

# Joint usage of OASIS3-MCT and XIOS in climate models

Arnaud Caubel (LSCE/IPSL), February 2022

Both OASIS3-MCT (CERFACS) and XIOS (LSCE/IPSL) are widely used in the climate modeling community, respectively to handle coupling and Input/Output in numerical models. Since the exchange protocol of these tools are based on messages within MPI communicators which can be shared by the tools, some good practices are useful to use conjointly OASIS3-MCT and XIOS within one same model (Fig 1.)

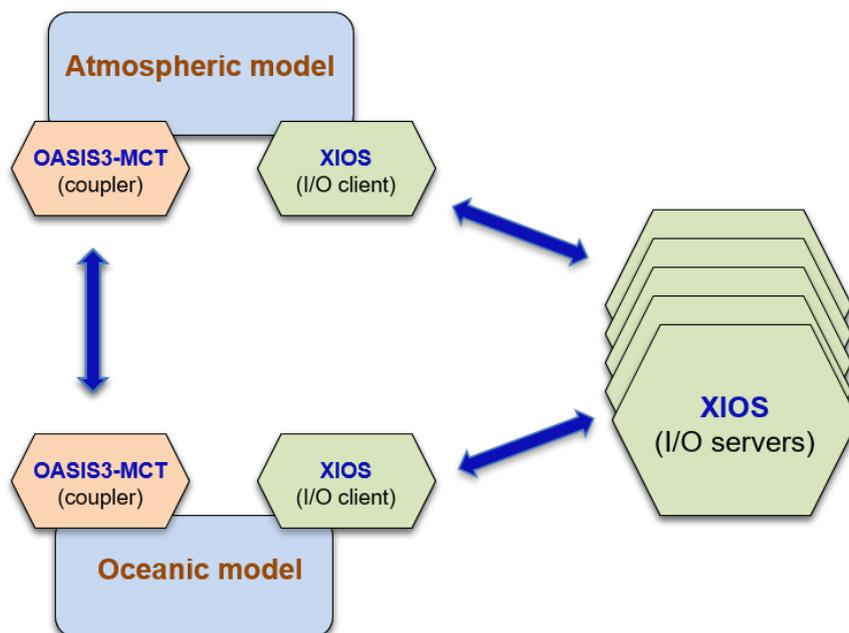


Fig 1: Coupled model using OASIS3-MCT as coupler and XIOS as I/O server

Note that following information is related to XIOS 2.5 or trunk version and OASIS3-MCT\_2.0 or later.

## 1. Interfacing with OASIS3-MCT and XIOS within a climate model

### a. Initialisation phase

From the OASIS3-MCT and XIOS point of view, the objective of this step is both to initialize the MPI library (calling *MPI\_Init*), if not already initialized by the model, and to create a local communicator to be used by the model for its internal parallelization (Fig 2). Both OASIS3-MCT and XIOS fulfill these functions thanks to *oasis\_init\_comp* and *oasis\_get\_localcomm* API routines for OASIS3-MCT and *xios\_initialize* (that calls the OASIS3-MCT API routines) for XIOS. In order to avoid duplicating tasks, the call to these API routines must be ordered in the component as proposed in the two following examples:

```
CALL oasis_init_comp( comp_id, "ocean", status)
CALL oasis_get_localcomm( ilocal_comm, status )           (1)
CALL xios_initialize( "ocean", local_comm=ilocal_comm , return_comm=ilocal_comm)
```

or

```
CALL xios_initialize( "ocean", return_comm=ilocal_comm)   (2)
```

In case (1), the call to OASIS3-MCT initialization API routines allows to initialize the MPI library (calling *MPI\_Init* if not already initialized by the model) and to get the value *ilocal\_comm* of a local communicator handle. Then, the call to *xios\_initialize* returns *return\_comm* the handle of a communicator that is in fact a duplication of the *local\_comm* communicator.

In case (2), XIOS handles the initialization of OASIS3-MCT (and MPI if needed) by calling internally *oasis\_init\_comp* and *oasis\_get\_localcomm* API routines and returning the split local communicator handle as *return\_comm* argument.

In both cases (1) and (2), the call to *xios\_initialize* allows XIOS to pick up an inter-communicator between the model component (XIOS client) and XIOS servers by calling the OASIS3-MCT *oasis\_get\_intercomm* API routine: this inter-communicator is needed by XIOS to create a communicator and achieve communications between model component processes and XIOS servers (Fig 2).

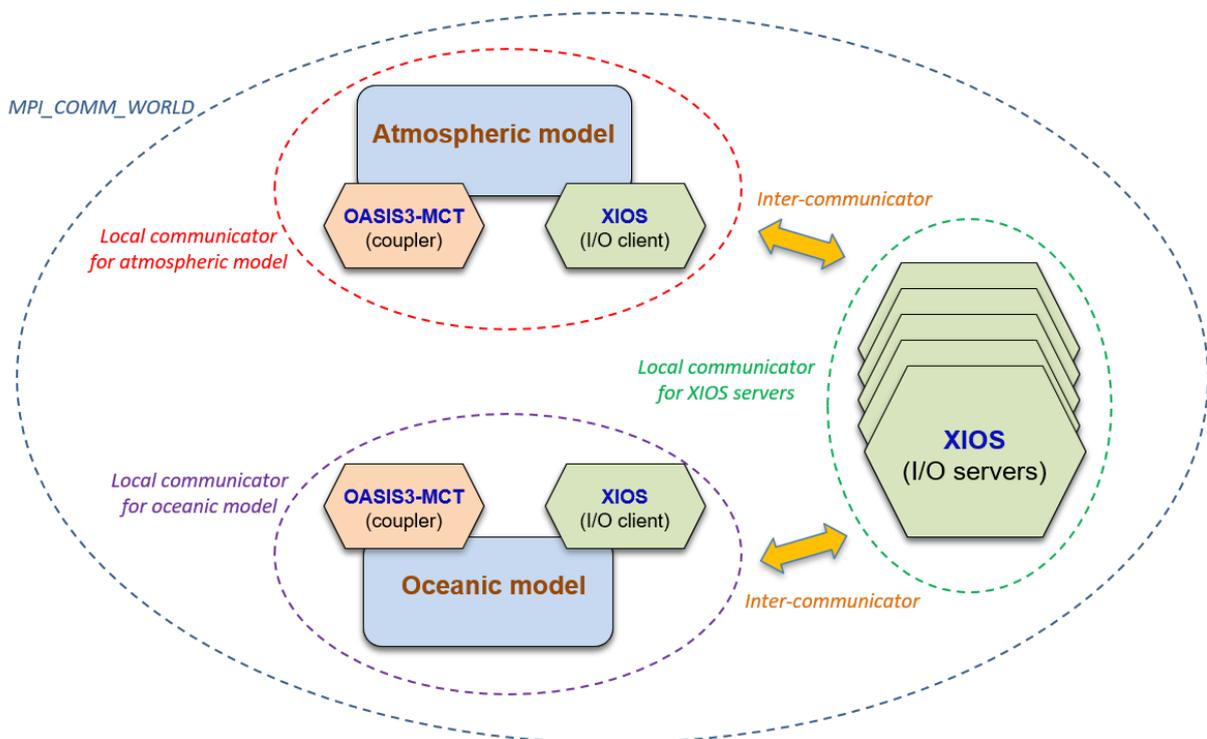


Fig 2: Handling of MPI communicators in a coupled model using OASIS3-MCT as coupler and XIOS as I/O server.

## b. Definition phase

In the model component, the OASIS3-MCT definition phase is finalised by calling the *oasis\_enddef* API routine. This routine includes global collective MPI communications (using the MPI\_COMM\_WORLD communicator). The XIOS *xios\_context\_initialize* subroutine initializes one or several XIOS context(s) and also includes collective MPI communications. Hence, deadlocks may occur if the developer does not pay any attention to the calling order of the two subroutines. The calling rule depends on the version of XIOS, as follows:

- XIOS version branch 2.5 or trunk (before revision 1587)

In order to avoid deadlocks, it is mandatory to call first *oasis\_enddef* and then *xios\_context\_initialize* in all the components involved in the coupling, to be coherent with the calling order in XIOS servers. A calling sequence in your models could be as follows:

```
CALL oasis_enddef
...
CALL xios_context_initialize (xios_context, local_comm)
```

- XIOS version trunk (revision 1587 or later)

An XIOS functionality was developed to explicitly inform XIOS servers they have to call *oasis\_enddef* function. This function, *xios\_oasis\_enddef*, also allows to solve deadlocks encountered in case of bad call order of *oasis\_enddef* and *xios\_context\_initialize* in the model. In all cases, however, the call to *xios\_oasis\_enddef* needs to be implemented before the call to *oasis\_enddef* on model side, so the two following implementations are allowed:

```
...
CALL xios_oasis_enddef
CALL oasis_enddef
...
CALL xios_context_initialize (xios_context, local_comm)
...
```

or

```
...
CALL xios_context_initialize (xios_context, local_comm)
...
CALL xios_oasis_enddef
CALL oasis_enddef
...
```

The XIOS parameter *call\_oasis\_enddef* (default value is *true*) must be set in *iodef.xml* file:

- *true*: *xios\_oasis\_enddef* function must be called in every component involved in the coupling.
- *false*: the functionality is deactivated and the behavior of XIOS version branch 2.5 or trunk < revision 1587 is retrieved, i.e. *xios\_oasis\_enddef* function does not need to be called in the models but deadlocks may happen in case of bad call order of *oasis\_enddef* and *xios\_context\_initialize*.

### c. Finalisation phase

The finalization sequence must be similar to the initialization one. If OASIS3-MCT has initialized MPI (*oasis\_init\_comp*), the MPI finalization will be done by OASIS3-MCT in the *oasis\_terminate* API routine, either called directly by the model (case (3)) or within *xios\_finalize* (case (4)).

<i>CALL xios_finalize</i> <i>CALL oasis_terminate</i>	(3)
--	-----

or

<i>CALL xios_finalize</i>	(4)
---------------------------	-----

## 2. Compilation parameters

### XIOS

Compilation with *--use\_oasis* is mandatory to use OASIS3-MCT API routines in XIOS (XIOS C-fortran interface of OASIS3-MCT API routines is defined only if *--use\_oasis* is specified).

<i>./make_xios --use_oasis...</i>
-----------------------------------

### OASIS3-MCT

Nothing to do at compilation step.

## 3. Runtime parameters

### XIOS *iodef.xml*

Some OASIS3-MCT parameters have to be enabled in *iodef.xml*:

- *using\_oasis*: *true* if OASIS3-MCT is used as coupler. Default value: *false*.
- *call\_oasis\_enddef* : see part 1.b; default value is *true*.
- *oasis\_code\_id*: identifiers of model component using both OASIS3-MCT and XIOS (these identifiers must be the same as the one used as argument of *oasis\_init\_comp* API routine). Only used if XIOS server mode is enabled.

```
<variable id="using_oasis"      type="bool">true</variable>
<variable id="oasis_codes_id"  type="string">atmos,ocean</variable>
```

### OASIS3-MCT namcouple

With OASIS3-MCT version prior to 3.0, model identifiers defined in XIOS *iodef.xml* must be the same as the one defined in the OASIS3-MCT *namcouple* configuration file and the one defined as argument of *oasis\_init\_comp* API routine. *xios.x* is the identifier to be used for XIOS component in *namcouple* file, as follows:

```
$NBMODEL
3 atmos ocean xios.x
```

We note however that since version 3.0 of OASIS3-MCT, this *namcouple* parameter is not used anymore.