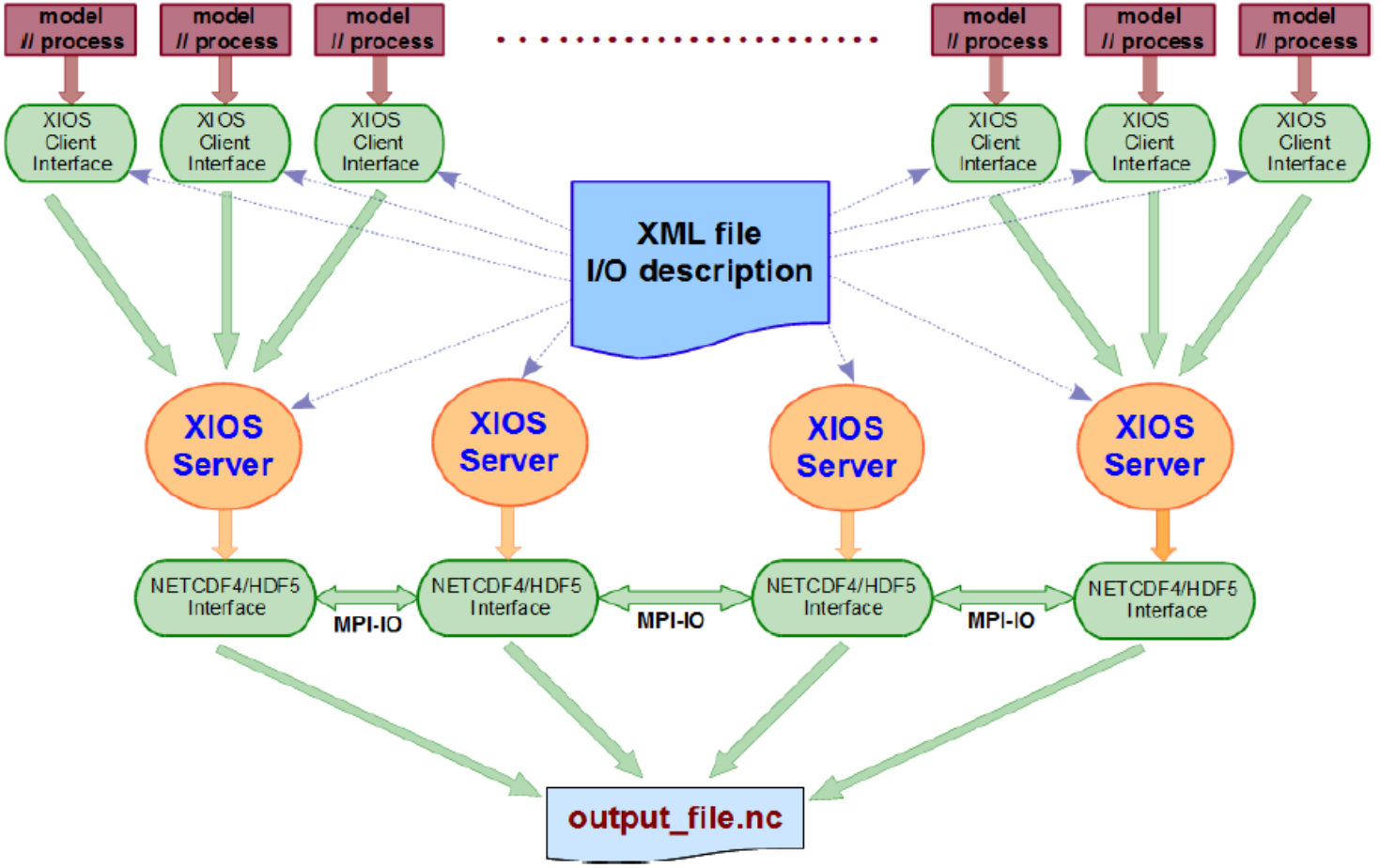
XIOS in ORCHIDEE
Implementation and how to use

Thanks to
Arnaud Caubel, first implementation in ORCHIDEE
Yann Meurdesoif, main developer of XIOS

Presentation by Josefine Ghattas
ORCHIDEE-DEV 31 mars 2015
Updated 4 nov 2015

XIOS in some words

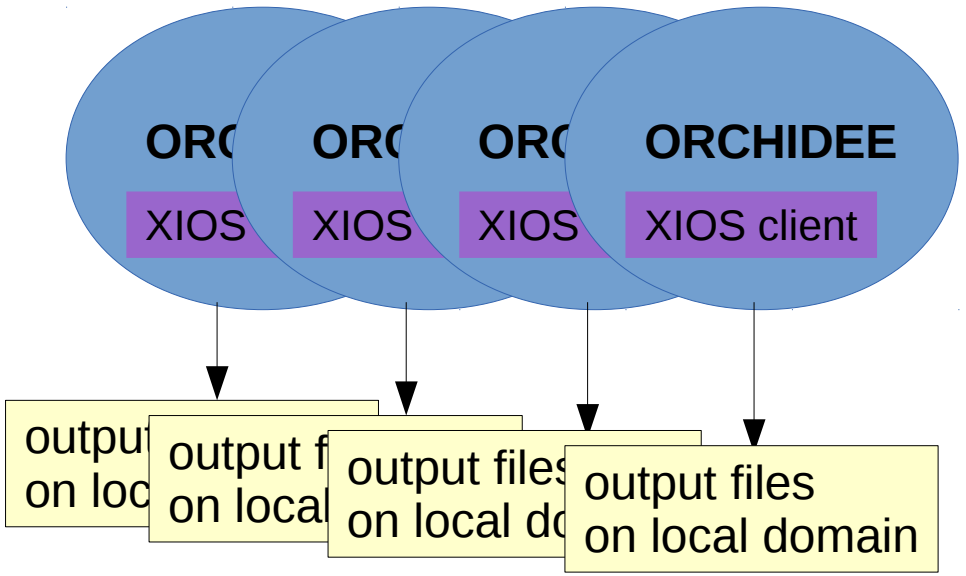
- Library dedicated to **IO management of climate codes**, developed at IPSL by Y. Meurdesoif
- **XML** configuration file
- **Attached** mode (library) or **server** mode (asynchronous transfer), multiple (sequential writing) or single (parallel writing) output file
- **NetCDF** format (GRIB2 in progress, ICHEC collaboration)



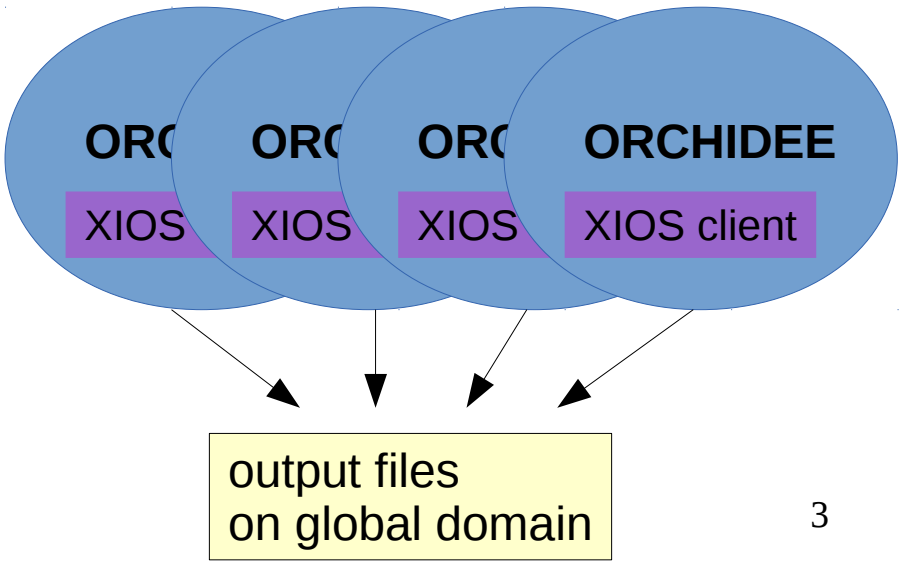
Attached mode or with server

- **<u>Attached mode:</u>**
  compile and link with XIOS library
  One executable: orchidee_ol or gcm.e
  XML : using_server=false
  file_definition type=”multiple_file” => rebuild needed
  file_definition type=”one_file” => no rebuild needed
- Advantage: Easy to use, for test cases, **sequential usage**

type=”multiple_file”

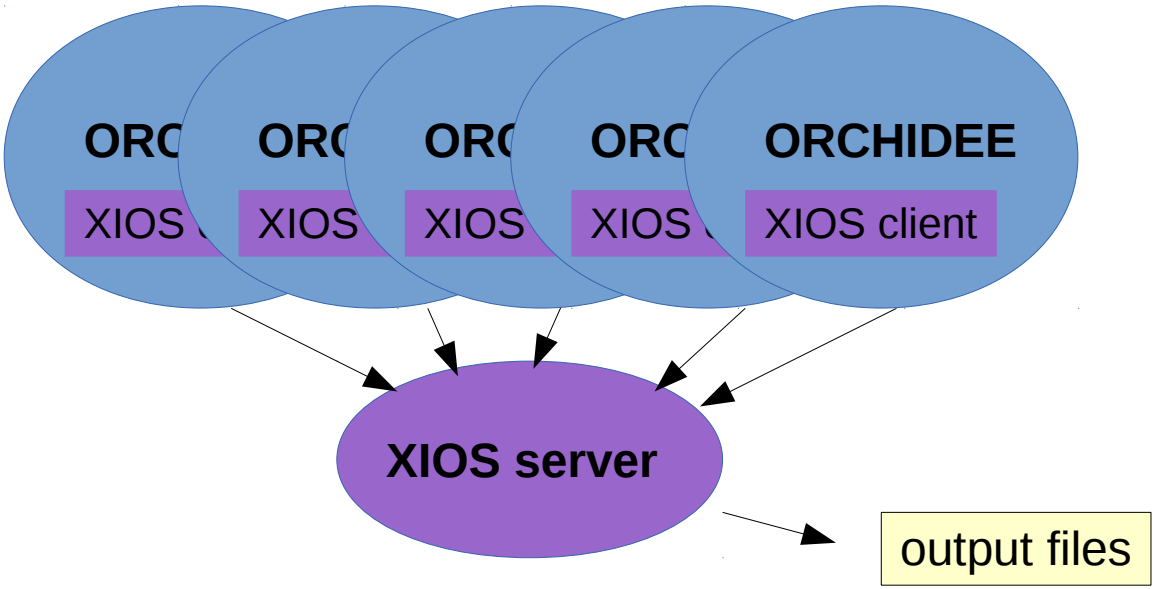


type=”one_file”



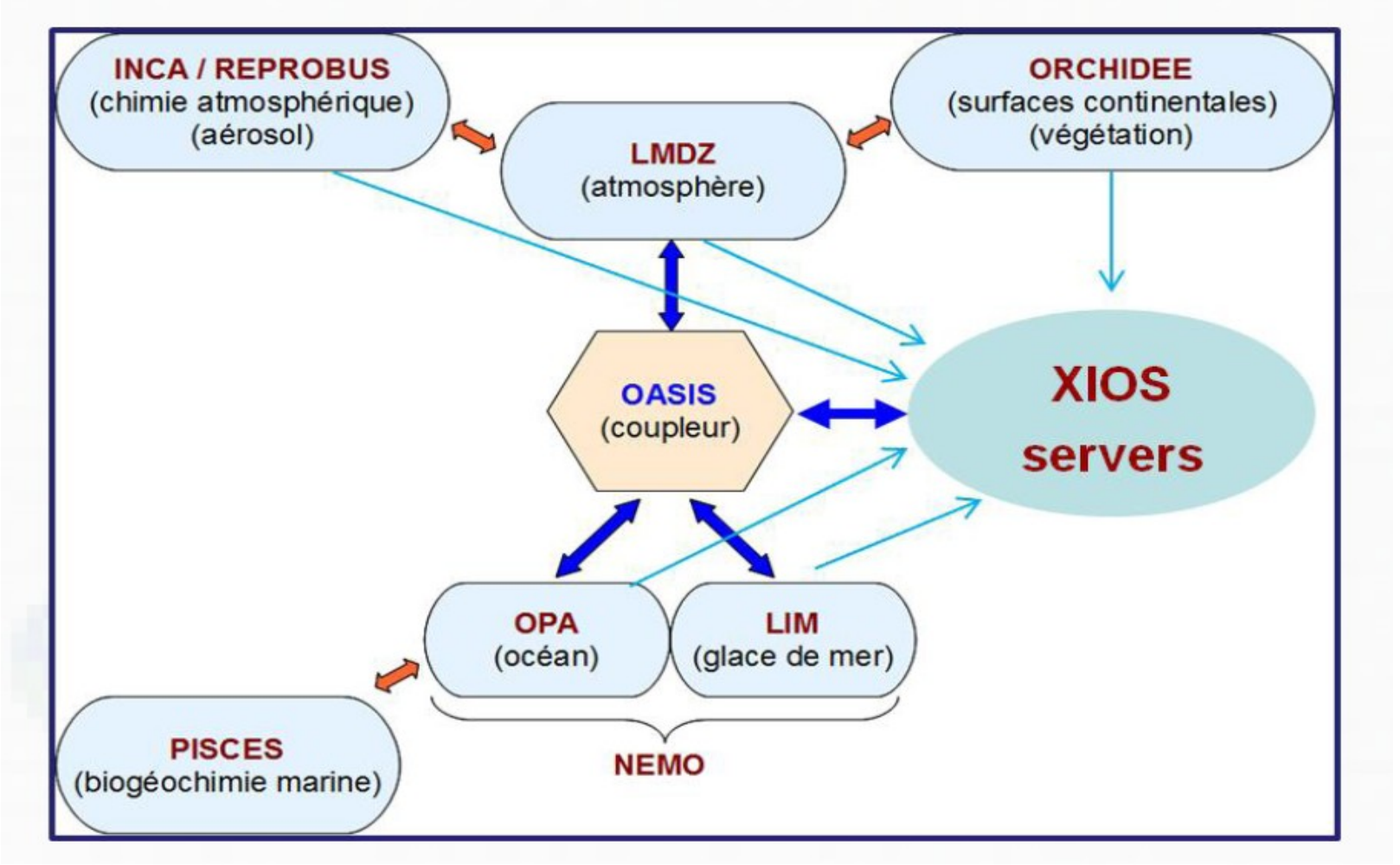
Attached mode or with server

- **Server mode:**
  compile and link with XIOS library
  Two executables: orchidee_ol or gcm.e + xios_server.exe
  XML : using_server=true
  file_definition type="multiple_file" => rebuild needed if more than one server
  file_definition type="one_file"
- Advantage: Better performances



Attached mode or with server

- **IPSLCM6 with XIOS server:**



Presentation plan

- Introduction to XIOS
- Implementation in ORCHIDEE
- Structure of XML parameter files
- Add new variables in the code and using xml
- Control output
- Compile and install
- Standard use with libIGCM configurations

Implementation in ORCHIDEE

- Implementation in ORCHIDEE done by Arnaud Caubel and Josefine Ghattas, introduced in ORCHIDEE **trunk revision 1788**, June 2014
- Results validated against IOIPSL output at curie.
- Running at **curie**/TGCC, **ada**/IDRIS and **obelix**/LSCE
- To be used with libIGCM configurations:
  ORCHIDEE_trunk, **LMDZOR_v6** and **IPSLCM6_rc0**

- In ORCHIDEE:
  - src_parallel/**xios_orchidee.f90** : One module doing all interfacing to XIOS
  - **src_xml** : new directory in ORCHIDEE containing xml files for running with XIOS
  - New parameter **XIOS_ORCHIDEE_OK** in run.def to activate running with XIOS

xios_orchidee.f90

```
! ================================================================================================================================
!  MODULE       : xios_orchidee
!
!  CONTACT      : orchidee-help _at_ ipsl.jussieu.fr
!
!  LICENCE      : IPSL (2006)
!  This software is governed by the CeCILL licence see ORCHIDEE/ORCHIDEE_CeCILL.LIC
!
!>\BRIEF   This module contains the initialization and interface to the XIOS code.
!!
!!\n DESCRIPTION: This module contains the interface for the use of the XIOS code. All call to XIOS is done in this module.
!!
!!                Summury of subroutines
!!                      xios_orchidee_comm_init       : First call to XIOS to get the MPI communicator
!!                      xios_orchidee_init            : Initialize variables needed for use of XIOS
!!                                                      Deactivation of fields not calculated due specific run options
!!                      xios_orchidee_update_calendar : Update the calandar in XIOS
!!                      xios_orchidee_finalize        : Last call to XIOS for finalization
!!                      xios_orchidee_send_field      : Interface to send fields with 1, 2 or 3 dimensions to XIOS
!!                      xios_orchidee_send_field_r1d  : Internal subroutine for 1D(array) fields
!!                      xios_orchidee_send_field_r2d  : Internal subroutine for 2D fields
!!                      xios_orchidee_send_field_r3d  : Internal subroutine for 3D fields
!!
!!                Note that compilation must be done with the preporcessing key XIOS and CPP_PARA. Compiling without these
!!                keys makes it impossible to activate XIOS. To activate run using XIOS, the flag XIOS_ORCHIDEE_OK=y must
!!                be set in run.def and the file iodef.xml must exist.
!!
!! RECENT CHANGE(S): Created by Arnaud Caubel(LSCE), Josefine Ghattas (IPSL) 2013
!!
!! REFERENCE(S) : None
!!
!! SVN          :
!! $HeadURL: $
!! $Date: $
!! $Revision: $
!! \n
! ================================================================================================================================
```

xios_orchidee.f90

This module contains the interface for the use of the XIOS code.
All call to XIOS is done in this module. All call to XIOS are protected by cpp key XIOS to make compiling without XIOS possible.

Public subroutines:

- **xios_orchidee_comm_init**
  First call to XIOS to get the MPI communicator instead of MPI_COMM_WORKD. Subroutine is called from init_orchidee_mpi, only in offline mode.

- **xios_orchidee_init**
  Initialize variables needed for use of XIOS. Define horizontal domain and axes. Deactivation of fields not calculated due specific run options. Subroutine is called from intersurf_initialize.

- **xios_orchidee_update_calendar**
  Update the time step in XIOS, called at each time step from intersurf_main.

- **xios_orchidee_finalize**
  Last call to XIOS for finalization.

xios_orchidee.f90

Public interface:
xios_orchidee_send_field

- Interface to send a field to XIOS.
- The field should be at landpoint compressed 1D grid and can have one (or two) extra dimensions
- To be called at each time step the variable is calculated
- Can be called from all modules in ORCHIDEE

The interface including the following private subroutines:

- xios_orchidee_send_field_r1d 1D(array) fields
- xios_orchidee_send_field_r2d 2D fields
- (xios_orchidee_send_field_r3d 3D fields, not yet to be used)

xios_orchidee_send_field

```
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)
```

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 parmeter 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 one suplementary axis:
REAL(r_std), DIMENSION(kjpindex) or
REAL(r_std), DIMENSION(kjpindex,:)

Specific case in xios_orchidee_init

In subroutine xios_orchidee_init:

```
!
!! 6. Deactivation of some fields if they are not calculated
!
IF ( .NOT. river_routing ) THEN
   CALL xios_set_field_attr("basinmap",enabled=.FALSE.)
   CALL xios_set_field_attr("nbrivers",enabled=.FALSE.)
   CALL xios_set_field_attr("riversret",enabled=.FALSE.)
   CALL xios_set_field_attr("hydrographs",enabled=.FALSE.)
   CALL xios_set_field_attr("fastr",enabled=.FALSE.)
   CALL xios_set_field_attr("slowr",enabled=.FALSE.)
   CALL xios_set_field_attr("streamr",enabled=.FALSE.)
   CALL xios_set_field_attr("lakevol",enabled=.FALSE.)
   CALL xios_set_field_attr("pondr",enabled=.FALSE.)
END IF


IF (hydrol_cwrr ) THEN
   CALL xios_set_field_attr("dss",enabled=.FALSE.)
   CALL xios_set_field_attr("gqsb",enabled=.FALSE.)
   CALL xios_set_field_attr("bqsb",enabled=.FALSE.)
   ...
```

Done to avoid variables to be written in output files if they are not calculated for a specific option. The same .xml files can therefore be used.

This is not done for stomate variables. If stomate is deactivated, the stomate file should be deactivated in file_def_orchidee.xml. Otherwise the variables will be declared but never written.

xml parameter files

To run ORCHIDEE with XIOS all diagnostic output files are configured through xml files. Following 4 files need to be present at each execution :

- iodef.xml — Main input file for XIOS
- context_orchidee.xml — Axis and domain information, include field and file def
- **field_def_orchidee.xml** — **=> Definition for each variable send in ORCHIDEE**
  => Only change if added new varible in ORCHIDEE
- **file_def_orchidee.xml** — **=> Specify all output files and there variables**
  => Change to set your output level
  => Remove variables, change levels, change freq...

And in run.def : XIOS_ORCHIDEE_OK=y

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

1- iodef.xml

```
<?xml version="1.0"?>
<!-- ============================================================================================================ -->
<!-- iodef.xml : Main configuration file for production of output files using XIOS                                -->
<!--             A seperatate file context_orchidee.xml contains all specifications for ORCHIDEE                 -->
<!-- ============================================================================================================ -->

<simulation>

  <!-- ============================================================================================================ -->
  <!-- XIOS context                                                                                               -->
  <!-- ============================================================================================================ -->
  <context id="xios">
    <variable_definition>
      <variable_group id="buffer">
            buffer_size = 80000000
            buffer_server_factor_size = 2
         </variable_group>
      <variable_group id="parameters">
        <variable id="using_server" type="boolean">false</variable>
        <variable id="info_level" type="int">0</variable>
      </variable_group>
    </variable_definition>
  </context>

  <!-- ============================================================================================================ -->
  <!-- ORCHIDEE context                                                                                           -->
  <!-- The file context_orchidee.xml is included here. This file needs to exist during run time.                  -->
  <!-- ============================================================================================================ -->
  <context id="orchidee" src="./context_orchidee.xml"/>

</simulation>
```

1- iodef.xml

```
<?xml version="1.0"?>
<!-- ============================================================================================================ -->
<!-- iodef.xml : Main configuration file for production of output files using XIOS                                -->
<!--             A seperatate file context_orchidee.xml contains all specifications for ORCHIDEE                 -->
<!-- ============================================================================================================ -->

<simulation>

  <!-- ============================================================================================================ -->
  <!-- XIOS context                                                                                               -->
  <!-- ============================================================================================================ -->
  <context id="xios">
    <variable_definition>
      <variable_group id="buffer">
            buffer_size = 80000000
            buffer_server_factor_size = 2
        </variable_group>
      <variable_group id="parameters">
        <variable id="using_server" type="boolean">false</variable>
        <variable id="info_level" type="int">0</variable>
      </variable_group>
    </variable_definition>
  </context>

  <!-- ============================================================================================================ -->
  <!-- ORCHIDEE context                                                                                           -->
  <!-- The file context_orchidee.xml is included here. This file needs to exist during run time.                  -->
  <!-- ============================================================================================================ -->
  <context id="orchidee" src="./context_orchidee.xml"/>

</simulation>
```

1- iodef.xml

```
<?xml version="1.0"?>
<!-- ============================================================================================================ -->
<!-- iodef.xml : Main configuration file for production of output files using XIOS                                -->
<!--             A seperatate file context_orchidee.xml contains all specifications for ORCHIDEE                 -->
<!-- ============================================================================================================ -->

<simulation>

  <!-- ============================================================================================================ -->
  <!-- XIOS context                                                                                               -->
  <!-- ============================================================================================================ -->
  <context id="xios">
    <variable_definition>
      <variable_group id="buffer">
            buffer_size = 80000000
            buffer_server_factor_size = 2
         </variable_group>
      <variable_group id="parameters">
        <variable id="using_server" type="boolean">false</variable>
        <variable id="info_level" type="int">0</variable>
      </variable_group>
    </variable_definition>
  </context>

  <!-- ============================================================================================================ -->
  <!-- ORCHIDEE context                                                                                           -->
  <!-- The file context_orchidee.xml is included here. This file needs to exist during run time.                  -->
  <!-- ============================================================================================================ -->
  <context id="orchidee" src="./context_orchidee.xml"/>

  <!-- ============================================================================================================ -->
  <!-- LMDZ context                                                                                               -->
  <!-- The file context_lmdz.xml is included here. This file needs to exist during run time.                      -->
  <!-- ============================================================================================================ -->
  <context id="LMDZ" src="./context_lmdz.xml"/>

</simulation>
```

2- context_orchidee.xml

```
<!-- ============================================================================================================== -->
<!-- ORCHIDEE context                                                                                               -->
<!-- context_orchidee.xml : Configuration file for ORCHIDEE for production of output files using XIOS              -->
<!-- ============================================================================================================== -->
<context id="orchidee">
  <!-- ============================================================================================================ -->
  <!-- Definition of all existing variables                                                                         -->
  <!-- DO NOT CHANGE THIS FILE                                                                                      -->
  <!-- ============================================================================================================ -->
  <field_definition src="./field_def_orchidee.xml"/>

  <!-- ============================================================================================================ -->
  <!-- Definition of output files                                                                                   -->
  <!-- Definition of variables or groups included in the different files                                            -->
  <!-- CHANGE THIS FILE BY ADDING THE FILES AND/OR VARIABLES YOU WANT TO PRODUCE                                     -->
  <!-- Only variables and groups existing in field_def_orchidee.xml can be used                                     -->
  <!-- ============================================================================================================ -->
  <file_definition src="./file_def_orchidee.xml"/>

  <!-- ============================================================================================================ -->
  <!-- Definition of horizontal domain                                                                              -->
  <!-- ============================================================================================================ -->
  <domain_definition>
    <domain id="domain_landpoints"/>
  </domain_definition>

  <!-- ============================================================================================================ -->
  <!-- Definition of vertical axis and extra dimensions                                                             -->
  <!-- ============================================================================================================ -->
  <axis_definition>
    <!-- Vertical axis and extra dimensions in sechiba -->
    <axis id="veget" standard_name="model_level_number" long_name="Vegetation types" unit="1"/>
    <axis id="laiax" standard_name="model_level_number" long_name="Nb LAI" unit="1"/>
    <axis id="solth" standard_name="model_level_number" long_name="Soil levels" unit="m"/>
    <axis id="soiltyp" standard_name="model_level_number" long_name="Soil types" unit="1"/>
    <axis id="nobio" standard_name="model_level_number" long_name="Other surface types" unit="1"/>
    <axis id="albtyp" standard_name="model_level_number" long_name="Albedo types" unit="1"/>
    <axis id="solay" standard_name="model_level_number" long_name="Hydrol soil levels" unit="m"/>
    <axis id="soildiag" standard_name="model_level_number" long_name="Diagnostic soil levels" unit="m"/>
    <axis id="snowlev" standard_name="model_level_number" long_name="Snow levels" unit="m"/>

    <!-- Vertical axis and extra dimensions in stomate -->
    <axis id="PFT" standard_name="model_level_number" long_name="Plant functional type" unit="1"/>
    <axis id="P10" standard_name="model_level_number" long_name="Pool 10 years" unit="1"/>
    <axis id="P100" standard_name="model_level_number" long_name="Pool 100 years" unit="1"/>
    <axis id="P11" standard_name="model_level_number" long_name="Pool 10 years + 1" unit="1"/>
    <axis id="P101" standard_name="model_level_number" long_name="Pool 100 years + 1" unit="1"/>
  </axis_definition>

</context>
```

2- context_orchidee.xml

```
<!-- ============================================================================================================== -->
<!-- ORCHIDEE context                                                                                               -->
<!-- context_orchidee.xml : Configuration file for ORCHIDEE for production of output files using XIOS              -->
<!-- ============================================================================================================== -->
<context id="orchidee">
  <!-- ============================================================================================================ -->
  <!-- Definition of all existing variables                                                                         -->
  <!-- DO NOT CHANGE THIS FILE                                                                                      -->
  <!-- ============================================
  <field_definition src="./field_def_orchidee.xml"

  <!-- ============================================
  <!-- Definition of output files
  <!-- Definition of variables or groups included in the different files                                            -->
  <!-- CHANGE THIS FILE BY ADDING THE FILES AND/OR VARIABLES YOU WANT TO PRODUCE                                     -->
  <!-- Only variables and groups existing in field_def_orchidee.xml can be used                                     -->
  <!-- ============================================================================================================ -->
  <file_definition src="./file_def_orchidee.xml"/>

  <!-- ============================================
  <!-- Definition of horizontal domain
  <!-- ============================================
  <domain_definition>
    <domain id="domain_landpoints"/>
  </domain_definition>

  <!-- ============================================================================================================ -->
  <!-- Definition of vertical axis and extra dimensions                                                             -->
  <!-- ============================================================================================================ -->
  <axis_definition>
    <!-- Vertical axis and extra dimensions in sechiba -->
    <axis id="veget" standard_name="model_level_number" long_name="Vegetation types" unit="1"/>
    <axis id="laiax" standard_name="model_level_number" long_name="Nb LAI" unit="1"/>
    <axis id="solth" standa
    <axis id="soiltyp" stan
    <axis id="nobio" standa
    <axis id="albtyp" stand
    <axis id="solay" standa
    <axis id="soildiag" sta
    <axis id="snowlev" stan

    <!-- Vertical axis and
    <axis id="PFT" standard_
    <axis id="P10" standard_name="model_level_number" long_name="Pool 10 years" unit="1"/>
    <axis id="P100" standard_name="model_level_number" long_name="Pool 100 years" unit="1"/>
    <axis id="P11" standard_name="model_level_number" long_name="Pool 10 years + 1" unit="1"/>
    <axis id="P101" standard_name="model_level_number" long_name="Pool 100 years + 1" unit="1"/>
  </axis_definition>

</context>
```

<field_definition src="./field_def_orchidee.xml"/>

<file_definition src="./file_def_orchidee.xml"/>

<axis_definition>
<axis id="veget" standard_name="model_level_number"
long_name="Vegetation tyes" unit="1"/>

3- field_def_orchidee.xml

```
<!-- ================================================================================================================== -->
<!-- field_def_orchidee.xml : Definition of all existing variables                                                      -->
<!-- This file must only be changed if a call xios_orchidee_send_field is added, changed or removed                     -->
<!-- ================================================================================================================== -->

<field_definition id="orchidee" prec="4" domain_ref="domain_landpoints" operation="average" freq_op="1ts" enabled=".TRUE." default_value="9.96921e+36">


  <!-- Definition of all variables in sechiba -->
  <field_group id="sechiba">
    <field id="mrsos" name="mrsos" le
    <field id="Areas" name="A
    <field id="LandPoi
    <field id="Co
    <field id
    <fiel
    <fi
    <
    <fie                                                                                                    ield>

    <field id="q
    <field id="albedo
    <field id="fluxsens" na
    <field id="fluxlat" name="flux
    <field id="emis" name="emis" long_name=
    <field id="rain" name="rain" long_name="Rainfall" unit=
    <field id="snowf" name="snowf" long_name="Snowfall" unit="mm/d"/>
    <field id="netrad" name="netrad" long_name="Net radiation" unit="W/m^2"/>
    <field id="lai" name="lai" long_name="Leaf Area Index" unit="1" axis_ref="veget"/>
    <field id="reinf_slope" name="reinf_slope" long_name="Slope index for each grid box" unit="1" operation="once"/>
    <fi
    <fi
    <fi
    <fi
```

Definition for each variable send in ORCHIDEE

- one line per variable

Only change if you added new variables in ORCHIDEE

Does not control output files

DO NOT REMOVE VARIABLES FROM HERE

This file is stored with the model source code in src_xml/ because it is closely related to the version of the code.

4- 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_r
    <field field_r
    <field field_r
    <field field_r
    <field field_r
    ...
  </file>

  <!-- Sechiba fil
  <file id="sechib
    <field field_r
    <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="CONTFRAC_STOMATE" level="1"/>
  </file>
</file_definition>
```

Information about all files written by ORCHIDEE

type “one_file” or “multiple_file” : XIOS will gather information from all processes on a single output file

enabled “.TRUE.” / “.FALSE.” : possiblity to deactivate all output files

4- 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="CONTFRAC_STOMATE" level="1"/>
  </file>
</file_definition>
```

4- 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>

  <!-- Sechi
  <file id="
    <field f
    <field f
    <field f
    <field f
    <field f
    ...
  </file>

  <!-- Stoma
  <file id="
    <field f
    <field f
    <field f
  </file>
</file_defin
```

Information line about one file

name	filename, suffix .nc will be added to the filename
output_level	**“x”** : all variables listed below with level less or equal to x will be added
output_freq	”1d”, “1800s”, “1ts”, “1mo”, “3h”, “1y” : frequency for the file
enabled	“.TRUE.” / “.FALSE.” : create the file, true is default
operation	can be added, overwrites settings in field_def “average”, “min”, “max”, “instant”

4- 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 id="se
    <field fie
    <field fie
    <field fie
    <field fie
    <field fie
    ...
  </file>

  <!-- Stomate
  <file id="st
    <field fie
    <field fie
    <field fie
  </file>
</file_definit
```

A line per variable added in the file

field_ref	reference id as set in field_def_orchidee.xml file
level	**“x”** : the variable is only written if this level is less or equal of output_level set at the file description line above.
name / long_name	”new_name” : name of the variable in the output file. If it is not set, the name set in field_def_orchidee.xml will be used.
enabled	“.TRUE.” / “.FALSE.” : write the variable, true is the default.
operation	can be added, overwrites settings in field_def “average”, “min”, “max”, “instant”

Add a new variable in ORCHIDEE

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

```
CALL xios_orchidee_send_field("newid",new_var)
```

2) In field_def_orchidee.xml, add declaration of the variable

3) In file_def_orchidee.xml : add the variable in all files 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 beeing written if you keep the same .xml files.

Create new variable from existing in field_def_orchidee.xml

=> Possibility to add operation: maximum, minimum, once, accumulate

=> Possibility to create new variables from an existing variable, using attribute field_ref

Example:
The variable with id=temp_sol_C is send in ORCHIDEE. Using this variable as reference, 2 new variables are defined in field_def_orchidee.xml.

<field **id="temp_sol_C"** name="temp_sol" long_name="New Surface Temperature" unit="C"/>

<field **id="tsol_max"** name="tsol_max" **field_ref="temp_sol_C"** long_name="Maximum Surface Temperature" unit="C" **operation="maximum"**/>

<field **id="tsol_min"** name="tsol_min" **field_ref="temp_sol_C"** long_name="Minimum Surface Temperature" unit="C" **operation="minimum"**/>

Create new variable from existing in field_def_orchidee.xml

=> Possibility to add or extract a scalar to a variable

Example: temperatures in Kelvin and/or Celsius

Currently we send the surface temperature both in Kelvin and Celsius, in src_sechiba/intersurf.f90:

CALL xios_orchidee_send_field("temp_sol_K",ztemp_sol_new)

~~CALL xios_orchidee_send_field("temp_sol_C",ztemp_sol_new-ZeroCelsius)~~

But we can define the temperature in Celsius directly in field_def_orchidee.xml:

<field id="temp_sol_K" name="AvgSurfT" long_name="Average surface temperature" unit="K"/>

<field id="temp_sol_C" name="temp_sol" **field_ref="temp_sol_K"** long_name="New Surface Temperature" unit="C"**> temp_sol_K - 273.15 <**/field>

Control output

How can I change the name for a variable?

- In file_def_orchidee.xml to change only for one specific file or in filed_def_orchidee.xml if you want to change in all output files

How can I change the long_name for a variable?

- As for the variable name, see above

How do I know if a variable is averaged, instant, min or max?

- See field_def_orchidee.xml. The defalut is average. Some variables are set to min, max or once. No variables are currently set to instant.

How can I write instant variables?

- Option 1) set output_freq=1ts in file_def_orchidee.xml for one file. You'll then have output at each time step.
- Option 2) set operation=instant on the file description line, in file_def_orchidee.xml.
    - For example operation=”instant” + output_freq=”1d”, once a day the instant variables will be written.

Control output

How can I change the frequency of an output file?

- Change output_freq on the line description for the file

How can I change the level for only one variable?

- Change the level for the variable in file_def_orchidee.xml

How can I create a new output file?

- Open file_def_orchidee.xml and add a new file section.

Why is the variable cimean set to enabel=”FALSE” in field_def_orchidee.xml?

- This variable is currently not correct in ORCHIDEE. In some cases it contains NAN. Thererfor this variable is deactivated from all files.

Control output

How can I change to alma output?

- Alma output are prepared in file_def_orchidee.xml but deactivated as default
  - You need to change enable=”TRUE” on the corresponding file description lines in file_def_orchidee.xml
  - No need to change in run.def

In ORCHIDEE source code

- If only the name changes between an alma and “no alma” variable, then the name is changed in file_def_orchidee.xml

- If the unit changes, both variables are send from ORCHIDEE with different names. For example in hydrol_main:

```
CALL xios_orchidee_send_field("snowf",precip_snow)
CALL xios_orchidee_send_field("snowf_alma",precip_snow/dt_sechiba)
```

- If one of the variables RootMoist, DelSoilMoist, DelIntercept, DelSWE or SoilWet are activated in file_def_orchidee.xml, then the variable almaoutput is set to true in ORCHIDEE. This variable activates some specific calculations needed for these variables.

Compilation

- XIOS must be compiled before ORCHIDEE
  - done at ada(IDRIS), curie(TGCC) and obelix(LSCE)

- The preprocessing key **`XIOS`** must be activated when compiling ORCHIDEE:
  - Use **makeorchidee_fcm** with argument **-xios**, this argument
    - activates cpp key XIOS
    - links to xios library

- Configuration ORCHIDEE_trunk
  - extraction of XIOS is always included
  - the main makefile compiles XIOS and ORCHIDEE if using ***with_xios***:

```
cd modipsl/config/ORCHIDEE_OL
gmake with_xios
```

- Configuration LMDZOR_v6 and IPSLCM6: compiling with XIOS is default

Install & compile

Install ORCHIDEE for offline use

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

cd modipsl/util
./model ORCHIDEE_trunk

cd ../config/ORCHIDEE_OL
gmake with_xios
```

After compiling you'll have 2 executables in modipsl/bin:
- xios_server.exe
- orchidee_ol

=> You can launch orchidee_ol only(attached mode), or together with xios_server.exe(server mode)
=> You can use XIOS or IOIPSL or both, use XIOS_ORCHIDEE_OK=y/n

Install & compile

Install ORCHIDEE for couled use with LMDZ

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

cd modipsl/util
./model LMDZOR_v6

cd ../config/LMDZOR_v6
gmake [resol]
```

After compiling you'll have 2 executables in modipsl/bin:

- xios_server.exe
- gcm.e

=> You can launch gcm.e only(attached mode), or together with xios_server.exe(server mode)
=> Server mode is the default
=> You can not switch off XIOS

Running in attached mode

Requirements for running ORCHIDEE with XIOS in attached mode:

- 1 executable: **orchidee_ol**
- 4 xml files : iodef.xml, context_orchidee.xml,
  field_def_orchidee.xml, file_def_orchidee.xml
- Parameter file: run.def
- Input files as usual: forcing_file.nc, PFTmap.nc, ...

Change in iodef.xml:

<variable id="**using_server**" type="boolean">**false**</variable>

Set in run.def:

```
XIOS_ORCHIDEE_OK=y          # Activate XIOS
WRITE_STEP=0                # Deactivate sechiba IOIPSL output
```

It it possible to run in sequential mode

Note: You can copy xml files from ORCHIDEE/src_xml

Running with server

Requirements for running ORCHIDEE with XIOS using server:

- 2 executables: **orchidee_ol** and **xios_server.exe**
- 4 xml files : iodef.xml, context_orchidee.xml,
  field_def_orchidee.xml, file_def_orchidee.xml
- Parameter file: run.def
- Input files as usual: forcing_file.nc, PFTmap.nc, ...

Change in iodef.xml:

<variable id="**using_server**" type="boolean">**true**</variable>

Set in run.def:

```
XIOS_ORCHIDEE_OK=y          # Activate XIOS
WRITE_STEP=0                # Deactivate all IOIPSL output
```

Note: You can copy xml files from ORCHIDEE/src_xml

Using libIGCM configurations
ORCHIDEE_trunk

- Running with XIOS can be activated in the experiments
    - OOL_SEC_STO
    - OOL_SEC
    - SPINUP_ANALYTIC

- In COMP/orchidee_ol.card, in UserChoices section, set XIOS=y. IOIPSL output will be deactivated by orchidee_ol.driver

- The copy of xml files are already done in section ParameterFiles in orchidee_ol.card.

- By default running will be done in attached mode.

Using libIGCM configurations
ORCHIDEE_trunk – server mode

config.card:
- Add component IOS
- Set number of cores MPI for each executables.

- *see example done in OOL_SEC_STO/config.card.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=
```

For several executables set here the number of cores MPI

Using libIGCM configurations
LMDZOR_v6

- Compiling and running with XIOS is default
- (Running without XIOS needs recompilation)
- Running with XIOS server is default

Using libIGCM configurations
control of output

- The control of output are done as before : only output level is changed and some files can be switched on and off
- In ORCHIDEE_OL, file_def_orchidee.xml file is stored in PARAM/ directory for the configuration
- Other xml files are copied from the source directory src_xml/
- iodef.xml is stored in PARAM/
- The driver will add context_orchidee.xml in iodef.xml if XIOS=y

Currently controlled by:

- WriteFrequency in config.card

But also: Recommended method
change directly in OOL_SEC_STO/PARAM/file_def_orchidee.xml:

```
<!-- Sechiba file 1 -->
<file id="sechiba1" name="sechiba_history" output_level="11" output_freq="_AUTO_" enabled="_AUTO_">
  <!-- level 1 -->
  <field field_ref="Areas" level="1"/>
  <field field_ref="LandPoints" level="1"/>
  <field field_ref="Contfrac" level="1"/>
  <field field_ref="evap" level="1"/>
```

Installing at a new platform

- Currently ORCHIDEE with XIOS has only been tested at obelix, curie and ada
- Requirements are MPI and netCDF4 library
- Additional requirements: parallel library NetCDF4/HDF5
  - several processes (XIOS clients or servers) can write into one single output file

Steps to follow for installation at a new platform:

1. Install configuration ORCHIDEE_trunk in a new modipsl

2. Modify compile options in following files:
   - modipsl/util/**AA_make.gdef**
   - modipsl/modeles/**ORCHIDEE/arch/arch-*yourtarget*.[fcm/path]**
   - modipsl/modeles/**XIOS/arch/arch/arch-*yourtarget*.[fcm/path/env]**

   Note: the variable FCM_ARCH in AA_make.gdef is the name of the arch files in ORCHIDEE/arch and XIOS/arch.

4. Recreate makefiles with target chosen above and compile as usual
   cd modipsl/util; ./ins_make -t yourtarget