

ORCHIDEE Training course

Technical introduction

6-7 February 2023, Jussieu

Josefine Ghattas, IPSL



ORCHIDEE
LAND SURFACE MODEL

Outline of presentation

ORCHIDEE different versions: trunk and branches

Coding Guidelines

Related configurations: offline and coupled

Install and compile: using modipsl and main compilation script

Experiences with libIGCM

Configure input and output files

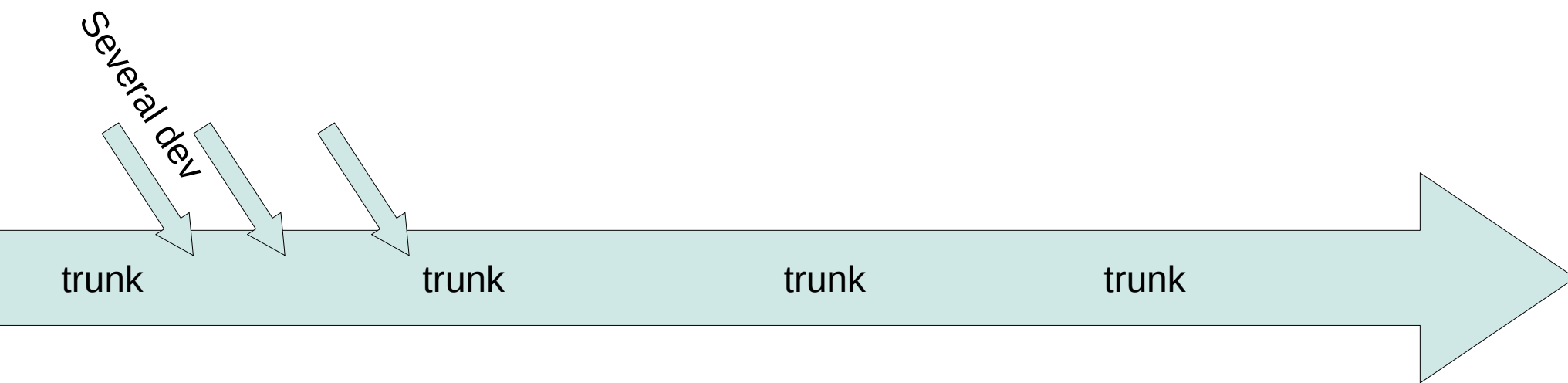
More about XIOS and creation of diagnostic output files

SVN: short guide

Finding information

ORCHIDEE trunk

= “main ORCHIDEE version evolving over time”



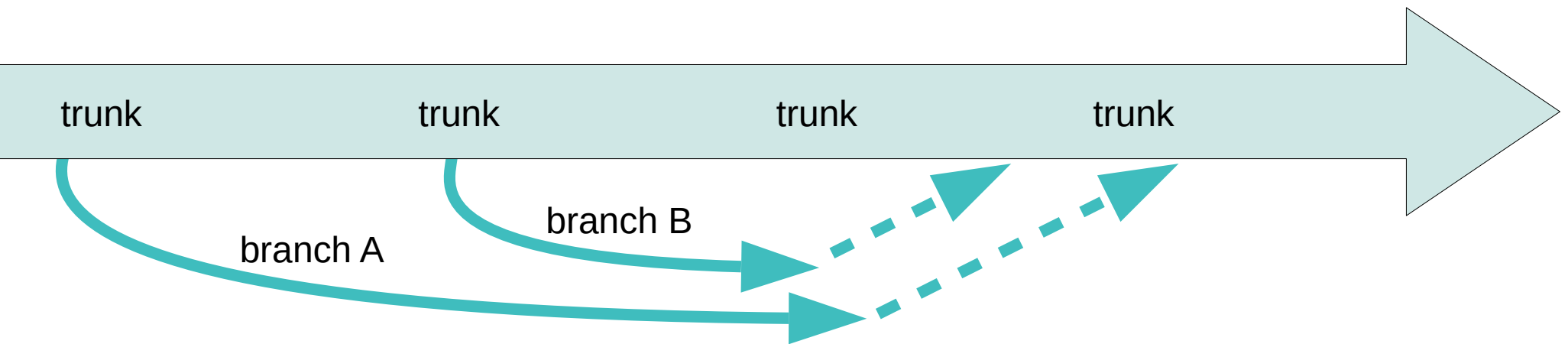
- The trunk is changing over time
- The trunk today and last week might not be the same
- You need to specify which specific revision of the trunk you use

time

ORCHIDEE branches

= “other versions of ORCHIDEE not yet in the trunk”

- A branch can be seen as a “temporary” version of the model used during the development and validation phase.
- A branch starts as a copy of the trunk and then the new developments are added. When the developments are finished and validated, if successful, the branch should be integrated in the trunk.

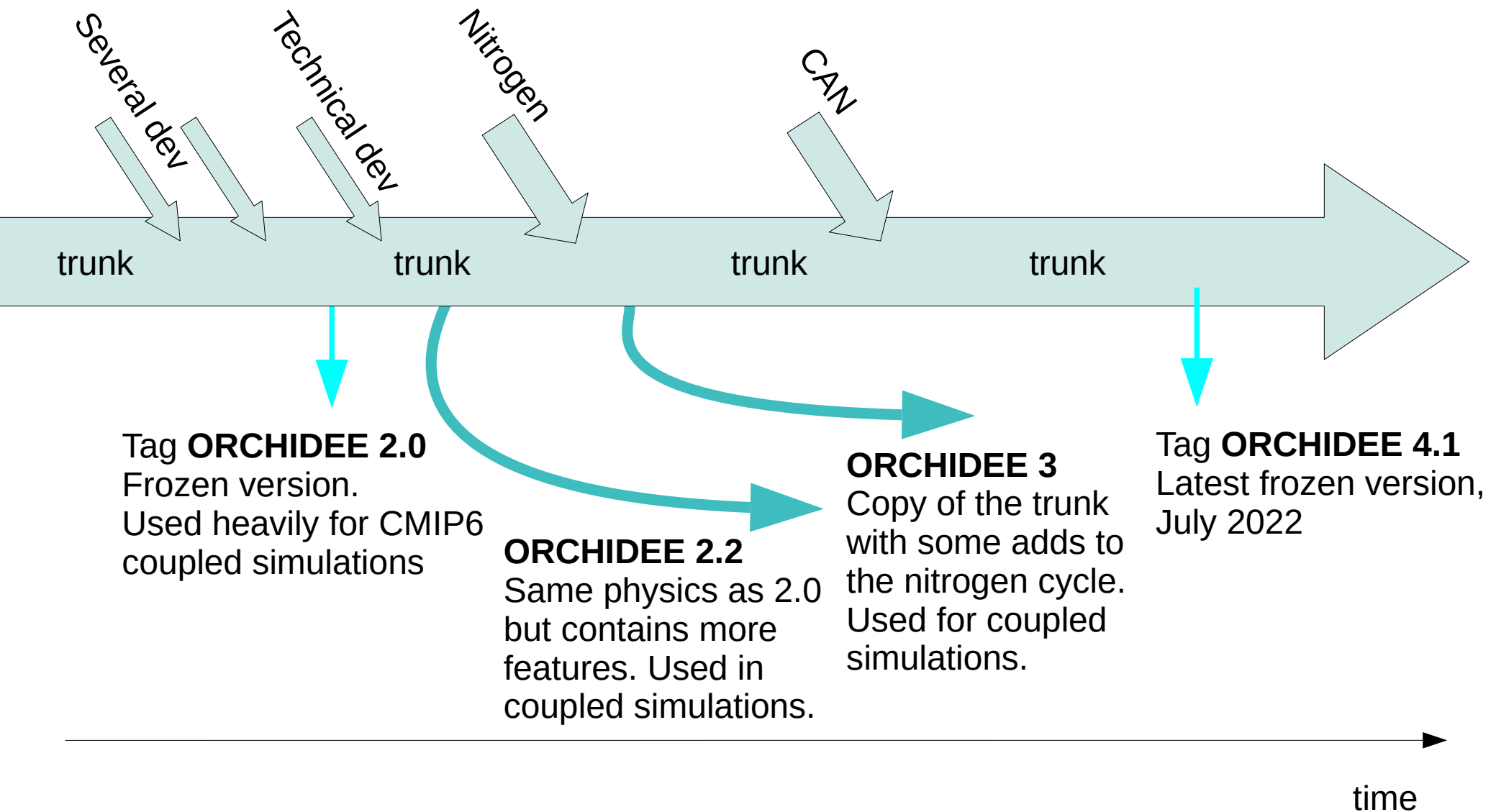


ORCHIDEE branches

= “other versions of ORCHIDEE not yet in the trunk”

- A branch can be seen as a “temporary” version of the model used during the development and validation phase.
- A branch starts as a copy of the trunk and then the new developments are added. When the developments are finished and validated, the branch should be integrated in the trunk.
- A branch is created when several people work together on a new development.
- When only one person work on a development, a personal version can be created. A personal version is technically the same as a branch.
- Each developer of ORCHIDEE can have a space on the SVN server to store one or several personal versions.
- Integration in the trunk of finalized developments must be planed in time with the ORCHIDEE project group.

Versions derivated from the trunk



Coding Guidelines

All new developments to be integrated in the ORCHIDEE trunk must follow the Coding Guidelines:

- **A bench of technical tests must be ok** (restartability, debug/prod mode, running on all platforms, ...)
- **Comments in english**
- **Indentation**
- **Key words in capital letters**
- **Contain a description part in each module and subroutine**
- ...

<http://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide/CodingGuidelines>

Use module `diffuco.f90` as example

Related configurations with ORCHIDEE

- A configuration contains the **model ORCHIDEE** and other codes needed to run the model such as **IOIPSL, XIOS and libIGCM**.
- There are **offline configurations** where only ORCHIDEE model is used and **coupled configurations** where ORCHIDEE is coupled to the atmospheric model LMDZ and sometimes other modeles as well.
- We use **modipsl** to predefine configurations that can be **install**. Modipsl is a tool developed at IPSL which we use to install the model.

Related configurations with ORCHIDEE

If you want to install :

- the trunk of ORCHIDEE

Use configuration **ORCHIDEE_trunk** or **LMDZOR_v6.4_work**

ORCHIDEE_trunk offline configuration contains the latest revision of the trunk. For new developments this is often the version to use but to be discussed with your colleagues/supervisors depending on the project.

- the branch ORCHIDEE_2_2

Use configuration **ORCHIDEE_2_2** or **LMDZOR_v6.2_work**

Close to ORCHIDEE_2_0 with some corrections and enhancements, includes possibility to be used with DYNAMICO.

- the tag ORCHIDEE_4_1

Use with offline configuration with the same name **ORCHIDEE_4_1**. No coupled configuration is predefined.

Different platforms

- Compiling options and running environment of ORCHIDEE are preconfigured at following platforms :

obelix at LSCE

irene at TGCC

jeanzay at IDRIS

spirit and **spiritx** at IPSL MESO-ESPRI

- Compiling at other machines need more time for installing pre-request (compiler, netcdf,..) and setting up compile options for ORCHIDEE.

How to install a configuration using modipsl

- **modipsl** is a tool used to install and compile predefined configurations, for example ORCHIDEE offline or ORCHIDEE coupled to LMDZ

- modipsl contains scripts for extraction of predefined configurations. modipsl is also a empty file tree that will receive the models and tools.

- use ***./model config*** to download a specific configuration

```
> svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
```

```
> cd modipsl/util
```

```
> ./model -h                # list predefined configurations
```

```
> ./model config           # extract a predefined configuration
```

Install a branch or personal version

You can also use modipsl to install other versions such as a branch, a personal version, or a specific revision of the trunk.

Modify in modipsl/util/mod.def :

Search for the section for the configuration which is the closest to your version of ORCHIDEE and change the specification for the version of ORCHIDEE.

For example for offline configuration, in modipsl/util/mod.def, modify line:

```
#-C- ORCHIDEE_trunk trunk/ORCHIDEE HEAD 14 ORCHIDEE modeles  
into  
#-C- ORCHIDEE_trunk branches/xxx/yyy revi_nb 14 ORCHIDEE modeles
```

Note, HEAD stands for the latest revision available and can be changed to a specific revision number.

Compiling ORCHIDEE

- Default compilation in production mode, containing optimization, to be used for simulations:
 - > `cd modipsl/config/ORCHIDEE_OL`
 - > `./compile_orchidee_ol.sh`
- With debug options, to be used to check your code and to search for errors:
 - > `./compile_orchidee_ol.sh -debug`
- For coupled models, a similar script is found in the corresponding config folder. For example, for LMDZOR_v6.2 configuration:
 - > `cd modipsl/config/LMDZOR_v6`
 - > `./compile_lmdzor.sh`

Compiling ORCHIDEE

- The main compilation script will launch compilation of all components: IOIPSL, XIOS, and finally ORCHIDEE (for the offline case)
- The compilation works as default at the machines that are maintained at IPSL: irene/TGCC, jeanzay/IDRIS, obelix/LSCE and spirit(x)/IPSL.
- Inside the script, another script **makeorchidee_fcm** is launched to compile **ORCHIDEE**. This compile script is based on the tool FCM.
- **Dependencies between modules are determined automatically.** No modifications are needed if you add a module in one of the existing src_ directories.

More about compiling ORCHIDEE

- Specific platform dependent compile options and paths are set in modipsl/modeles/**ORCHIDEE/arch/**. 2 files per platform: **arch-X.fcm** and **arch-X.path** where X corresponds to the current platform (X=ifort_LSCE for obelix, X=ifort_MESOIPSL for spirit and spiritx,..)
- Modules needed are set in modipsl/config/**ORCHIDEE_OL/ARCH/arch-X.env** file, X as before.
- The compilation script loads the modules from the arch-X.env file.
- **The same modules need to be loaded while running the model.** In the run script, or in the terminal (done by libIGCM) :
> source modipsl/config/ORCHIDEE_OL/ARCH/arch-X.env

Older version compile using gmake

- In older versions of the model, **a main Makefile was used instead of a compilation script**. In the same way as the compilation script, the main makefile will launch compilation of all components: IOIPSL, XIOS, and finally ORCHIDEE (for the offline case)
 - > cd modipsl/config/ORCHIDEE_OL
 - > gmake
- Open the makefile to see which arguments it can take.
- The modules needed for compilation must be loaded in the terminal before starting the compilation with the makefile.

libIGCM: a tool for running

- Running the model is done using **libIGCM**.
- libIGCM is a script library developed at IPSL and used to run all different type of configurations, coupled as well as offline.
- Several predefined experiments are available for each configuration.
- A training course in IPSL modeling tools and environment (modipsl and libIGCM) is set up each year. It is highly recommended to follow this training.

Offline experiences using libIGCM

There are different predefined experiments that you find in modipsl/config/ORCHIDEE_OL.

SPINUP_ANALYTIC_FG1, OOL_SEC_STO_FG1trans, OOL_SEC_STO_FG2 : Serie **FG** of experiments on **Forced Global** grid using CRUJRA forcing files.

SPINUP_ANALYTIC_FG1nd, OOL_SEC_STO_FG1transnd, OOL_SEC_STO_FG2nd : Same as above but using the orchideedriver « the new driver ».

SPINUP_ANALYTIC_FF1, OOL_SEC_STO_FF1trans, OOL_SEC_STO_FF2 : Serie **FF** of experiments on **Forced France** grid using SAFRAN forcing files. These experiments use the new driver.

Read and write with IOIPSL

IOIPSL is a fortran library developed at IPSL, used since long time in ORCHIDEE to

- **read input files** (call flinopen, call flinget)
- **read and write restart files** (call restget, restput)
- **read parameter files run.def** (call getin)
- (- **write output files** with diagnostic variables (call histdef, call histwrite) - obsolete)

(*) Reading and interpolating input files can also be done by XIOS by setting XIOS_INTERPOLATION=y in run.def.

Read parameters from run.def

```
SUBROUTINE slowproc_xios_initialize

  CHARACTER(LEN=255) :: filename
  LOGICAL           :: lerr
  REAL(r_std)      :: slope_noreinf
  LOGICAL          :: get_slope
  INTEGER          :: l

  IF (printlev>=3) WRITE(numout,*) 'In slowproc_xios_initialize'
  !! 1. Prepare for reading of soils_param file
  ! Get the file name from run.def file and set file attributes accordingly

  filename = 'soils_param.nc'
  CALL getin_p('SOILCLASS_FILE',filename)
```

Parameters and variables that need to be set at run time, can be coded in ORCHIDEE using: **CALL getin_p("VARNAME",var)**

"VARNAME" can now be set in one of the .def files: **run.def**, **orchidee.def** or **orchidee_pft.def** without recompilation of the model. Note that this function is case sensitive.

Convention in ORCHIDEE : use the same varname as the variable name

Files created by ORCHIDEE

Restart files

- Containing all state variables in ORCHIDEE at the last time step of the execution
- These files are needed as input to start next iteration
- sechiba_rest_out.nc, stomate_rest_out.nc
- driver_rest_out.nc only for orchidee_of “the old driver”
- Red and written by IOIPSL

Diagnostic output files

- Optional files containing variables from ORCHIDEE
- One file per frequency, different operations possible
- As many files as wanted, as many variables as wanted
- For example sechiba_history.nc, stomate_history.nc,...
- Produced by **XIOS or by IOIPSL** (not maintained)

Produce output files using XIOS

ORCHIDEE is installed and compiled together with **XIOS**.

XIOS is a tool which handles reading and writing of files. It is used to produce output netcdf files containing diagnostic variables used to analyse the simulations.

XIOS is a tool developed for the IPSL models to obtain better performances and more flexible management of output files. XIOS also read and interpolate files in ORCHIDEE but this is not used in the standard set-up.

In ORCHIDEE:

- **src_parallel/xios_orchidee.f90** : all interfacing to XIOS
- **src_xml** : directory with all xml files for running with XIOS

Inside ORCHIDEE

CALL xios_orchidee_send_field

Example from thermosoil_main:

```
USE xios_orchidee
```

```
REAL(r_std),DIMENSION (kjpindex)      :: soilflx  
REAL(r_std),DIMENSION (kjpindex)      :: surfheat_incr  
REAL(r_std),DIMENSION (kjpindex, ngrnd) :: ptn  
...
```

```
CALL xios_orchidee_send_field("ptn",ptn)  
CALL xios_orchidee_send_field("Qg",soilflx)  
CALL xios_orchidee_send_field("DelSurfHeat",surfheat_incr)
```

Syntax: **CALL xios_orchidee_send_field(field_id, field)**

field_id: a unique identifier, the same id is set in the field definition in parameter file field_def_orchidee.xml which must be present at run time
CHARACTER(len=*)

field: the variable to send to XIOS. The variable is on landpoint grid, it can have 1 or 2 supplementary axis:
REAL(r_std), DIMENSION(kjpindex) or
REAL(r_std), DIMENSION(kjpindex,:)

Convention in ORCHIDEE : use the same name for the id as the variable name

xml parameter files

To run ORCHIDEE with XIOS all diagnostic output files are configured through xml files. Following 5 files need to be present at each execution :

- **iodef.xml** => Main input file for XIOS. This file includes the 2 context files below
- **context_orchidee.xml** => Grid and axis information, include field and file def
- **context_input_orchidee.xml** => Specify all reading of input files. Reading with XIOS is optional but this file is needed for all cases

- **field_def_orchidee.xml** => Definition for each variable send in ORCHIDEE
=> Only change if added new variable in ORCHIDEE

- **file_def_orchidee.xml** => Specify all output files and their variables
=> Change to set your output level
=> Remove variables, change levels, change freq.

The above xml file are stored in ORCHIDEE/src_xml directory.

field_def_orchidee.xml

- Definition for each variable in ORCHIDEE with a call `xios_orchidee_send_field`
- Only change if added a new call `xios_orchidee_send_field`

```
<field id="npp" name="npp" long_name="Net Primary Production" unit="gC/m^2/s" grid_ref="grid_nvm"/>
<field id="q_cdrag" name="cdrag" long_name="Drag coefficient for LE and SH" unit="-"/>
<field id="soilalb_vis" name="soilalb_vis" long_name="Soil Albedo visible" unit="1"/>
<field id="soilalb_nir" name="soilalb_nir" long_name="Soil Albedo near infrared" unit="1"/>
<field id="z0m" name="z0m" long_name="Surface roughness for momentum" unit="m"/>
<field id="z0h" name="z0h" long_name="Surface roughness for heat" unit="m"/>
<field id="albedo_vis" name="alb_vis" long_name="Albedo visible" unit="1"/>
<field id="albedo_nir" name="alb_nir" long_name="Albedo near infrared" unit="1"/>
<field id="albedo_glob" name="albedo_glob" long_name="Mean albedo" unit="1"> (albedo_vis +
albedo_nir)*0.5 </field>
```

```
...
...
...
...
...
```

file_def_orchidee.xml

```
<!-- ===== -->
<!-- file_def_orchidee.xml : Definition of output files -->

<file_definition type="one_file" par_access="collective" enabled=".TRUE." min_digits="4">

  <!-- Sechiba file 1 -->
  <file id="sechiba1" name="sechiba_history" output_level="11" output_freq="1d" enabled=".TRUE.">
    <field field_ref="Areas" level="1"/>
    <field field_ref="LandPoints" level="1"/>
    <field field_ref="Contfrac" level="1"/>
    <field field_ref="evap" level="1"/>
    <field field_ref="coastalflow" level="1"/>
    <field field_ref="riverflow" level="2"/>
    <field field_ref="temp_sol_C" level="2"/>
    ...
  </file>

  <!-- Sechiba file 2 -->
  <file id="sechiba2" name="sechiba_out_2" output_level="2" output_freq="1d" enabled=".TRUE.">
    <field field_ref="Areas" level="1"/>
    <field field_ref="LandPoints" level="1"/>
    <field field_ref="Contfrac" level="1"/>
    <field field_ref="mrsos" level="1"/>
    <field field_ref="mrro" level="2"/>
    ...
  </file>

  <!-- Stomate file 1 -->
  <file id="stomate1" name="stomate_history" output_level="10" output_freq="86400s">
    <field field_ref="RESOLUTION_X" level="1"/>
    <field field_ref="RESOLUTION_Y" level="1"/>
    <field field_ref="CONTRAC_STOMATE" level="1"/>
  </file>
</file_definition>
```

Add a new variable in ORCHIDEE

1) Add in the ORCHIDEE module where the variable is calculated:

CALL xios_orchidee_send_field("new_var",new_var)

2) In **field_def_orchidee.xml** : add declaration of the variable

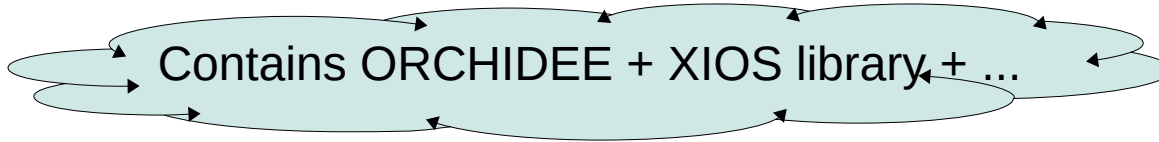
3) In **file_def_orchidee.xml** : add the variable in all file sections where you want to write it

*) If the variable is only calculated for a specific option, add an exception in **xios_orchidee_init**. This avoid that the variable will be initialized in the output file without being written if using the same .xml files.

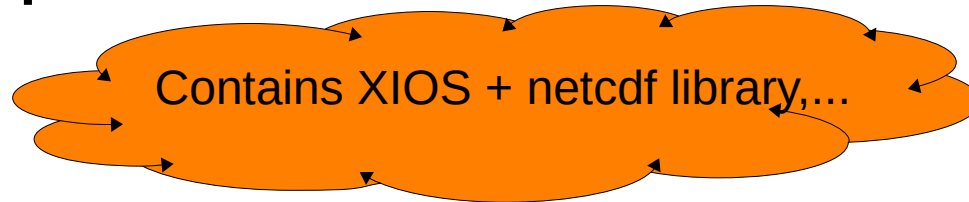
Running ORCHIDEE with XIOS in attached or server mode

After compilation 2 executables are found in modipsl/bin

orchideedriver_prod (“new driver”) or **orchidee_ol_prod** (“old driver”)



xios_server_prod.exe

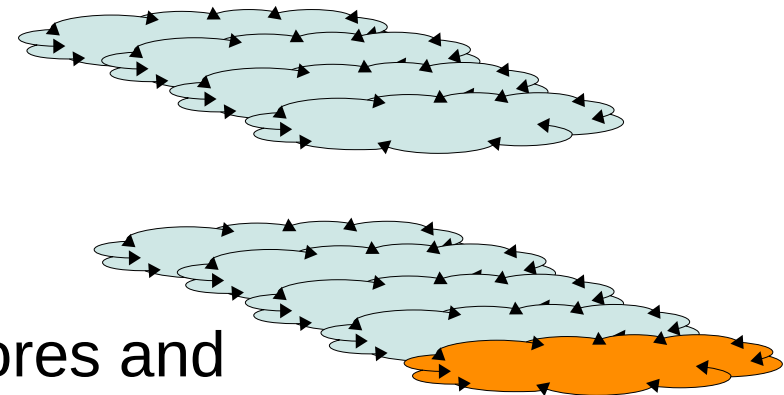


Launch only orchideedriver on one or several cores, this is called to use

XIOS in attached mode

Launch orchideedriver on one or several cores and the executable XIOS, this is called to use

XIOS in server mode



Using libGCM configurations server mode by default

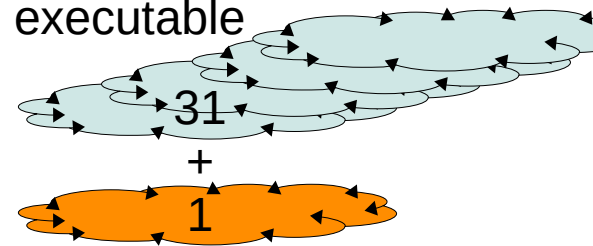
Default mode is using 1 server XIOS in libGCM configurations (ORCHIDEE_trunk, LMDZOR_v6, IPSLCM6)

config.card

- Component IOS represents XIOS
- Set number of cores MPI for each executable with 1MPI for the xios server.

```
#####  
#D-- ListOfComponents -  
[ListOfComponents]  
#D- For each component, Name of component, Tag of component  
SRF= (sechiba, orchidee_trunk)  
SBG= (stomate, orchidee_trunk)  
OOL= (orchidee_ol, orchidee_trunk)  
IOS= (xios, XIOS)  
  
#D-- Executable -  
[Executable]  
#D- For each component, Real name of executable  
SRF= ("", "")  
SBG= ("", "")  
OOL= (orchidee_ol, orchidee_ol, 31MPI)  
IOS= (xios_server.exe, xios.x, 1MPI)  
...  
  
#D-- IOS -  
[IOS]  
WriteFrequency=""  
Restart= n  
RestartDate=  
RestartJobName=  
RestartPath=  
...  
...
```

Number of core MPI
per executable



Using libIGCM configurations

Activating and setting output frequency can be done from **COMP/sechiba.card** and **COMP/stomate.card**:

```
# Specify output level for output files
# output_level_filename=0   : lowest level writing only variables needed for the monitoring
# output_level_filename=12  : highest level which will output all variables
# output_level_filename=NONE : deactivate the file
output_level_sechiba_history = 11
output_level_sechiba_out_2 = NONE
output_level_sechiba_history_4dim = 1

# Specify output frequency for each file [1y, 1mo, 1d, 10800s, 1ts]
# Settings using WriteFrequency in config.card are not longer used
output_freq_sechiba_history = 1mo
output_freq_sechiba_out_2 = 10800s
output_freq_sechiba_history_4dim = 1mo
```

Subversion (SVN) - a version control software

- Store different versions of ORCHIDEE, also scripts and other tools
- Keep track of changes done over the time
- Makes it easier to work in a group on the same version and exchange developments (“branches”) before inclusion in the main version (“trunk”)
- Archive the work done by phd-students, post-docs, researchers,.. (stored in “perso” folder)
- **Make sure that the code is on SVN for important simulations**
 - > **this is your reference while writing papers**

ORCHIDEE wiki:

<http://forge.ipsl.jussieu.fr/orchidee>

 Search

logged in as jgipsl | [Logout](#) | [Help/Guide](#) | [About Trac](#) | [Preferences](#)

- Wiki
- Timeline
- Roadmap
- Browse Source
- View Tickets
- New Ticket
- Search
- Admin

wiki: [WikiStart](#)

[Start Page](#) | [Index](#) | [History](#)

Model developments | Documentation | Source Code | Reference Simulations | Group Activities & Contact

Wiki of ORCHIDEE model

This wiki aims at gathering information on ORCHIDEE model : code versions and documentation, configurations used by the model, evaluation, seminars... Information is organized according different sections which are described below :

Section	Description	Highlights or short cuts to sub-sections
Model Developments	In this section, you will find all the informations on the ongoing developments	CMIP6
Documentation	'News', Scientific documentation, Information on the implementation of the code, Users guide/How To, Informations on the forcing	UserGuide/How To
Source Code	Where you will find the source code of the different versions, restricted access for some ongoing developments	See the trunk here.
Reference Simulations	All the information on evaluation protocol and reference simulation	validation simulation with rev 2724
Group Activities	Include: ORCHIDEE-POLICY , meetings, seminars, users list, contact, training courses	Training courses

Web-interface of the ORCHIDEE svn server

<https://forge.ipsl.jussieu.fr/orchidee/browser>

Login to see also read protected directories

forge.ipsl.jussieu.fr/orchidee/browser/tags/ORCHIDEE_2_1/ORCHIDEE

logged in as jgipsi | Logout | Help/Guide | About Trac | Preferences

Wiki | Timeline | Roadmap | **Browse Source** | View Tickets | New Ticket | Search | Admin

source: tags / ORCHIDEE_2_1 / ORCHIDEE

Name	Size	Rev	Age	Author	Last Change
..					
arch		5574	2 weeks	josefine.ghattas	Change in compilation : * Now always put -L and -I in arch.path file ...
DOC		4384	18 months	josefine.ghattas	Clairifications on t2m variable names. No changes in the calculation. - ...
src_driver		5609	9 days	josefine.ghattas	Integrated correction done in changeset [5329] in ...
src_global		5566	3 weeks	josefine.ghattas	Corrected call to ipsierr_p into ipsierr when called only from ...
src_oasisdriver		5573	2 weeks	josefine.ghattas	Do not use t2m/q2m coming from the atmospheric model anymore and instead ...
src_parallel		5573	2 weeks	josefine.ghattas	Do not use t2m/q2m coming from the atmospheric model anymore and instead ...
src_parameters		5605	10 days	josefine.ghattas	Added new option VEGETMAP_RESET. This option should be used to to change ...
src_sechiba		5613	9 days	josefine.ghattas	Added reading of variable pond from floodplains.nc, see ticket #441 A ...
src_stomate		5614	9 days	josefine.ghattas	Add reinitialization of leaf_age and leaf_frac. See ticket #444 A. Jomet
src_xml		5628	2 days	josefine.ghattas	Updated version of DataRequest? for CMIP6 as done in the tag ORCHIDE_2_0 ...
tools		1513	5 years	josefine.ghattas	- Added FCM version PATCHED/V1.2 in directory tools. - Modified ...
AA_make	1.5 KB	1249	6 years	josefine.ghattas	Correction for "make clean" target.
AA_make.ldef	82 bytes	12	8 years	mmaipsi	correct Id, HeadURL, Date, Author and Revision svn properties.
bld.cfg	2.3 KB	5523	6 weeks	josefine.ghattas	Changes related to the compilation: * Move variables CONFIG_PATH from ...
Doxyfile_ORCHIDEE.init	10.4 KB	947	6 years	didier.solyga	Merge Hydrology branch into ORCHIDEE trunk version.
makeorchidee_fcm	18.6 KB	5576	2 weeks	josefine.ghattas	Add link to lib and inc folder
orchidee.default	71.6 KB	5608	9 days	josefine.ghattas	Update orchidee.default due to changes in rev [5605]
ORCHIDEE_CeCILL.LIC	2.1 KB	8	8 years	orchidee	import first tag equivalent to CVS orchidee_1_9_5 + OOL_1_9_5

View changes...

Click to compare 2 directories

Note: See [TracBrowser](#) for help on using the repository browser.

Click to show modifications done in this directory or file

2 different logins are needed

<http://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide/DifferentLogin>

1- Login SVN

SVN anonymous login

« sechiba »

Only for extracting public versions of ORCHIDEE : trunk and tags

SVN personal login

« firstname.lastname »

To extract all versions where you have the permission

To be used to commit changes

2- Login “forge”

Forge is a machine at IPSL where the orchidee project is technically managed using trac, wiki and svn.

A specific login at forge is needed to use web-interface for your personal folder and private branches.

Same login is also used to modify the wiki and to create tickets.

Ask for login by sending a mail to [orchidee-help @ ipsl.jussieu.fr](mailto:orchidee-help@ipsl.jussieu.fr)

svn info

Information will be printed on the screen about extracted version

Example :

```
> cd modipsl/modeles/ORCHIDEE  
> svn info
```

```
Chemin : .  
URL : svn://forge.ipsl.jussieu.fr/orchidee/tags/ORCHIDEE_1_9_6/ORCHIDEE  
Racine du dépôt : svn://forge.ipsl.jussieu.fr/orchidee  
UUID du dépôt : f489ceea-5127-0410-b15c-c4a6149ed9a7  
Révision : 881  
Type de nœud : répertoire  
Tâche programmée : normale  
Auteur de la dernière modification : didier.solyga  
Révision de la dernière modification : 880  
Date de la dernière modification: 2012-05-09 16:08:00 +0200 (mer. 09 mai 2012)
```

In this example the version of ORHCIDEE is tags/ORCHIDEE_1_9_6 and the revision is 881.

svn stat

Local version: The current version on your computer that you just modified

Extracted version: The original version that you downloaded before modifications

Latest version on the server: the original version as it is currently on the server

svn stat Compare “local version” with the “extracted version”

```
> svn stat
?      Makefile
?      src_sechiba/Makefile
M      src_sechiba/intersurf.f90
M      src_sechiba/enerbil.f90
```

svn -u stat Compare “local version” to “latest version on the server”

```
> svn -u stat
?      Makefile
?      src_sechiba/Makefile
M      src_sechiba/intersurf.f90
*      src_sechiba/routing.f90
M      * src_sechiba/enerbil.f90
```

```
> svn help stat
'A' Added
'C' Conflicted
'D' Deleted
'M' Modified
'?' item is not under version control
'!' item is missing
* a newer revision exists on the server
```

svn diff

svn diff Show local modifications compared to extracted version

Example :

```
> svn diff
```

```
Index: src_sechiba/intersurf.f90
```

```
=====
--- src_sechiba/intersurf.f90 (revision 1054)
+++ src_sechiba/intersurf.f90 (working copy)
@@ -257,6 +257,7 @@
     itau_sechiba = kjit + itau_offset
     !
-    CALL intsurf_time( itau_sechiba, date0_shifted, xrdt )
+    CALL intsurf_time( itau_sechiba, date0_shifted, xrdt, toto )
+    WRITE(numout,*) 'toto'
     !
     ! 1. gather input fields from kindex array
```

Lines starting with “+” are added in the local version (also called working copy). Lines starting with “-” are removed. In this example, the line “CALL intsurf....” has been modified and the line “WRITE(...” has been added.

svn revert

svn revert *one_or_several_files*

Get back to the version of the file on the server, without your changes.

For example:

```
> svn revert src_sechiba/intersurf.f90
```

svn update

svn update [-r X] : Update working copy with the latest revision or revision X on the server

- Updates only with changes on the same branch
- Local changes will be kept. Conflicts can occur if the same file is modified locally and on the server
- Changes are done only in the local working directory

If you extracted a tag or a branch, changes done on the trunk will not be added in your directory.

If there is a conflict on a file, type p for postpone. svn will then save your modifications in a separate file. The file without modifications is also saved in your directory.

Commit to svn

svn add newfile.f90

Add locally new files and/or directories under version control. They will be added on the server in next commit

svn rm file.f90

The file will be removed locally and the file is scheduled to be removed from the svn repository in next commit.

svn commit
(or **svn ci**)

Commit all changes to the server
The revision number is increased.

Best practice for commit to svn

<https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/UserGuide/CommitOnTrunk>

- Prepare before commit : ***Clean your code, comment it, follow ORCHIDEE coding guidelines***
- Update to the latest revision on your version (branch or trunk)
- Add a log message to each commit
- Commit all files concerned by the modification in the same commit, avoid to commit file by file
- Discuss with the people concerned before commit

Summary - Example of a work cycle

- > svn info # Which version did you extract ?
- > svn stat # Verify the files you changed
- > svn diff # Verify each change in all files
- > svn revert toto.f90 # Clean if modifications not needed
- > svn -u stat # Check if up to date
- > svn update # If needed, update working copy
- > svn stat/ svn diff # Check again after update
- > svn add / svn rm # Declare adds or removes

- > svn ci # Commit all changes into the server

Your changes are now on the svn repository. Tell your colleagues so they can update their version of the branch.

Finding information

Wiki and web site

ORCHIDEE official web site (update seldom)

<http://orchidee.ipsl.fr>

ORCHIDEE wiki (updated frequently)

On the wiki you find useful information about on-going developments and help to use the model. Technical information in Documentation/UserGuide

<http://forge.ipsl.jussieu.fr/orchidee/wiki>

You need a **“login forge” to write on the wiki**. This login is also needed to see the full content of the wiki and also to see the SVN repository on the web interface. Write to orchidee-help to get a login.

Finding information @listes.ipsl.fr

All ORCHIDEE user's are invited to subscribe to the email lists:
orchidee-dev Discussion and information about ORCHIDEE
platform-users Ask and answer questions about libIGCM
Information about IPSL-cmc tools

2 email addresses for contact:

orchidee-help For technical questions

orchidee-projet To contact the ORCHIDEE project team

See how to subscribe :

<http://forge.ipsl.jussieu.fr/orchidee/wiki/GroupActivities/Contact>

Finding information

“Developer's meeting”

All users and developers are welcome to ORCHIDEE developer's meetings organized several times during the year. These meetings consist in a presentation of a specific topic followed by discussions and questions. Meeting place at Jussieu/Paris or LSCE/Saclay but often a videoconference is set up.

See reports and presentations here :

<http://forge.ipsl.jussieu.fr/orchidee/wiki/GroupActivities/Meetings>

Information about these meetings are done at orchidee-dev email list.