this keyword determines the year from which the simulations starts.

Possible values:
0 = include submodel computations

Attention: In the preparations, restart files must have been generated for time =F <yr>-1!

Note: In the preparations, the keyword *exportgoal* must have been set to 'export for restart'.

Note: The restart run itself may produce export files, but then the keyword *exportgoal* must keep the value 'export for restart'.

Note: In the present version, no submodel may have the *readfromfile* option set in a restart run.

8.4 Nocompute

The user may run a simulation in which one or more submodels are switched off. This option is particularly useful in case (other) submodels have to be tested independently. To switch off certain submodels, the keywords <subsys>_nocompute must be set:

<subsys>_nocompute The value of this keyword determines whether the associated submodel has part in the simulation or not.
Possible values:
0 = include submodel computations

1 = exclude submodel computations
Standard value: 0, include
computations.

Note: When the *nocompute* option is set, the related submodel performs no actions except the initialisation of submodel variables.

Note: The user must be very aware of the consequences of switching off a submodel, because results of one submodel are often used as input for another.

Note: No preparations have to be made for the *nocompute* option.

8.5 Readfromfile

The user has the possibility of having selected submodels reproduce the exact same results as in a previous run. To achieve this, the keyword `<subsys>_readfromfile` has to be set for the desired submodel(s).

Attention: this keyword can only be used in combination with the keyword *nocompute*.

`<subsys>_readfromfile` The value of this keyword determines whether the associated submodel gets its results from a previous run or not.
Possible values:
0 = do *not* read submodels results from a previous run
1 = read submodel results from a previous run
Standard value: 0, do not read previous results.

Note: In the preparations, export files must have been generated for every year of the simulation.

Note: In the preparations, the keyword *exportgoal* must have been set to 'export for readfromfile'.

Note: The restart run itself may produce export files, but the keyword *exportgoal* must keep the value 'export for readfromfile'.

Appendix A File structures

Control files

Control files are found in the folders *dat* and *const* (see Sections 4.2.11 and 4.2.6). These files have the following structure. Each file is textual and consists of a number of blocks. Each block consists of a comment part and a keyword part. The comment part is a set of text lines each starting with '#', containing a short description of the keyword. The keyword part consists of a line that contains the sequence "<keyword> =F <value>".

Regional files

Regional files are found in the folders *input* (see Section 4.2.12) and *output*, and have *.scn* and *.out* as file-name extensions. These files are textual and have a special M format so that they can be directly viewed by means of the scenario viewer. They have the following structure.

Each file contains a header block, a number of time blocks, and a small trailer block.

The header block consists of two lines. The first line is a comment line that starts with an exclamation point, which is followed by the array type, the array name, the dimension name(s), a keyword *unit* followed by the data unit, and a keyword *label* followed by a short description. The second line is a line that contains the array type, the array name, and the array dimension, followed by '(t) =F ['.

Each time block consists of a line with the year number, which is followed by a number of lines with array values.

The trailer block merely consists of the symbol '];'.

Example:

```
! real AGRPRODC [NFCT,NR24T] (t); Unit= Gg/yr; Label=
Production of food crops
real AGRPRODC [8,46] (t) =F [
1970.
  30123. ...
];
```

Unformatted grid files

Unformatted grid files are found in several folders. For a description of their names, see the IMAGE Programming Standard. Each grid file is associated with one array; the structure is as follows. It is a binary Fortran direct-access file containing precisely one record. That record contains

one array instance. The length of the record is equal to the number of array elements.

Restart files

Restart files are found in the subdirectory structure under the directory *state*.

There are two types of restart files: regular restart files and unformatted grid files. Regular files may be either binary or textual. They have the following structure. Each file consists of a number of variable blocks, one for every exported variable. The variable block consists of several lines. The first line contains the length of the variable name. The second line contains the variable name itself. The third line contains the variable type and the fourth line the number of array dimensions. The latter number is zero for a single variable. If the number of dimensions is 1 or more, then the following lines each contain a dimension length. The block is closed by a number of lines with an array value on each line.

For the structure of unformatted grid restart files, see the description of unformatted grid files above.

Appendix B Calibration

In this appendix, a short overview of the calibration process is given. Detailed information can be found in the `IMAGEcalibration.doc` file (see Section 1.2).

To perform a calibration, four steps have to be taken by the user:

1. Preparation
In this phase, the user generates many data files by means of special utilities. The input files for these utilities are gathered from sources, such as FAOSTAT (<http://faostat.fao.org>).
2. Execution
In the execution phase, the IMAGE program is run in calibration mode (by setting the appropriate calibration run number in the file `calibration.dat` in the `dat`-directory) with data files made in the Preparation step. The output of these runs is copied to directories that are available for the actual simulation. See also the programmers guide for information about the way in which the calibration functionality is implemented in IMAGE.
3. Completion
Some files may contain unrealistic values. These have to be corrected by hand.
4. Create scenario
After the calibration, which only covers the period `timestart` – `timescen`, many of the input files can be created by using the ‘scenario cooker’ which is described in the file `IMAGEscenarioCooker.doc` (see Section 1.2).

Colophon

Responsibility

Netherlands Environmental Assessment Agency (PBL)

Authors

PBL (E.Stehfest, L. de Waal, R. Oostenrijk), VORtech

Lay out

Studio RIVM

Contact

Elke.Stehfest@pbl.nl

IMAGE-info@pbl.nl

IMAGE User Manual

This user manual contains the basic information for running the simulation model IMAGE (“Integrated Model to Assess the Global Environment”) of PBL. The motivation for this report was a substantial restructuring of the source code for IMAGE version 2.5. The document gives concise content information about the submodels, tells the user how to install the program, describes the directory structure of the run environment, shows how scenarios have to be prepared and run, and gives insight in the restart functionality.