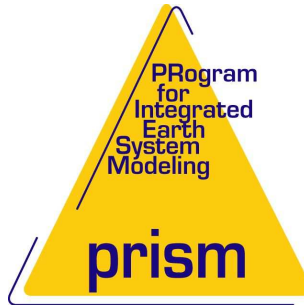


**PRISM**  
**Project for Integrated Earth System Modelling**  
**An Infrastructure Project for Climate Research in Europe**  
**funded by the European Commission**  
**under Contract EVR1-CT2001-40012**



**The PRISM TOYCLIM Coupled Model**  
Adaptation Guide

*Edited by:*  
*Sophie Valcke*

PRISM-Report Series-7

1st Edition  
(January 15th, 2005)

## Copyright Notice

© Copyright 2005 by CERFACS

All rights reserved.

No parts of this document should be either reproduced or commercially used without prior agreement by CERFACS representatives.

## How to get assistance?

Assistance can be obtained as listed below.

PRISM documentations can be downloaded from the WWW PRISM web site under the URL:

<<http://prism.enes.org>>

## Phone Numbers and Electronic Mail Adresses

Name	Phone	Affiliation	e-mail
Sophie Valcke	33 (0)5.61.19.30.76	CERFACS	

# Chapter 1

## Introduction

The PRISM project has been established to enhance the efficiency of Earth System modelling through development of a common coupling software infrastructure, standardized and portable compilation and running environments (SCE/SRE) and graphical user interfaces (GUIs). The coupling software was developed by CERFACS in collaboration with NEC-CCRLE, SGI, NEC-HPCE, Fecit/Fujitsu and IPSL within PRISM workpackage WP3A. It comprises the coupler OASIS3, its associated PRISM System Model Interface Library (PSMILe) for synchronized data exchange and model I/O in CF/NetCDF format, and its grid transformation library. A parallel coupler OASIS4 has also been developed in this workpackage (see (?)).

In order to test the OASIS3 coupler in a light coupled configuration, CERFACS has written 3 “toy” component models, mimicking an atmosphere model (toyatm), an ocean model (toyoce), and a chemistry model (toyche). These “toy” component models are ‘empty’ in the sense that they do not model any real physics or dynamics. The coupled combination of these 3 “toy” component models through OASIS3 coupling software is referred to as the TOYCLIM coupled model; the TOYCLIM coupling is realistic as the coupling algorithm linking the toy component models, the size and the grid of the 2D coupling fields, and the operations performed by OASIS3 on the coupling fields are realistic.

This report describes the 3 TOYCLIM component models (Chapter 2), their coupled combination (Chapter 3), how the TOYCLIM system can be compiled (Chapter 4), configured (Chapter 5) and run (Chapter 6). The TOYCLIM components and the PRISM software described herein are those tagged ‘prism\_2-4’ from the PRISM CVS repository. The PRISM software itself is documented in detail in other PRISM reports (see (1), (2), (3)). The demonstration runs using the TOYCLIM coupled model performed during the PRISM project are described in (4).



## Chapter 2

# Component Model Descriptions

### 2.1 The toyoce model

The toyoce model has a 2D logically-rectangular, stretched and rotated grid of 182x152 points, which corresponds to a real ocean model grid (the pole of convergence is shifted over Asia). Toyoce timestep is 14400 seconds; it performs therefore 36 timesteps per 6-day run.

OASIS3 PRISM System Model Interface (PSMILe) routines are detailed in detail in (3). At the beginning of a run, toyoce performs appropriate PSMILe calls to initialize the coupling, define its grids, and declare its I/O or coupling fields. As toyoce is not parallel, it calls the PSMILe `prism_def_partition` routine to define only one Serial partition containing the 182X152 grid points.

Then, toyoce starts its timestep loop. At the beginning of its timestep, toyoce calls the PSMILe `prism_get` routine 7 times to request the fields named Field3 to Field9 on table 3.1. At the end of its timestep, toyoce calls PSMILe `prism_put` routine to send fields named Field1 and Field2 on table 3.1. The fields will be effectively received or sent only at the coupling frequency defined by the user (see section 5.2). As toyoce contains no real physics or dynamics, it defines a simple feed back between Field1 and Field3 and between Field2 and Field7 such as:

$$Field1 = Field3 + 1$$

$$Field2 = Field7 + 1$$

Finally, at the end of the run, toyoce performs the PSMILe finalization call.

### 2.2 The toyatm model

The toyatm model has a realistic atmospheric T31 Gaussian grid (96x48 points). Its timestep is 3600 seconds; it therefore performs 144 timesteps per 6-day run.

As toyoce, toyatm performs, at the beginning of a run, appropriate PSMILe calls to initialize the coupling, define its grids, and declare its I/O or coupling variables. If MPI1 communication technique is chosen by the user, toyatm retrieves a local communicator for its internal parallelization with a call to PSMILe `prism_get_localcomm` routine. Toyatm can run on 1 or 3 processes, depending on the variable `il_nbcp_lproc` (hardcoded to 3 by default). If the user modifies this variable and hardcodes `il_nbcp_lproc = 1`, toyatm runs on only one process and defines only one Serial partition containing the 96X48 grid points. If `il_nbcp_lproc = 3`, toyatm runs on 3 processes and its decomposition depends on the `cdec` parameter, hardcoded to APPLE. In this case, each of the 3 toyatm processes calls the PSMILe `prism_def_partition` routine to define 1 segment of an APPLE decomposition (1536 grid points per segment). If the user changes the hardcoded value of `cdec` to BOX, each process will define 1 ‘box’ of a BOX decomposition; the first two processes treat a box of 64X24 points, while the third process treats

a box of 128X12 points. If the user hardcodes `cdec= 'ORANGE '`, each process will define a partition of two segments of 768 points distant of 1536 points.

Then, `toyatm` starts its timestep loop. At the beginning of its timestep, `toyatm` calls the PSMILe `prism_get` routine 3 times to request the fields named `Field1`, `Field2` and `Field11` on table 3.1. At the end of its timestep, `toyatm` calls PSMILe `prism_put` routine to send fields named `Field4` to `Field10` on table 3.1. The fields will be effectively received or sent only at the coupling frequency defined by the user (see section 5.2). As `toyatm` contains no real physics or dynamics, it defines a simple feed back between the different fields as:

$$\begin{aligned} Field5 &= Field1 + 1 \\ Field7 &= Field2 + 1 \\ Field10 &= Field11 + 1 \\ Field4 &= Field1 + 1 \\ Field8 &= Field2 + 1 \\ Field6 &= Field1 + 1 \\ Field9 &= Field2 + 1 \end{aligned}$$

Finally, at the end of the run, `toyatm` performs the PSMILe finalization call.

## 2.3 The toyche model

`Toyche` is integrated on the same atmospheric model grid than `toyatm`. Its timestep is 7200 seconds; it therefore performs 72 timesteps per 6-day run.

As the other `toymodels`, `toyche` performs, at the beginning of a run, appropriate PSMILe calls to initialize the coupling, define its grids, and declare its I/O or coupling variables; it also retrieves a local communicator if needed. As `toyche` has the same grid than `toyatm`, a direct exchange of coupling fields can occur between those two models, without going through OASIS3 interpolation process. To insure this, the coupling field must have a field status 'IGNORED' or 'IGNOUT' in the OASIS3 configuration file `namcouple` (see section 5.2) and the two models must have also the same parallel decomposition. `Toyche` decomposition is hardcoded the same way than `toyatm`, and if the user modifies the `toyatm` decomposition, he has to modify the `toyche` decomposition the same way by changing `toyche` values for `il_nbcplproc` and `cdec` (see section 2.2).

At the beginning of its timestep, `toyche` calls the PSMILe `prism_get` routine to request `Field10` (see table 3.1). At the end of its timestep, `toyche` calls PSMILe `prism_put` routine to send `Field11`. As `toyche` contains no real physics or dynamics, it defines a simple feed back between `Field11` and `Field10` such as:

$$Field11 = Field10 + 1$$

Finally, at the end of the run, `toyche` performs the PSMILe finalisation call.

## Chapter 3

# The TOYCLIM coupling algorithm

The coupling algorithm between the TOYCLIM component models toyoce, toyatm, and toyche is described here.

Table 3.1 lists the coupling fields exchanged between those 3 model components, giving the symbolic name used in each component and indicating whether the model produces the field (src) or receives it (tgt).

	toyoce	toyatm	toyche	restart	function
Field1	SOSSTSST (src)	SISUTESU (tgt)		fldo1.nc	$F_1$
Field2	SOICECOV (src)	SIICECOV (tgt)		fldo1.nc	$F_1$
Field3	SOALBEDO (tgt)			SOALBEDO.nc	
Field4	SONSHLDO (tgt)	CONSFTOT (src)		flda1.nc	$F_2$
Field5	<i>SOSHFLDO</i>	COSHFTOT (src)			$F_2$
Field6	SOWAFLDO (tgt)	COWATFLU (src)		flda1.nc	$F_2$
Field7	SORUNOFF (tgt)	CORUNOFF (src)		flda1.nc	$F_2$
Field8	SOZOTAUX (tgt)	COZOTAUX (src)		flda2.nc	$F_3$
Field9	SOMETAUY (tgt)	COMETAUY (src)		flda2.nc	$F_3$
Field10		COSENHFL (src)	SOSENHFL (tgt)	flda3.nc	$F_1$
Field11		COTHSHSU (tgt)	SOTHSHSU (src)	flda4.nc	$F_1$

**Table 3.1:** Coupling and I/O fields of the TOYCLIM coupled model. The symbolic name used in each toy model is given and it is indicated whether the model produces the field (src) or receives it (tgt). The function used to create the field in the initial restart file is also given.

Figure 3.1 illustrates the coupling algorithm between the 3 TOYCLIM toy models for  $Field_1$ ,  $Field_3$ ,  $Field_4$ ,  $Field_{10}$ , and  $Field_{11}$ .

$Field_1$  is sent from toyoce component to toyatm component at the coupling frequency  $dtF_1$  defined by the user in the configuring file *namcouple* (see 5.2). As interpolation is needed between toyoce and toyatm grids, this exchange must go through OASIS3 interpolation process. In the *namcouple*,  $Field_1$  field status must therefore be *EXPORTED* and the interpolation must be defined. If the user wants the field to be also automatically written to files before being sent (below the *prism\_put*), and after being received (below the *prism\_get*), he can choose the field status *EXPOUT*. In toyoce and toyatm codes, the *prism\_put* and *prism\_get* routines are respectively called every timestep with an argument corresponding to the time at the beginning of the timestep. The lag of  $Field_1$ , defined as 4 hours (14400 seconds) in the *namcouple*, is automatically added to the *prism\_put* time argument; the *prism\_put* called at the toyoce timestep preceding the coupling period therefore matches the *prism\_get* called in toyatm at the coupling period.

At the beginning of the run (i.e. at time = 0), the toyoce *prism\_put* for  $Field_1$  is not activated (as a positive lag is defined for  $Field_1$ ) and OASIS3 automatically read  $Field_1$  in its coupling restart file, *fldo1.nc*, and

sends it to toyatm component after interpolation. The different functions used to create the fields in the initial restart file are also indicated in table 3.1. They are defined as follows:

$$F_1 = 2 + \cos[\pi * a \cos(\cos(\theta)\cos(\phi))]$$

$$F_2 = 2 + \cos^2(\theta)\cos(2\phi)$$

$$F_3 = 2 + \sin^{16}(2\phi)\cos(16\phi)$$

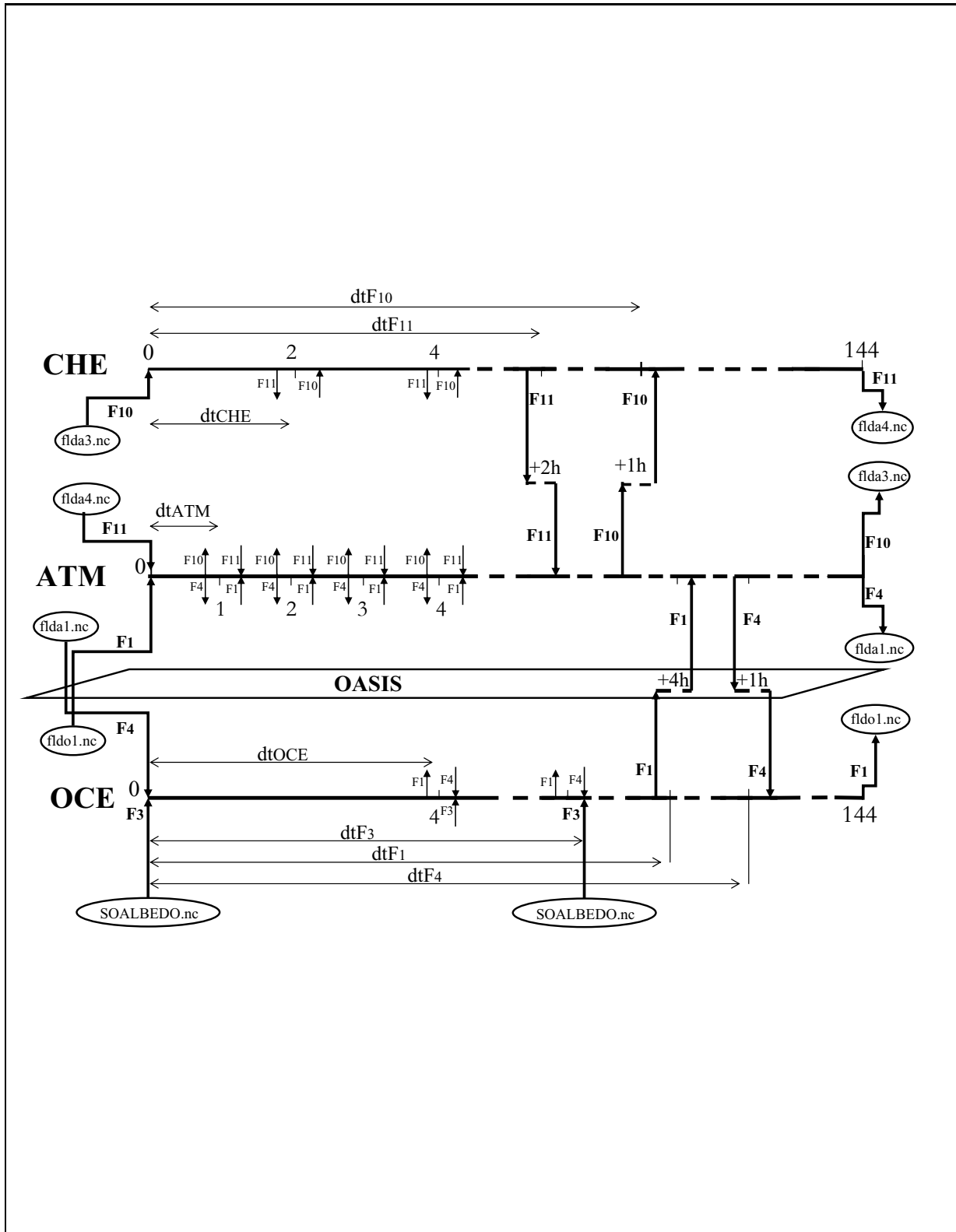
$F_1$  represents a global dipole reaching its maximum and minimum values at the equator, respectively at the date-line and at the Greenwich meridian.  $F_2$  represents two dipoles reaching respectively their maximum and minimum values at the equator, respectively at the date-line and at  $90^\circ$  W, and at the Greenwich meridian and at  $90^\circ$  E; it is similar to a spherical harmonic with  $l = 2$  and  $m = 2$ , where  $l$  is the spherical harmonic order and  $m$  is the azimuthal wave number.  $F_3$  represents a series of dipoles centered at  $45^\circ$  N and  $45^\circ$  S; it is similar to a spherical harmonic with  $l = 32$  and  $m = 16$  and is useful for testing interpolation of fields with relatively high spatial frequency and rapidly changing gradients.

The exchange of  $Field_2$  from toyoce to toyatm and  $Field_4$ ,  $Field_6$ ,  $Field_7$ ,  $Field_8$  and  $Field_9$  from toyatm to toyoce follow exactly the same logic as for  $Field_1$ .

$Field_3$  as a status INPUT in the *namcouple* (see 5.2).  $Field_3$  will therefore not be exchanged between two models but will be read from a file automatically below the target model toyoce prism\_get calls, at the user-defined frequency in the input file also specified in the *namcouple*, SOALBEDO.nc.

$Field_5$  as a status of OUTPUT in the *namcouple*. It will therefore be only automatically written to a file at the user-defined frequency, below the source model toyatm prism\_put calls. The name of the file will be automatically composed of the field symbolic name (here COSHFTOT) and of the begin and end dates of the run. The prism\_get calls in the toyoce model will not be activated at all.

$Field_{10}$  and  $Field_{11}$  are exchanged respectively from toyatm to toyche and from toyche to toyatm following  $Field_1$  logic described here above, except that the exchanges take place directly between the component models without going to OASIS3 interpolation process, as toyatm and toyche have the same grid and the same parallel partition. The fields status chosen by the user for those fields in the *namcouple* should therefore be IGNORED (or IGNOUT if the user wants the fields also automatically be written to files below the prism\_put and after the prism\_get). At the beginning of the run (i.e. at time = 0), the toyoce prism\_get called to receive those fields will automatically read the fields in their corresponding coupling restart files flda3.nc and flda4.nc.



**Figure 3.1:** Exchange algorithm between the 3 TOYCLIM component models for fields  $Field_1$ ,  $Field_3$ ,  $Field_4$ ,  $Field_{10}$ , and  $Field_{11}$ .



## Chapter 4

# Compiling the TOYCLIM coupled model

### 4.1 Getting the TOYCLIM coupled model sources, data, and scripts

The model sources, data, and related scripts for the TOYCLIM coupled model can be retrieved from the PRISM CVS repository, which presently is hosted on *bedano* in Zurich (see more details in (1) or refer to the PRISM web site at <http://prism.enes.org/>).

TOYCLIM coupled model sources and data can be retrieved with the following CVS module names :

*TOYCLIM*: sources of OASIS3, toyoce, toyatm and toyche, associated libraries, utilities, input and example output data files;

*TOYCLIMSRC*: only the sources of OASIS3, toyoce, toyatm and toyche, associated libraries and utilities;

*TOYCLIMATA*: only the input and example output data files.

### 4.2 Compiling the TOYCLIM coupled model sources

Once the sources, data and scripts are extracted, the next step is to create the compiling scripts for the coupler, the 3 toy models, and related libraries (see also (1) for more details on the PRISM Standard Compilation Environment):

1. Go in the directory `prism/util/compile/frames`.
2. Create the include files for your platform if they do not already exist in directory `prism/util/compile/frames/include_<node>` where `<node>` is the name of the platform. OASIS3 and its toyclim coupled model has been successfully compiled and run on Fujitsu VPP5000 (xbar), NEC SX5 (rhodes) and SX6 (SFV880), SGI Octane (elnino), IRIX MIPS (dcm23), Linux (total1), and IBM Power4 (hpca).
3. Generate a compile script for the libraries using the script `Create_COMP_libs.frm`:

```
Create_COMP_libs.frm "" "" "node"
```

The first parameter can be either "" or "-" to direct the standard output to a file or the screen.

The second parameter can be either "", "-", or "+" to direct the standard error to a file, the screen or the standard output.

If the compile scripts shall be created for another platform than the one where the Create\_COMP\_libs.frm script is launched, the third parameter has to contain the abbreviated node name "node".

4. Generate a compile scrip for OASIS3 and for each of the component models using the script Create\_COMP\_models.frm:

```

Create_COMP_models.frm oasis3 "mp"
Create_COMP_models.frm toyoce "mp" "" "" "node" "ID" "toyatm
toyoyce toyoyche"
Create_COMP_models.frm toyatm "mp" "" "" "node" "ID" "toyatm
toyoyce toyoyche"
Create_COMP_models.frm toyoyche "mp" "" "" "node" "ID" "toyatm
toyoyce toyoyche"

```

The second parameter "mp" specifies the message passing used, which determines how the models are launched (see also section 4.1 of (3)). If the default 'MPI2' is chosen, the string has to be empty (specification of MPI2 results in an error message). Otherwise, MPI1 has to be given. The OASIS3 executable will have the string MPI1 or MPI2 appended to its name. The 3 toy models can also be compiled with either the MPI1 option or the default MPI2 option (empty string).

The third parameter can be either "" or "-" to direct the standard output to a file or the screen.

The fourth parameter can be either "", "-", or "+" to direct the standard error to a file, the screen or the standard output.

If the compile scripts shall be created for another platform than the one where the Create\_COMP\_models.frm script is launched, the fifth parameter has to contain the abbreviated node name "node".

The sixth parameter "ID" is version acronym for differentiation of executables (not relevant for OASIS3 and TOYCLIM toy models).

Finally the last parameter gives the name of all the component models in the coupled constellation. This list is not relevant for OASIS3, but it has to be given for the toy models. The specified partner models are checked against allowed partners and no default is set.

5. The compilation scripts created can now be used to compile OASIS3 and the 3 toy models. All four compile scripts have then to be launched explicitly by the user.

```

COMP_oasis3_<mp>.<node>
COMP_toyatm_<ID>.<node>
COMP_toyoyce_<ID>.<node>
COMP_toyoyche_<ID>.<node>

```

The scripts compile the models with the MPI library specified during their generation. The script that triggers the update of the libraries is automatically called by the model compilation scripts for the libraries they need. Libraries needed by OASIS3 are anaisg, anaism, scrip, fscint, and clim. The toy models need psmile and mpp\_io.

6. The result should be executables oasis3.<mp>.x, toyatm.<mp>.x, toyoyce.<mp>.x, and toyoyche.<mp>.x in the \$BLDROOT/bin directory defined by the compile scripts, where <mp> is either MPI1 or MPI2.

## Chapter 5

# Configuring the TOYCLIM coupled model

### 5.1 Step-by-step configuration

To set-up and configure a TOYCLIM experiment, one has to go through the following steps:

1. Go to the directory `prism/util/running/frames`
2. Create the include files for your platform if they do not already exist in directory `prism/util/running/frames/include_<node>` where `<node>` is the name of the platform (see also 4.2).
3. Run the script `Create_TASKS.frm` to generate a setup file for your TOYCLIM experiment:

```
Create_TASKS.frm toyclim <expid>
```

where `<expid>` is your experiment name.

4. To change the configuration of your experiment, modify the values of the configurable entries in the setup file `prism/util/running/frames/setup/setup_toyclim_<expid>`, which contains default values for these entries (see detail in 5.2 below).
5. Type `'Create_TASKS.frm toyclim <expid>'` a second time. The script will check the parameters you specified in `setup_toyclim_<expid>`. If a parameter is not supported or a combination of parameters does not make sense `Create_TASKS.frm` writes an error message and stops.
6. Correct the experimental setup file if necessary and run `Create_TASKS.frm` again until the setup check is passed successfully.

Once the setup is done, all appropriate files and the script to start the experiment (see section 6) are available in the directory `<home>/<expid>`, where `<home>` and `<expid>` are defined in the setup file.

### 5.2 Configurable variables

Different entries of the setup file can be configured by the user. Some of these configurable entries directly enter the OASIS3 configuration file *namcouple*, other affect the running script only<sup>1</sup>.

The *namcouple* file is created from the *namcouple* base file (see appendix A) by replacing the configurable entries (which begin with `"#"`) by the value defined in the setup file (see 5.1). The *namcouple* will be read

---

<sup>1</sup>Additional *namcouple* entries are also configurable by editing directly the *namcouple* base file. Refer to (3) for more details.

by OASIS3 at runtime. The variables that can be defined in the set-up file and correspond to configurable *namcouple* entries are the following:

1. *jobname*: Experiment identifier; it is composed of three digits; (*#Jobname*).
2. *nlogprt*: Integer controlling the amount of information written to the OASIS output file *cpout*. 0: minimum output, 1: medium output, 2: maximum output; (*#Nlogprt*).
3. *extrapwr*: Flag to provoke the calculation of weights and addresses for nearest neighbour extrapolation (EXTRAP/NINENN) within OASIS (1) or to read them from file (0); (*#Extrapwr*).
4. *stat\_fieldxx*, where *xx* is the number of the field in the *namcouple*: The status of the *xx* coupling/I/O field can be either ‘EXPOUT’ or ‘EXPORTED’, except for *Field<sub>3</sub>* for which it is ‘INPUT’, for *Field<sub>5</sub>* for which it is ‘OUTPUT’ and for *Field<sub>10</sub>* and *Field<sub>11</sub>* for which it is either ‘IGNOUT’ or ‘IGNORED’ (see chapter 3); (*#Stat\_fieldxx*).
5. *dtFxx*, where *xx* is the number of the field in the *namcouple*: The coupling or I/O period of the *xx* coupling/I/O field, which must be a multiple of 14400, except for *Field<sub>3</sub>* for which it must be a multiple of 43200, and for *Field<sub>5</sub>*, *Field<sub>10</sub>* and *Field<sub>11</sub>* for which it must be a multiple of 3600; (*#Dtxx*).
6. *iniyear*, *inimonth* and *iniday*: the initial date of the experiment, respectively as YYYY, MM and DD(*#Inidate*).
7. *message\_passing*: Message passing method, either MPI2 or MPI1 (for more details, see section 4.2, or (3)); (in *#Channel*)
8. *bsend*: either ‘yes’ or ‘no’, indicates whether the buffered MPI\_BSend or the standard blocking send MPI\_Send will be used to send the coupling fields (for more details, see (3)) (in *#Channel*)

The variables that can be additionally defined in the setup file but do not correspond to any configurable *namcouple* entries are the following:

1. *ncplvers*: the *namcouple* version. Put " " to use the *namcouple* base file completed with the values defined in the setup file. To use another *namcouple*, a particular value has to be given to *ncplvers*, and a *namcouple* named *namcouple\_toyclim-<ncplvers>* has to be available in *prism/util/running/adjunct\_files/oasis3*.
2. *gridswr*: either ‘0’ if you want the models to use the grid description files if they exist, or ‘1’ if you want the models to unconditionally (re)generate those grid description files (for more details, see (3)).

## Chapter 6

# Running the TOYCLIM coupled model

After the setup and configuration phase, the experiment is ready to be started with the running script `RUN_toyclim_<expid>` in directory `<work>/<expid>/scripts`, where `<work>` and `<expid>` are as defined in the setup file `setup_toyclim_<expid>` (see section 5.1).

To run a TOYCLIM experiment, one has to go through the following steps:

1. Login on the compute server (if different from the compile server).
2. Change to the directory `<work>/<expid>/scripts`.
3. Submit the runscript `RUN_toyclim_<expid>`.

Running TOYCLIM with the default parameters will result in a 30-day experiment executed as five 6-day runs. The final results obtained in the directory `<work>/<expid>/work` should match the ones in `prism/data/toyclim/outdata`. Intermediate results are also saved in different sub-directories :

- the OASIS3 log files of each run in `<data>/<expid>/log` or `<archive>/<expid>/log`
- the output netCDF files containing the EXPOUT fields in `<data>/<expid>/outdata/oasis3` or `<archive>/<expid>/outdata/oasis3`
- the coupling restart files in `<data>/<expid>/restart` or `<archive>/<expid>/restart`.

where `<data>` and `<archive>` are defined in the runscript `RUN_toyclim_<expid>`.



# Appendix A

## The namcouple base file

The *namcouple* base file from which the OASIS3 configuration file *namcouple* will be created can be found in `prism/util/running/adjunct_files/oasis3/namcouple_toyclim`:

```
# This is a typical input file for OASIS 3.0, using netCDF format
# for restart input files. Oasis reads in this file at run-time.
#
# Any line beginning with # is ignored. Blank lines are not allowed.
#
$SEQMODE
# This keyword concerns the coupling algorithm. Put here the maximum number
# of fields that have to be, at one particular coupling timestep,
# exchanged sequentially in a given order.
    1
$END
#####
$CHANNEL
# The communication technique you want to use (MPI1, MPI2, or NONE).
# - if you want to use MPI1 or MPI2 message passing, you must write
# MPI1 or MPI2 (+ NOBSEND if you do not want to use the default MPI_BSend)
# + one line per model giving for the model the total number of procs,
# the number of procs implied in the coupling and, for MPI2 only, an
# optional launching argument
#
#Channel
#Mod1procs
#Mod2procs
#Mod3procs
$END
#####
$NFIELDS
# This is the total number of fields being exchanged.
# For the definition of the fields, see under $STRINGS keyword
#
    11
$END
#####
$JOBNAME
# This is an acronym for this run (3 characters).
#Jobname
$END
#####
$NBMODEL
# This gives you the number of models running in this experiment +
# their names (6 characters) + , in option, the maximum Fortran unit
# number used by each model; 1024 will be used if none are given.
#
    3 toyatm  toyoce  toyche  99  99  99
$END
#####
$RUNTIME
# This gives you the total simulated time for this run in seconds (here 6 days)
#
    518400
$END
#####
$INIDATE
```

```

# This is the initial date of the run. This is important only if FILLING
# analysis is used for a coupling field in the run. The format is YYYYMMDD.
#Inidate
$END
#####
$MODINFO
# Indicates if a header is encapsulated within the field brick
# in binary restart files for all communication techniques, and for
# coupling field exchanges for PIPE, SIPC and GMEM. (YES or NOT)
NOT
$END
#####
$NLOGPRT
# Index of printing level in output file cplout: 0 = no printing
# 1 = main routines and field names when treated, 2 = complete output
#Nlogprt
$END
#####
$CALTYPE
# Calendar type : 0      = 365 day calendar (no leap years)
#                 1      = 365 day, or 366 days for leap years, calendar
#                 n (>1) = n day month calendar
# This is important only if FILLING analysis is used.
#
#Caltype
$END
#####
$STRINGS
#
# The above variables are the general parameters for the experiment.
# Everything below has to do with the fields being exchanged.
#####
#                 OCEAN --->>>  ATMOS
#                 -----
# Field 1 : sea surface temperature
#
# First line:
# 1) and 2) Symbolic names for the field before and after interpolation
#           (8 characters maximum)
# 3) Index of field in cf_name_table.txt
# 4) Exchange frequency for the field in seconds (here 1 day)
# 5) Number of analysis to be performed
# 6) Restart input NetCDF file names
# 7) Field status (EXPORTED, AUXILARY, IGNORED, EXPOUT, IGNOUT, INPUT, OUTPUT)
#
SOSSTSST SISUTESU 1 #Dt01 5 fld01.nc #Stat_field01
#
# Second line:
# 1) and 2) locator prefix (4 characters) used to read the source and
# target grid characteristics in grid auxiliary files,
# + possibly LAG, SEQ, DEL, XTS additional indices if needed
#
topa at31 LAG=+14400
#
# Third line:
# Source grid characteristic (P or R) and number of overlapping grid points
# + Target grid characteristic (P or R) and number of overlapping grid points
P 2 P 0
#
# List of analyses
#
LOCTRANS CHECKIN MOZAIC CHECKOUT REVERSE
#
# Specific parameters for each analysis
#
AVERAGE
# CHECKIN: indicate computation of global, land and sea field integrals.
INT=1
# Mozaic: 1) mapping filename 2) connected unit 3) dataset rank 4) Maximum
# number of neighbors used
at31topa 91 2 48
#
# CHECKOUT: indicate computation of global, land and sea field integrals.
INT=1
# Reverse: 1) and 2) describes the ordering of the target model
NORSUD WSTEST
#

```

```

#####
#
#####
# Field 2 : Sea ice extent
#
SOICECOV SIICECOV 2 #Dt02 4 fldo1.nc #Stat_field02
topa at31 LAG=+14400
P 2 P 0
#
CHECKIN MOZAIC CHECKOUT REVERSE
#
INT=1
at31topa 91 2 48
INT=1
NORSUD WSTEST
#
#####
#
# --->>> OCEAN
# -----
#####
#
# Field 3 : Surface albedo
#
SOALBEDO SOALBEDO 17 #Dt03 0 SOALBEDO.nc INPUT
#
#####
#
# ATMOSPHERE --->>> OCEAN
# -----
#####
#
# Field 4 : Non solar heat flux
#
CONSFTOT SONSHLDO 6 #Dt04 7 flda1.nc #Stat_field04
at31 topa LAG=+3600
P 0 P 2
#
LOCTRANS INVERT CHECKIN MASK EXTRAP INTERP CHECKOUT
T_MIN
NORSUD WSTEST
INT=1
999999.9
NINENN 2 #Extrapwr 1
BICUBIC G SCALAR
INT=1
#
#####
#
#####
#
# Field 5 : Solar heat flux
#
COSHFTOT COSHFTOT 7 #Dt05 0 OUTPUT
at31 at31
#
#####
#
#####
#
# Field 6 : water flux
#
COWATFLU SOWAFLDO 29 #Dt06 7 flda1.nc #Stat_field06
at31 topa LAG=+3600
P 0 P 2
#
LOCTRANS INVERT CHECKIN MASK EXTRAP INTERP CHECKOUT
#
ACCUMUL
NORSUD WSTEST
INT=1
999999.9
NINENN 2 0 1
BICUBIC G SCALAR
INT=1
#
#####
#
#####

```

```

#
# Field 7 : Runoff
#
CORUNOFF  SORUNOFF  32 #Dt07   5  flda1.nc  #Stat_field07
  at31  topa  LAG=+3600
P  0  P  2
#
LOCTRANS  INVERT  CHECKIN  MOZAIC  CHECKOUT
#
ACCUMUL
NORSUD    WSTEST
INT=1
runoff31  93   1   41
INT=1
#
#####
#
# Field 8 : zonal wind stress -> ugrid
#
COZOTAUX  SOZOTAUX  23 #Dt08   7  flda2.nc  #Stat_field08
  at31  uopa  LAG=+3600
P  0  P  2
#
LOCTRANS  INVERT  CHECKIN  MASK  EXTRAP  INTERP  CHECKOUT
#
AVERAGE
NORSUD    WSTEST
INT=1
999.999
NINENN  2 0 1
BICUBIC  G  VECTOR
INT=1
#
#####
#
# Field 9 : meridional wind stress -> vgrid
#
COMETAUY  SOMETAUY  24 #Dt09   7  flda2.nc  #Stat_field09
  at31  vopa  LAG=+3600
P  0  P  2
#
LOCTRANS  INVERT  CHECKIN  MASK  EXTRAP  INTERP  CHECKOUT
#
AVERAGE
NORSUD    WSTEST
INT=1
999.999
NINENN  2 0 1
BICUBIC  G  VECTOR
INT=1
#
#####
#
#           ATMOSPHERE  --->>>  ATMOSPHERIC CHEMISTRY
#           -----
#####
#
# Field 10 : windmixing power
#
COSENHFL  SOSENHFL  37 #Dt10   1  flda3.nc  #Stat_field10
at31  at31  LAG=+3600
LOCTRANS
AVERAGE
#####
#           ATMOSPHERIC CHEMISTRY  --->>>  ATMOSPHERE
#           -----
#####
#
# Field 11 : chemical concentration
#
SOTSHSU  COTSHSU  49 #Dt11   0  flda4.nc  #Stat_field11
at31  at31  LAG=+7200
#
#####
#
$END

```

# Bibliography

- [1] Legutke, S. and V.Gayler, 2004: The PRISM Standard Compilation Environment. PRISM Report Series No 4.
- [2] Gayler, V. and S. Legutke, 2004: The PRISM Standard Running Environment. PRISM Report Series, No 5.
- [3] Valcke, S., A. Caubel, R. Vogelsang, and D. Declat, 2004: OASIS3 User Guide (oasis3\_prism\_2-4). PRISM Project Report No 2, 5th Ed., 60 pp.
- [4] Carril, A., R. Budich, S. Valcke, 2004: The TOYCLIM demonstration run report, PRISM Report Series No 13.

