

Laure Coquart and Jean-Marie Epitalon

OASIS4-GUI User Guide

Technical Report, TR/CMGC/10/36

SUC at CERFACS, URA CERFACS/CNRS No1875 France

OASIS4-GUI User Guide

Edited by
L. Coquart (CNRS-CERFACS), J.- M. Epitalon (Consultant)
May 2010

1.	Introduction.....	3
2.	Installing the libraries	4
2.1.	Automatic installation of the libraries.....	4
2.2.	Manual installation of the libraries	5
2.3.	Troubleshooting	5
3.	General characteristics of the User Interfaces	6
4.	Running wizard.tcl to create AD and PMIOD files	7
4.1.	Characteristics of the application.....	7
4.2.	Characteristics of each component included in the application.....	9
4.3.	Characteristics of each field of each component	10
4.4.	Saving the configuration files	12
5.	Running oasis-gui.tcl to create SCC and SMIOC files.....	13
5.1.	Description of the global graphical interface.....	13
5.2.	Description of the buttons bar.....	15
5.3.	Loading the AD and PMIOD files.....	15
5.4.	Attributes of the applications and components.....	17
5.5.	Attributes of the components	19
5.6.	Input NetCDF files.....	20
5.7.	Output NetCDF files	21
5.8.	Coupling fields.....	24
5.9.	Experiments Settings	29
5.10.	Saving and/or loading a coupling configuration	30
6.	The mode standalone	32
7.	Acknowledgement	36

1. Introduction

The OASIS4 Graphical User Interface helps OASIS4 user in creating the AD (Application Description), PMIOD (Potential Model Input and Output Description), SCC (Specific Coupling Configuration) and SMIOC (Specific Model Input and Output Configuration) XML files needed by OASIS4 to run a coupled model.

These files are described in details in the OASIS4_User_Guide.pdf that you can find in the directory oasis4/doc in the sources of OASIS4.

The OASIS4-GUI is based on Tclxml3.2. It is intended only for Linux/Unix systems.

The utility sources are divided into four directories:

- a - gui : main directory containing four Tcl/Tk programs (get-version.tcl, install.tcl, wizard.tcl, oasis-gui.tcl) and the other directories
- b - gui/package-source : directory with the libraries needed to run correctly the utility
- c - gui/IMAGES : directory with the bimages used by the graphical interface oasis-gui.tcl
- d - gui/examples : directory with the examples used to present and illustrate the document

In the main directory, the four programs are written in Tcl/Tk version 8.4. If you run the program ./get-version.tcl you get the version of Tcl/Tk installed on your computer. The OASIS4-GUI interface requires version 8.4, or greater.

The program wizard.tcl is used to create the AD and the PMIOD XML files, which will be necessary to run the second program oasis-gui.tcl. The program oasis-gui.tcl is used to generate the SCC and the SMIOC from the AD and PMIOD XML files, as represented on **Figure 1-1**:

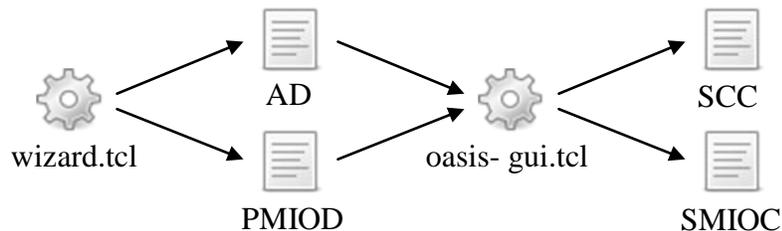


Figure 1-1: Functioning of the OASIS4-GUI

The program `install.tcl` installs automatically the libraries needed by the two programs `wizard.tcl` and `oasis-gui.tcl`, as described below in the **section 2.1**.

The directory `package-source` contains the source files of the Tcl packages needed for the execution. These are written in C and Tcl/Tk.

The directory `Examples` contains the AD, PMIOD, SCC and SMIOC XML files that were created with `wizard.tcl` and `oasis-gui.tcl` to present the interfaces below.

The first part of this document describes how to install on the computer the libraries contained in the directory `package-source`. The second part is dedicated to the description of the `wizard.tcl` interface while the third part describes the `oasis-gui.tcl` program. Finally we describe how to create the configuration files for a stand alone application (see the OASIS4 User Guide for more details).

2. Installing the libraries

The `libxml2`, `libiconv` and `libxslt` are mandatory to run the interface OASIS4-GUI. They are not provided with this package but you can download separately `libxml2` and `libxslt` from `xmlsoft.org` ([\[http://xmlsoft.org/\]](http://xmlsoft.org/)) and `libiconv` from a GNU mirror site. On Linux RedHat or Fedora, you can check for `libxml2` invoquing the command `'rpm -q'`.

If Tcl/Tk librairies, and then the `tclConfig.sh` configuration file, are located in standard location (as `/usr/local/lib` or `usr/lib`), you should be able to install automatically the libraries. Else you will have to manually install them (see the **section 2.2**).

2.1. Automatic installation of the libraries

To install automatically, you must type the following commands:

```
chmod u+x install.tcl  
./install.tcl
```

At the end of the installation, a new directory **`$(directory_of_mine)/oasis4/util/gui/packages/lib`** is created containing the different libraries needed by the interface.

There are the two directories in it: `tcllib-1.10` and `Tclxml3.2` and the file `TclxmlConfig.sh`.

In some non standard cases, for example if Tcl/Tk or `libxml2` is not in the default location, the automatic installation will not work on your system. Therefore you may have to manually install the libraries.

2.2. Manual installation of the libraries

If you did not succeed to install automatically the libraries, you can install them manually, remembering that they must be located in the directory `$(directory_of_mine)/oasis4/util/gui/packages/lib`. Else they will not be found by the programs.

The installation of TclLib is done thanks to the script `installer.tcl` with the options given below:

Installing TclLib:

```
- cd package-source/tcllib-1.10
  ./installer.tcl -no-wait -no-gui -no-html -no-nroff -no-examples -pkg-path
  ".../packages/lib/tcllib-1.10" -no-apps
```

Tclxml uses the configure script. You must check the appropriate options for your system:

Use the `--exec-prefix` and `--prefix` to specify the destination directory (lib is created automatically under this directory)

Use the `--enable-threads` switch if Tcl has been built with threads enabled

Use the `--with-tcl` switch if Tcl configuration file `tclConfig.sh` is not in a standard location

Use the `--with-xml2-config` and `--with-libxslt-config` switches to specify where are located the configuration files `xml2-config` and `libxslt-config` of `libxml2` and `libxslt` libraries if they are not in a standard location.

For example, on a 64 bits Linux machine, Tcl/Tk 8.5 and `libxml2` are not in the standard paths `/usr/local/lib` and `/usr/lib`. They are located in `/usr/lib64`, so it is not possible to use the automatic installation.

The different commands to install the source packages on such kind of machine are given below. Be careful that TclLib must be installed before Tclxml. The directory `lib` is automatically created under the directory `packages`.

Installing Tclxml:

```
- cd ../tclxml-3.2
  ./configure --prefix=$(directory_of_mine)/oasis4/util/gui/packages
  --exec-prefix=$(directory_of_mine)/oasis4/util/gui/packages
  make
  make install
```

2.3. Troubleshooting

One problem commonly found on some systems is the absence of the `tclConfig.sh` file which results in the following error message:

```
configure: WARNING: Can't find Tcl configuration definitions
```

The file in question is the configuration parameters of Tcl as installed on the system.

Usually, the simplest way to get it is to download the linux package tcl-devel from the Internet. This package contains tclConfig.sh.

tcl-devel package comes in different releases as does tcl package. Please, first check the version of Tcl you have (`./get_version.tcl`) and then search the Internet for the corresponding tcl-devel package version.

Be also careful that each Linux distribution has its own package format. Some of them are becoming standard as the following:

- RPM package : originally from RedHat, it is used on Fedora, OpenSuse and Mandriva
- Debian package : originally from Debian, it is used on Debian and Ubuntu

If nothing of the above works for you, you can use the sample-tclConfig.h provided in this package in the directory package-source.

To install everything in the standard locations, you must be root, or ask an administrator to do that for you.

Then, depending on the location where you put this file, you may install the libraries automatically or manually using the option of configure:
--with-tcl=PATH.

If you did not succeed to install the libraries neither automatically nor manually, please contact Laure Coquart at CERFACS: coquart@cerfacs.fr

3. General characteristics of the User Interfaces

Online help is available for both wizard.tcl and oasis-gui.tcl programs. A box is displayed on the screen for each input required from the user.

For experienced users, these online help may be deactivated by using the option **-silent** on the command line when launching the programs (see the **sections 4 and 5**).

On every form presented to the user, mandatory information is marked in red.

We describe and explain the two programs wizard.tcl and oasis-gui.tcl through the creation of a coupling configuration step by step. An application named "source" with one component "ocn" is coupled to an application named "target", with one component "atm". The component ocn reads data from an existing NetCDF file, while the component atm writes data in a NetCDF file. The component ocn also send a coupling field to the component atm. The

application source can run on 1 or 3 processors while the application target can run on 1, 2 or 3 processors.

In the last **section 6**, we finally present how to create the configuration files for the application source with the component ocn that will run in the stand alone mode (see the OASIS4 User Guide for details).

4. Running wizard.tcl to create AD and PMIOD files

The interface wizard.tcl must be called before invoking the oasis-gui.tcl to create the AD and PMIOD XML files. As there is one AD file per application, wizard.tcl must be launched for each application source and target of the coupling. In the following, we only present to creation step by step of the AD and PMIOD XML files of the source application.

In order to run the program wizard.tcl, just type:
./wizard.tcl (-silent)

4.1. Characteristics of the application

The first window that appears is about the characteristics of the application, as illustrated on the *Figure 4.1-1* below:

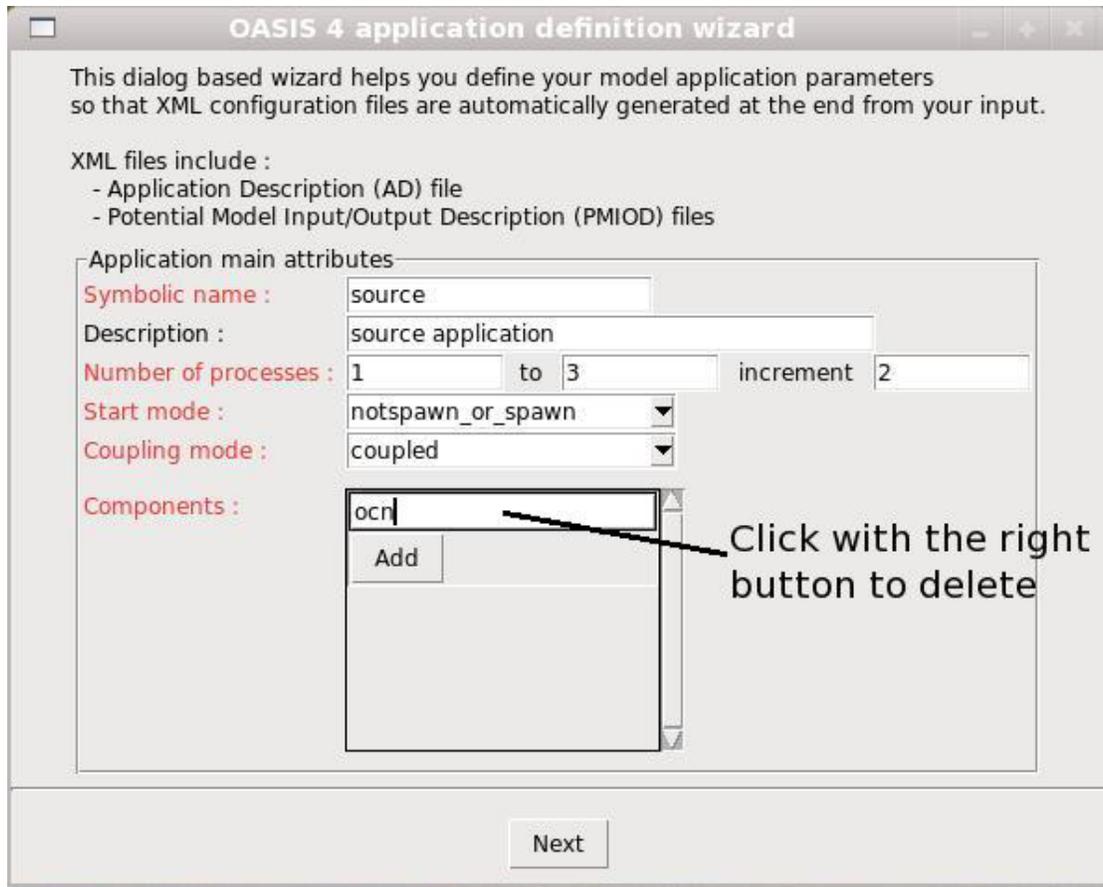


Figure 4.1-1: Characteristics of the application source

The symbolic name in red is mandatory and must correspond to the argument **appl_name** entered in the routine **prism_init** of OASIS4.

You must define the total number of processes on which the application can run. In our case, the application source can run only on 1 or 3 processors, which is expressed by the increment value equal to 2.

You must define the mode in which the application may be started (notspawn for MPI1, spawn for MPI2 or notspawn_or_spawn if both may be used), then the mode in which the application may run (coupled, standalone or coupled_or_standalone).

Finally you must enter the list of components included in the application. These names must correspond to the argument **comp_name** entered in the routine **prism_init_comp** of OASIS4. In our example, we only have one component ocr. When clicking with the right mouse button in one particular line of the list, a pop-up menu appears that lets you delete the line.

Then by clicking on **Next**, you will go to the characteristics of every component within this application.

4.2. Characteristics of each component included in the application

The window corresponding to the only component ocn defined in the application source of our example is shown *Figure 4.2-1*:

The screenshot shows a dialog box titled "OASIS 4 application definition wizard". It contains the following information:

- XML files include:
 - Application Description (AD) file
 - Potential Model Input/Output Description (PMIOD) files
- Component:
- Component main attributes:
 - Description:
 - Simulated:
 - Default component:
 - Number of processes: to increment
- Grids:
 -
 - Table with columns: Name, I periodic, J periodic, K periodic.
 - Row 1: grid_ocn, true, false, false. A red 'X' is in the right margin.
- Coupling fields:
 -
 -
 -
- Buttons: and

Figure 4.2-1: Characteristics of the component ocn

One component can not run on more processes than the number defined for the application.

For each component you must define all the grids that will be involved in the coupling with their directions of periodicity, and then all the fields that can be exchanged.

It is recommended to declare the fields of a component in identical ordering as in the corresponding coupled component.

In our example, there is only one grid `grid_ocn`, periodic in the I direction for the component `ocn`. This component reads data from a NetCDF file (coupling field `input_data_ocn`) and exchanges data with the component `atm` of the application target (coupling field `ocn_fnc_sent`).

You go to the characteristics of each field of each component by clicking on **Next**.

You can always come back by using the **Previous** button.

4.3. Characteristics of each field of each component

When clicking on **Next**, the window on the *Figure 4.3-1* appears, corresponding to the first field `input_data_ocn` entered in the list of the component `ocn`:

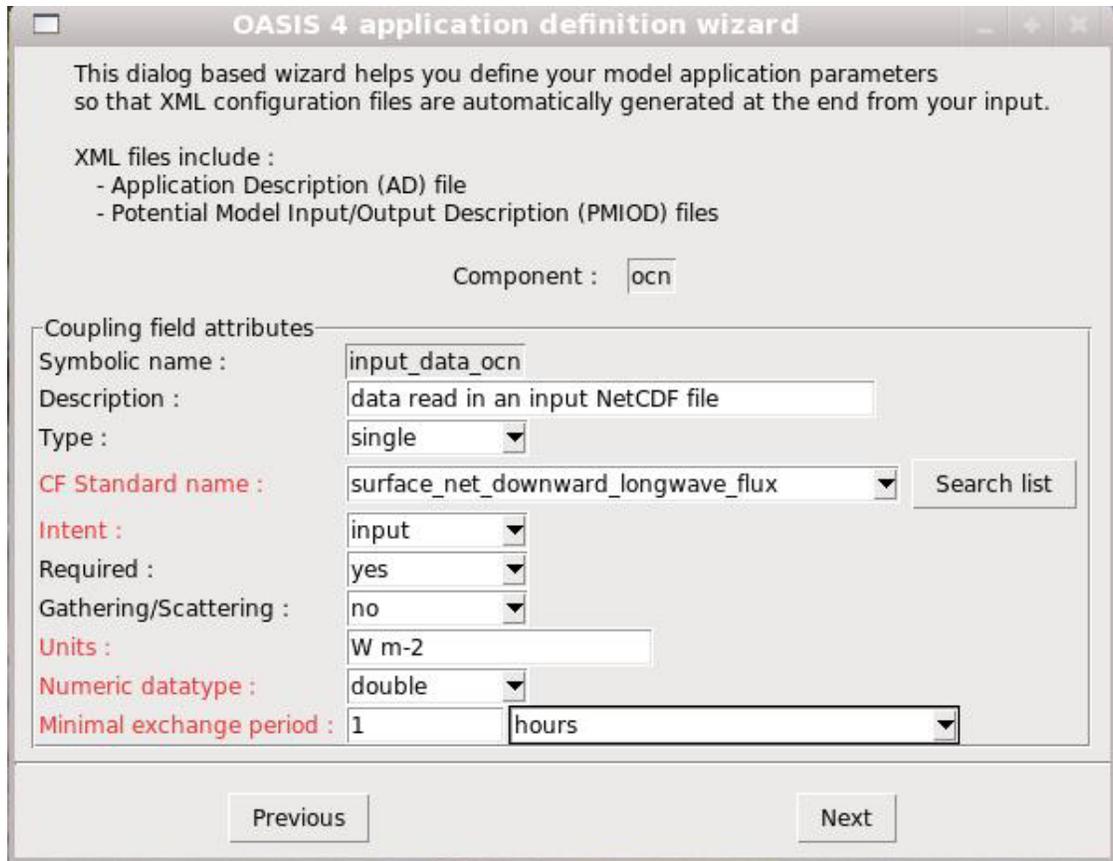


Figure 4.3-1: Characteristics of the field `input_data_ocn`

When clicking on **Next**, you will have to fill up the characteristics of the second field `ocn_fnc_sent` of the component `ocn` as shown on the **Figure 4.3-2**:

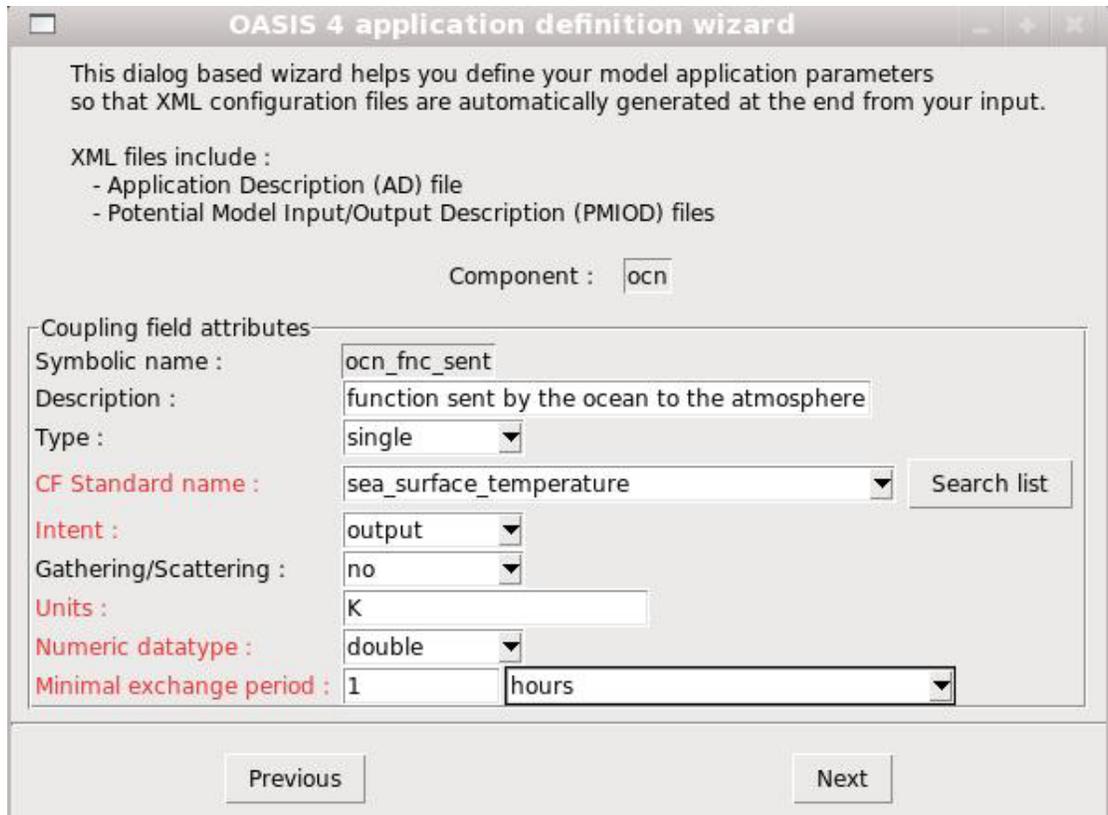


Figure 4.3-2: Characteristics of the field ocn_fnc_sent

The description of all the characteristics needed for the exchanged fields can be found in the OASIS4 User Guide. The CF standard names are read in the XML file cf-standard-name-table.xml provided with the utility and taken from the PCMDI (Program for climate model diagnosis and intercomparison) web site.

Be aware that the minimal period of exchanges is the time at which the calls to the routines of OASIS4 **prism_put** and **prism_get** are done in the component.

4.4. Saving the configuration files

When everything is fulfilled, you have to save your AD and PMIOD XML files by clicking on **OK** as illustrated on the *Figure 4.4-1* below. The files are stored in the directory where you launched the wizard.tcl, and named accordingly to OASIS4 naming rules : **<appl_name>_ad.xml** and **<appl_name>_<comp_name>_pmiod.xml**

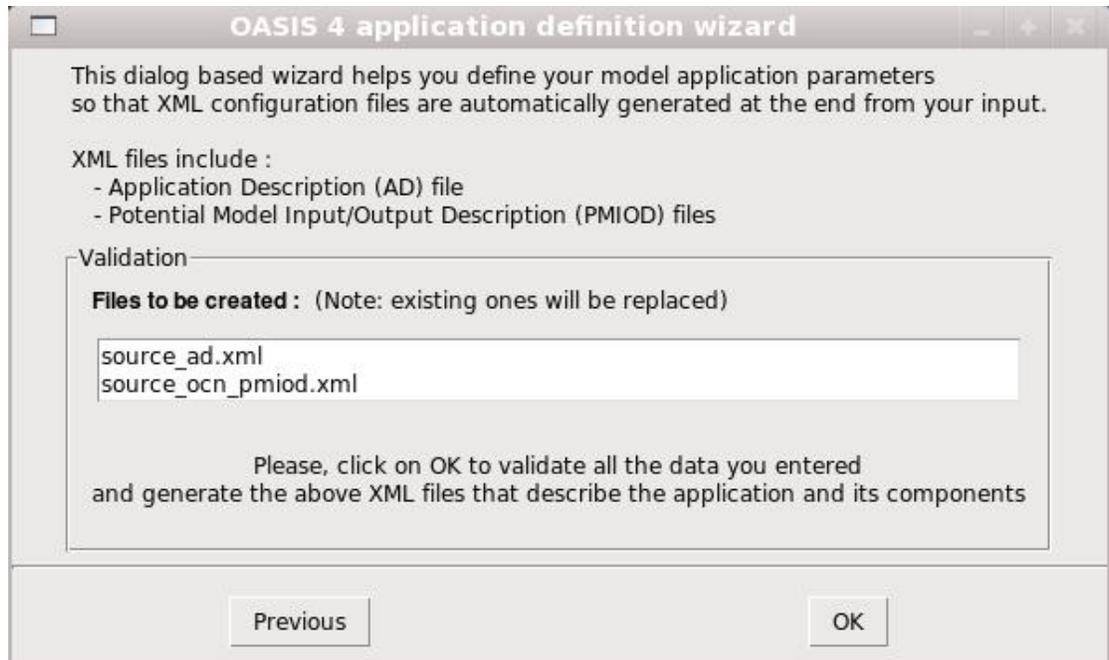


Figure 4.4-1: Creation of the AD and PMIOD file for the source application

5. Running oasis-gui.tcl to create SCC and SMIOC files

The Oasis-gui is a tool that allows the user to graphically define a coupling configuration (SCC and SMIOC XML files), from the AD and PMIOD XML files created with the wizard.tcl interface. We will now describe the creation step by step of the coupling configuration and the associated SCC and SMIOC XML files, corresponding to the AD and PMIOD XML files created above.

To run the interface oasis-gui.tcl, type :
`./oasis-gui.tcl (-silent)`

5.1. Description of the global graphical interface

The global interface, representing our example of coupling when everything has been full filled, is plotted on the *Figure 5.1-1*:

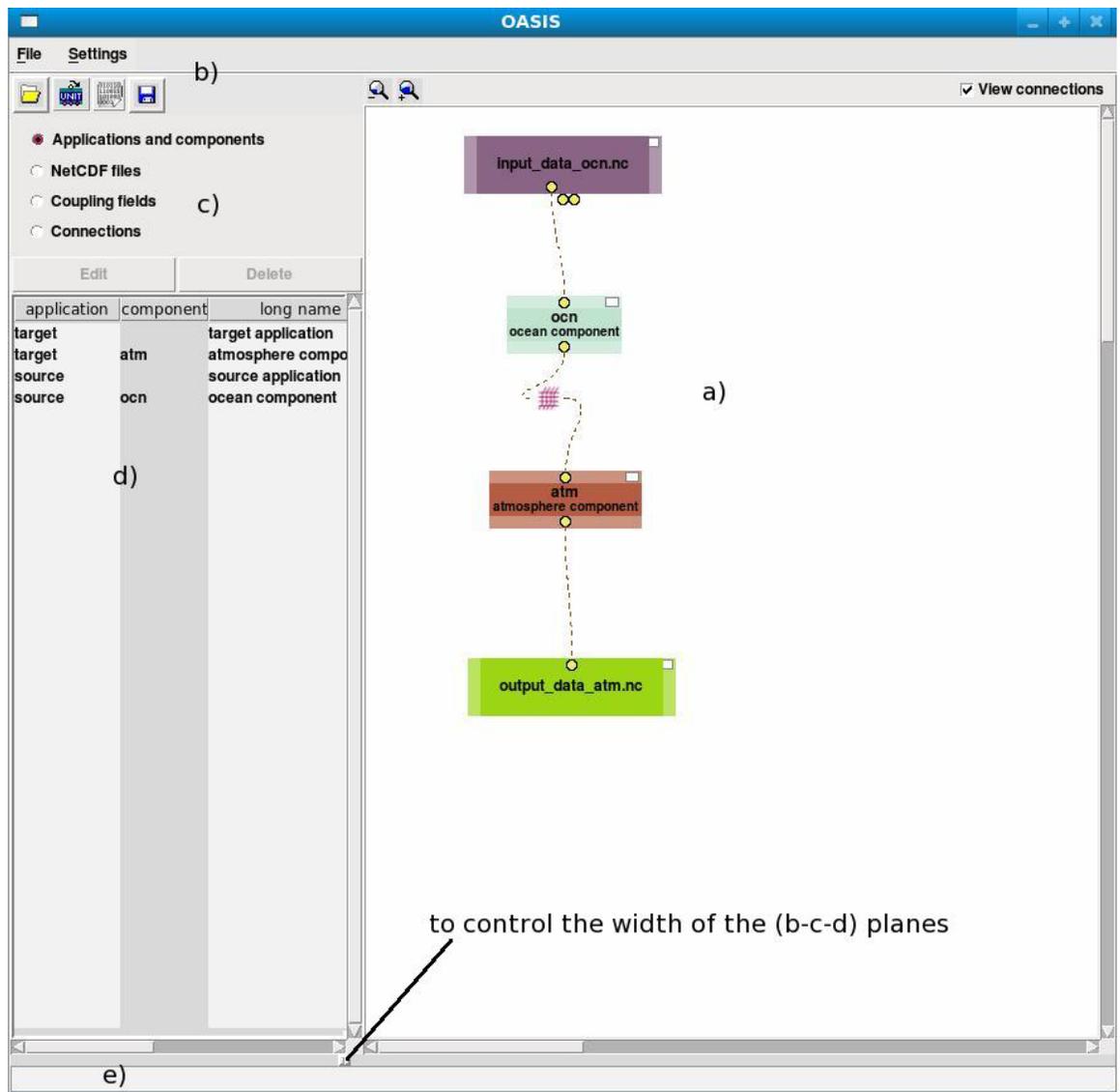


Figure 5.1-1: oasis-gui interface with our coupling example

The main window consists of five planes:

- a- The canvas where a graphical representation of the current coupling is drawn
- b- The menu and command buttons
- c- The radio buttons selecting the type of elements listed in (d)
- d- The list of elements participating to the coupling
- e- The message bar where is displayed information related to mouse cursor position

The user can control the width of the left panes (c-d) by clicking on the slider appearing below the bottom-right corner of pane (c-d).

The connections you created will disappear from the screen if you deactivate the **View connections** option.

5.2. Description of the buttons bar

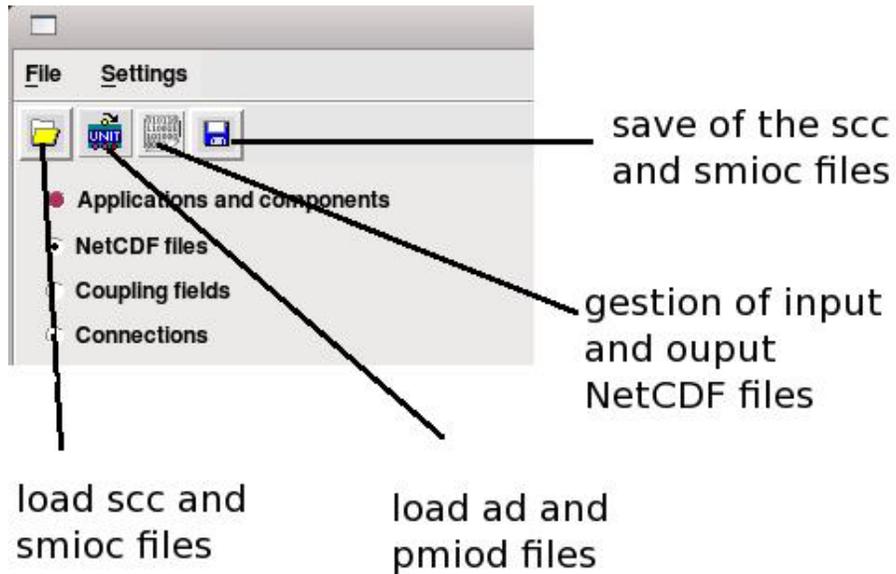


Figure 5.2-1: Buttons bar

The first widget, on the left, on the *Figure 5.2-1* can be used to load an already existing SCC XML file and its associated SMIOCI XML files (see also the **section 5.10**). The second one is used to load an application and its components (AD and PMIOD XML files). The third one is used to define new or to load existing NetCDF files. The last one is used to save the SCC and SMIOCI XML files at the same time (see also the **section 5.10**).

The two menus **File** and **Settings** will be described throughout the document.

5.3. Loading the AD and PMIOD files

The first step is to load the XML files AD and PMIOD which have been created with the wizard.tcl program before by using the menu:

File/Load Application Descriptions

You must choose the directory where the files are located and then select the AD files as shown on the *Figure 5.3-1* below:

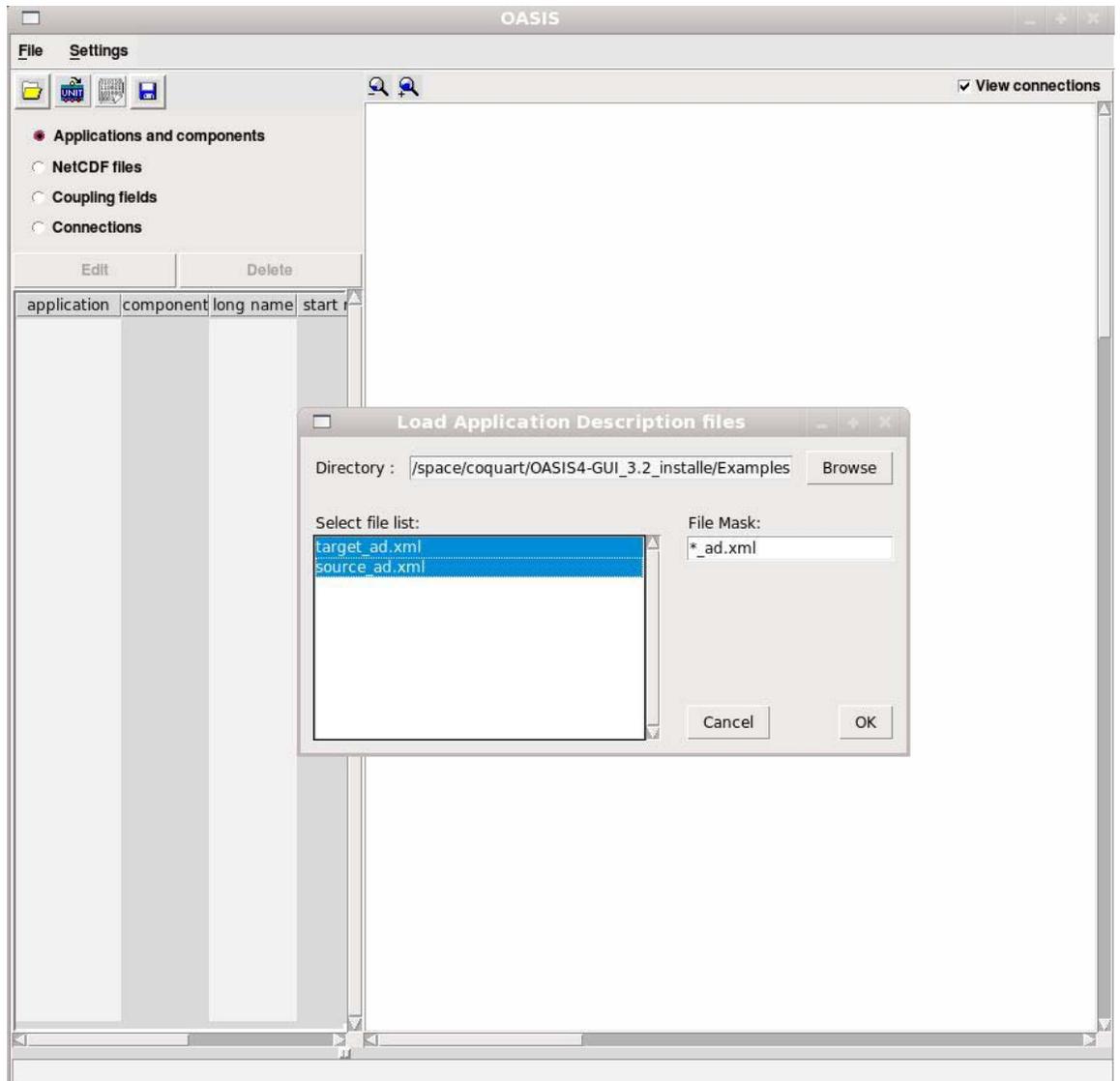


Figure 5.3-1: Loading the AD and PMIOD files

The PMIOD XML files are automatically loaded when you load the AD XML files.

Once the AD and PMIOD XML files are loaded, a box associated to each component appears in the canvas (see the **Figure 5.3-2** below):

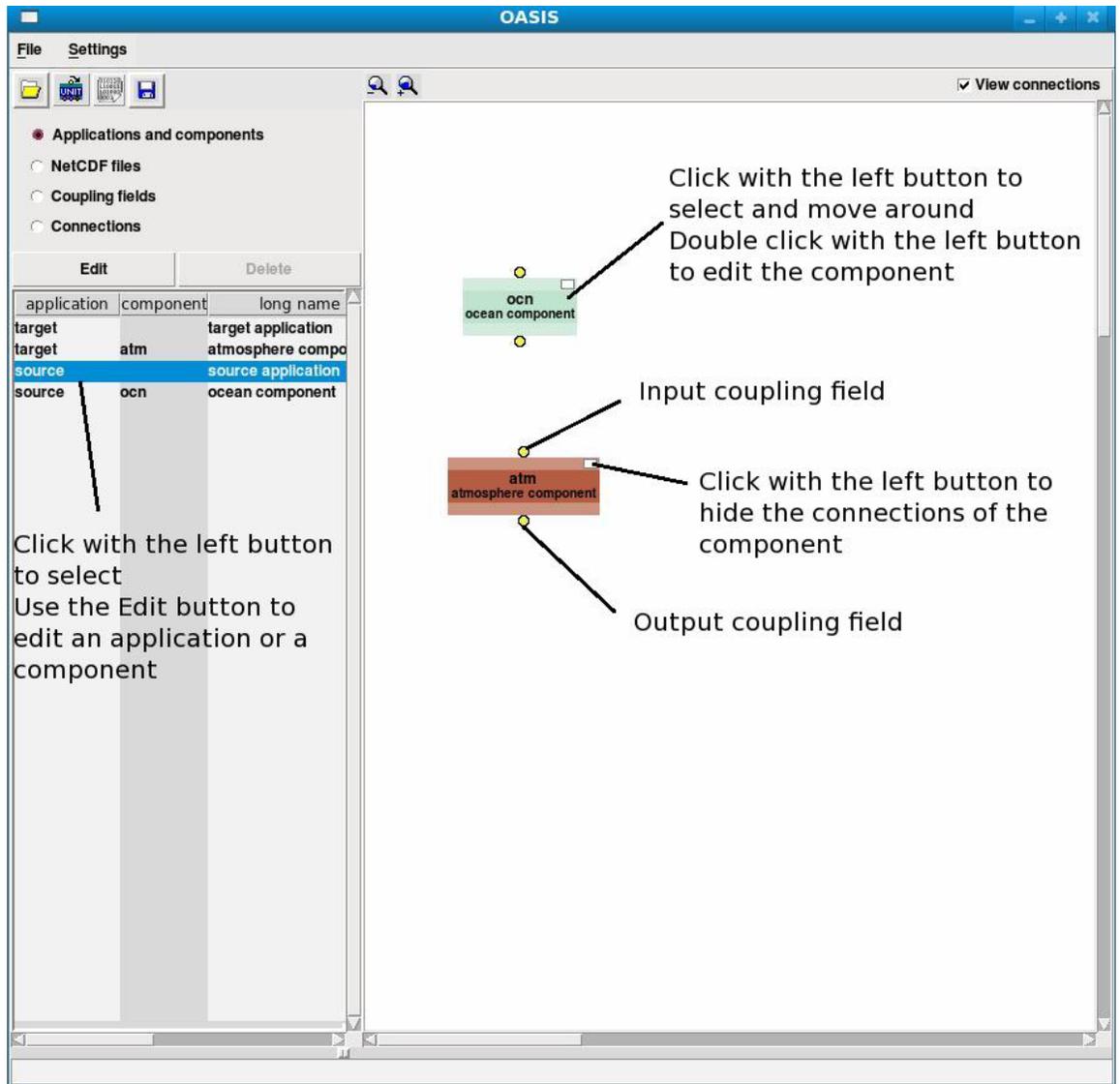


Figure 5.3-2: Components ocn and atm

The boxes can be moved inside the window by using the left button of the mouse.

The coupling fields of the component are represented by the yellow bullets attached to the boxes. The INPUT fields are at the top of each box while the OUPUT fields are at the bottom of the boxes. Their number and their names have been obtained from the information given in the PMIOD XML files.

5.4. Attributes of the applications and components

You must define the attributes of the applications that will appear in the SCC XML file. You first have to select an application in the left window and then

use the **Edit** button. For example, if you select and edit the application source you get the *Figure 5.4-1* below:

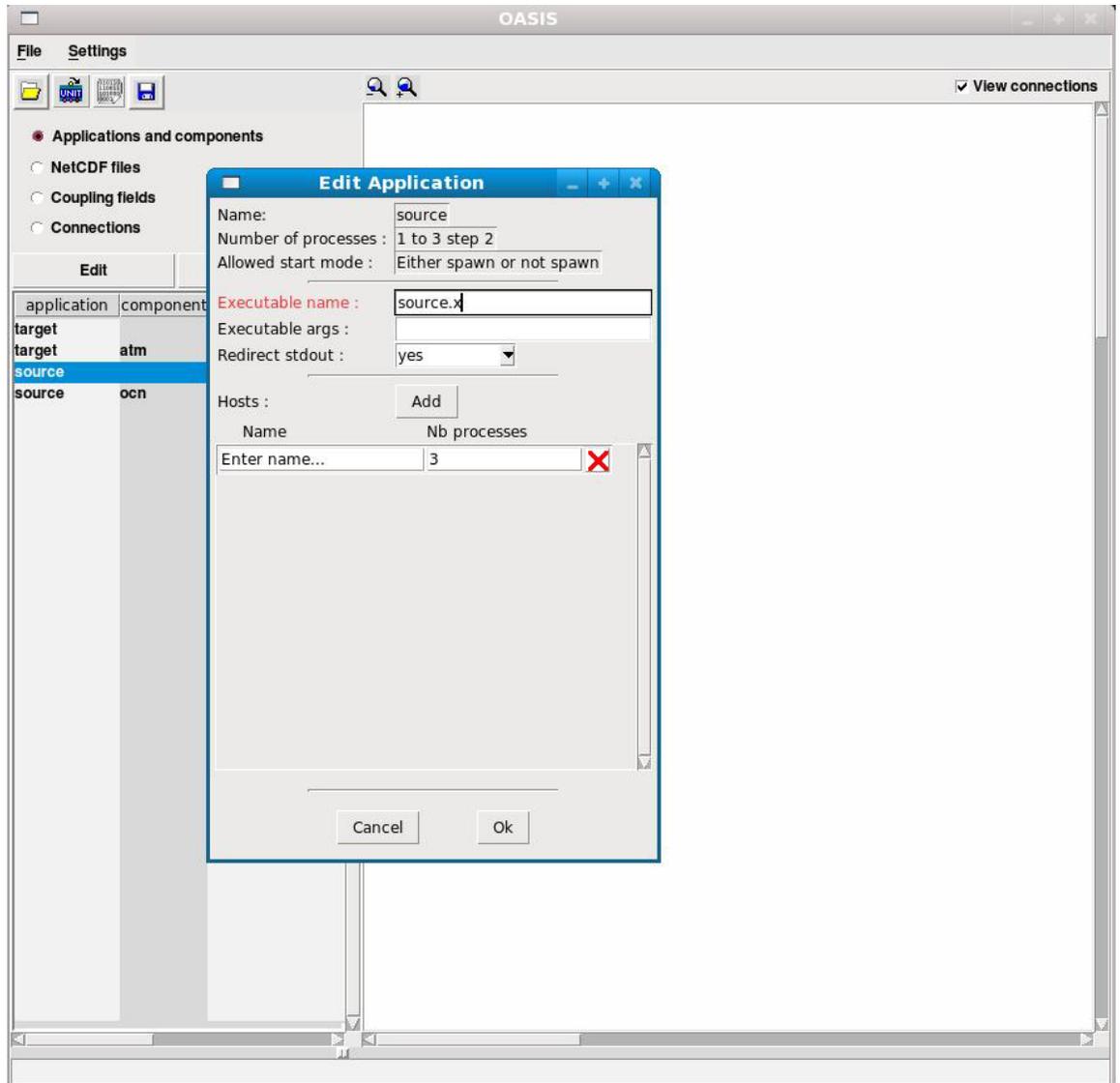


Figure 5.4-1: Characteristics of the application source

The name of the executable (in red) is mandatory. The arguments needed by the executable are optional. We recommend to **keep “yes”** for the **Redirect stdout** option, as in this case each application has its own output files and the results are much simpler to analyze.

In MPI1, only the total number of processes (here 3) is required, on one line. In MPI2, the names of the different hosts are required with the number of processors assigned to them.

The total number of processes must stay lower or equal to the maximum number given in the AD XML file, which is displayed in the second line of the form. We recall that in this example the application source can only run on 1 or 3 processors.

We do not present any picture of it, but you must do the same for the application target.

5.5. Attributes of the components

You must also define the attributes of the components that will appear in the SCC XML file. You have first to select a component in the left window and then use again the **Edit** button. If you edit the component ocn, the following window appears on the *Figure 5.5-1*:

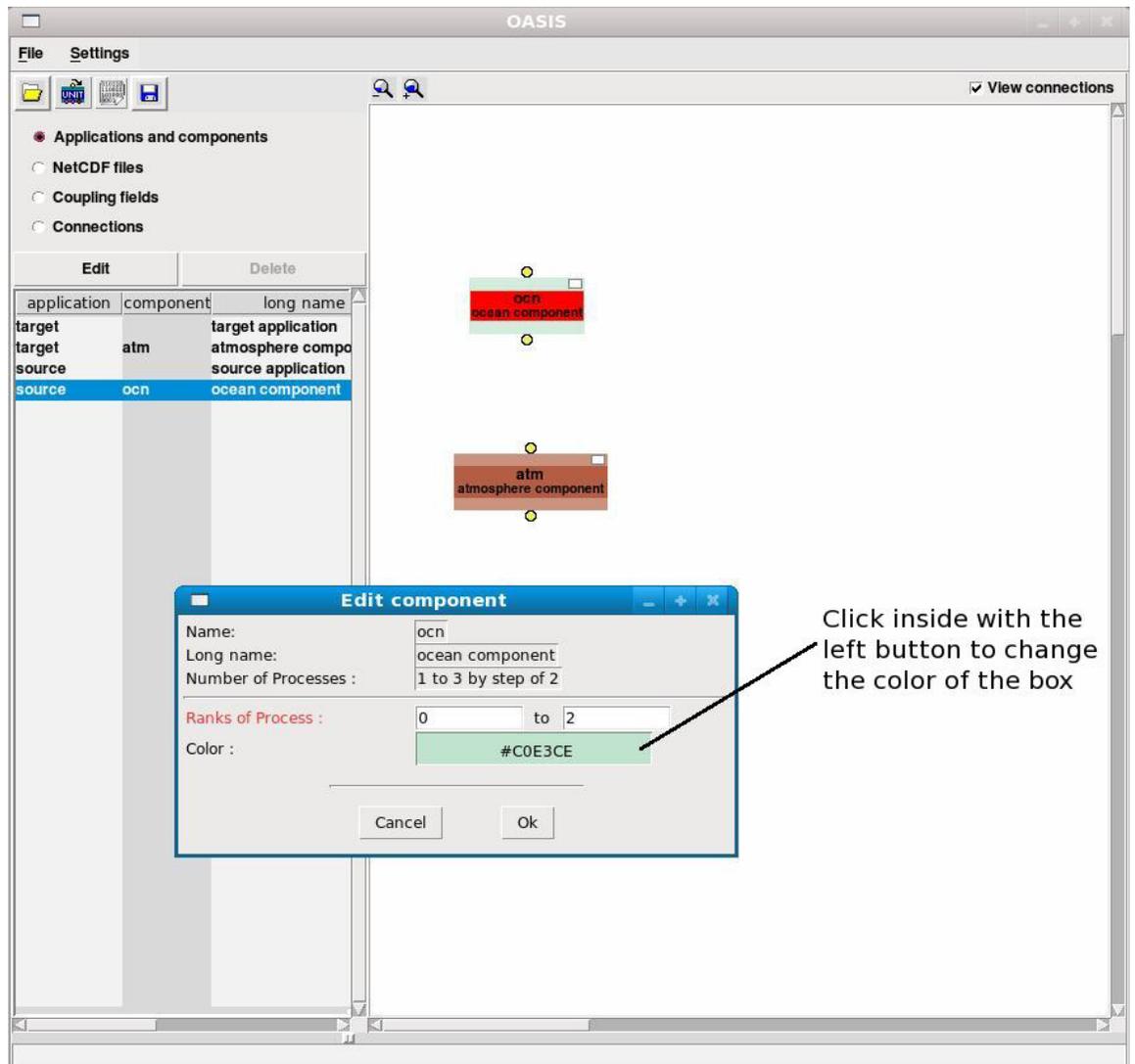


Figure 5.5-1: Characteristics of the component ocn for the SCC file

You must enter the ranks on which the component will run. You can also change the color of the box of the component, by clicking with the left button of the mouse on the corresponding line.

Be aware that you have to edit all the components even if you do not change the ranks of the processes (initialized by default at 0 to `nbr_proc_appli - 1`), to be able to save the SCC XML file in the **section 5.10**.

5.6. Input NetCDF files

You can load existing input NetCDF files by using the menu:

File/Load NetCDF Files

These files can be restart files (see OASIS4 User guide for their names) or not. In our example, we have to load the NetCDF file `input_data_ocn.nc` from which the component `ocn` will read its data (see the creation of the AD and PMIOD files in the **section 4.3**). When using the menu **File/Load NetCDF Files**, a new window appears, shown on the **Figure 5.6-1**:

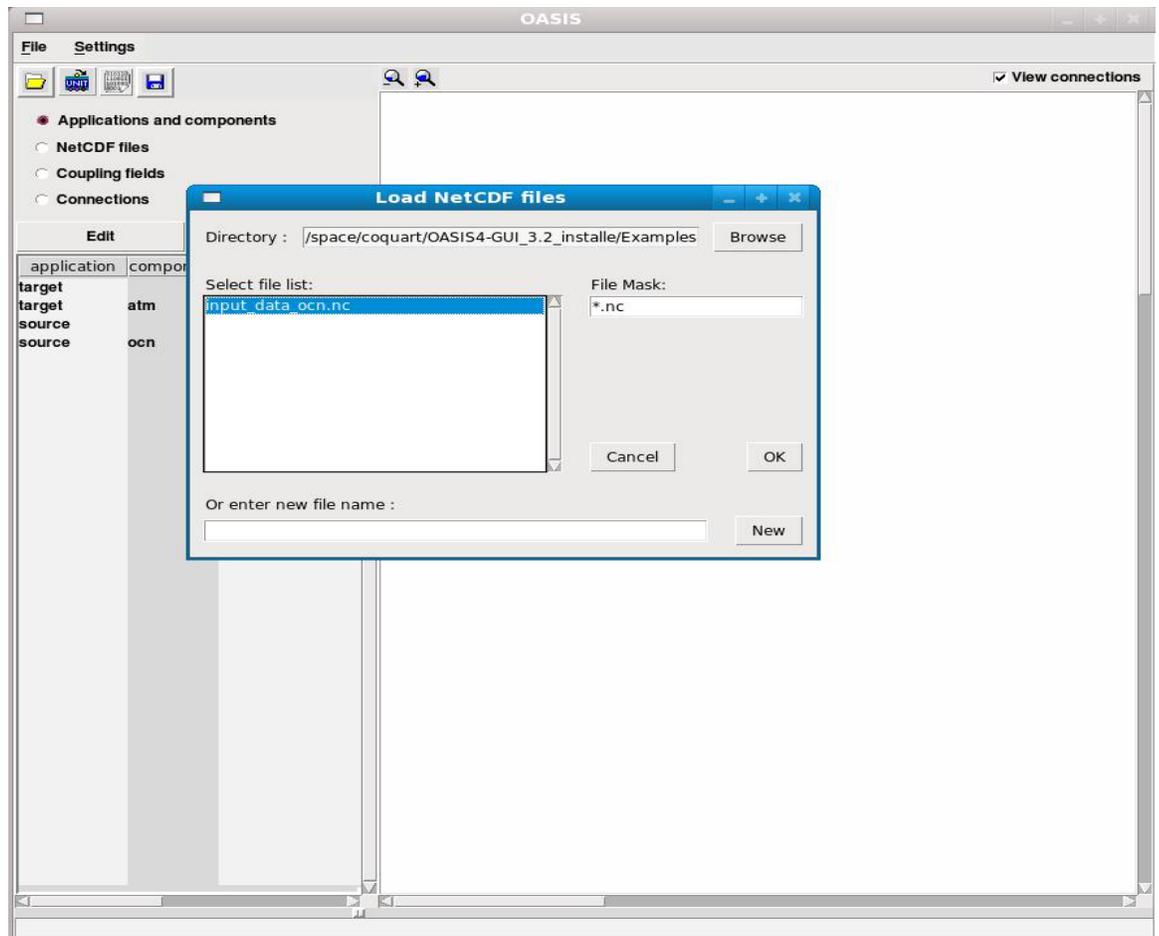


Figure 5.6-1: Loading the input NetCDF file for the component ocn

When clicking on the **OK** button, another window appears with the characteristics of the file as illustrated on the *Figure 5.6-2*. You validate your choices by clicking on the **OK** button.

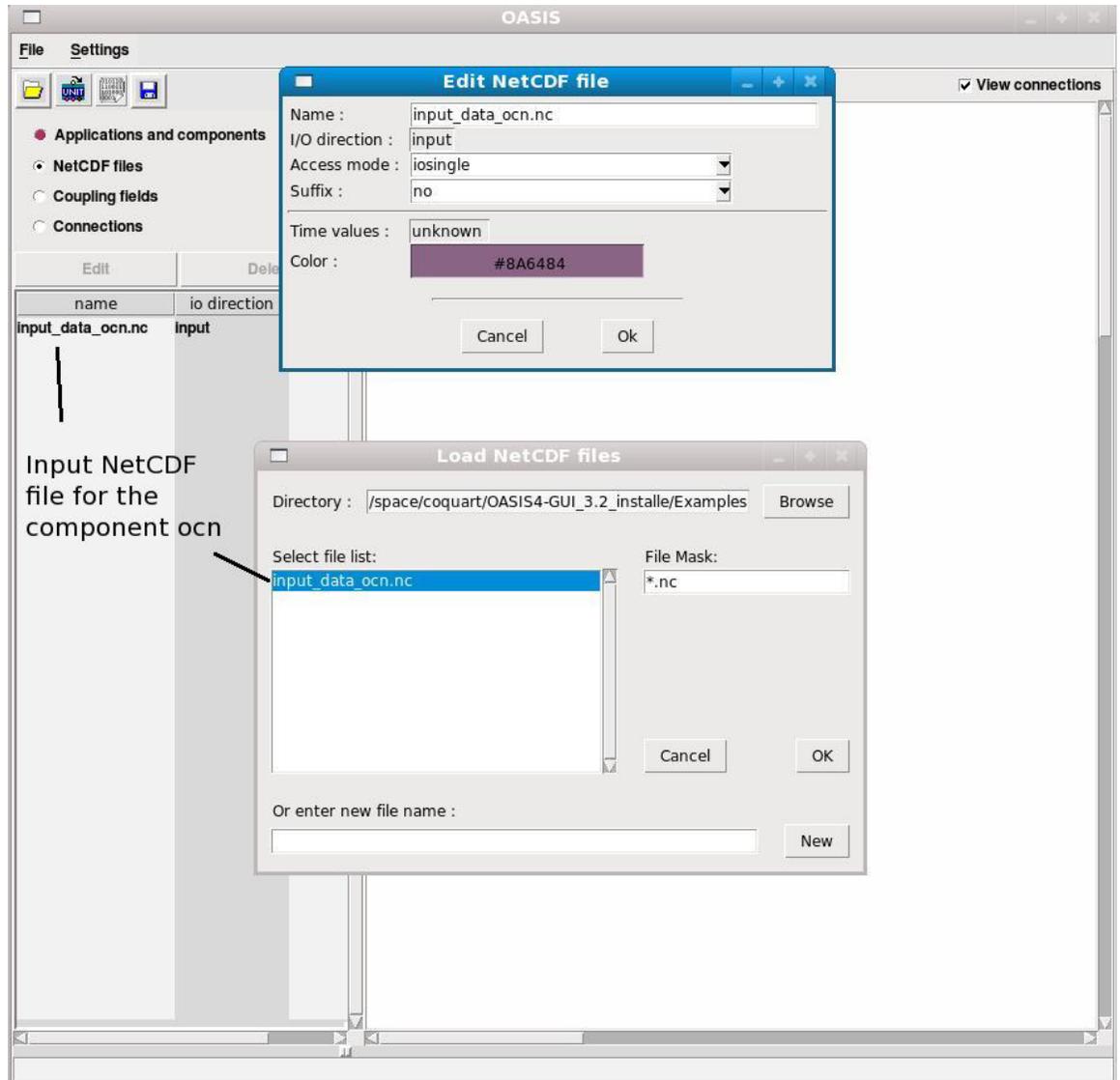


Figure 5.6-2: Characteristics of the input NetCDF file of the component ocn

5.7. Output NetCDF files

You also define the output NetCDF files by using the menu:

File/Load NetCDF Files

You must enter, in the window **enter new file name**, the name of a new file that will be written during the run as shown on the *Figure 5.7-1*:

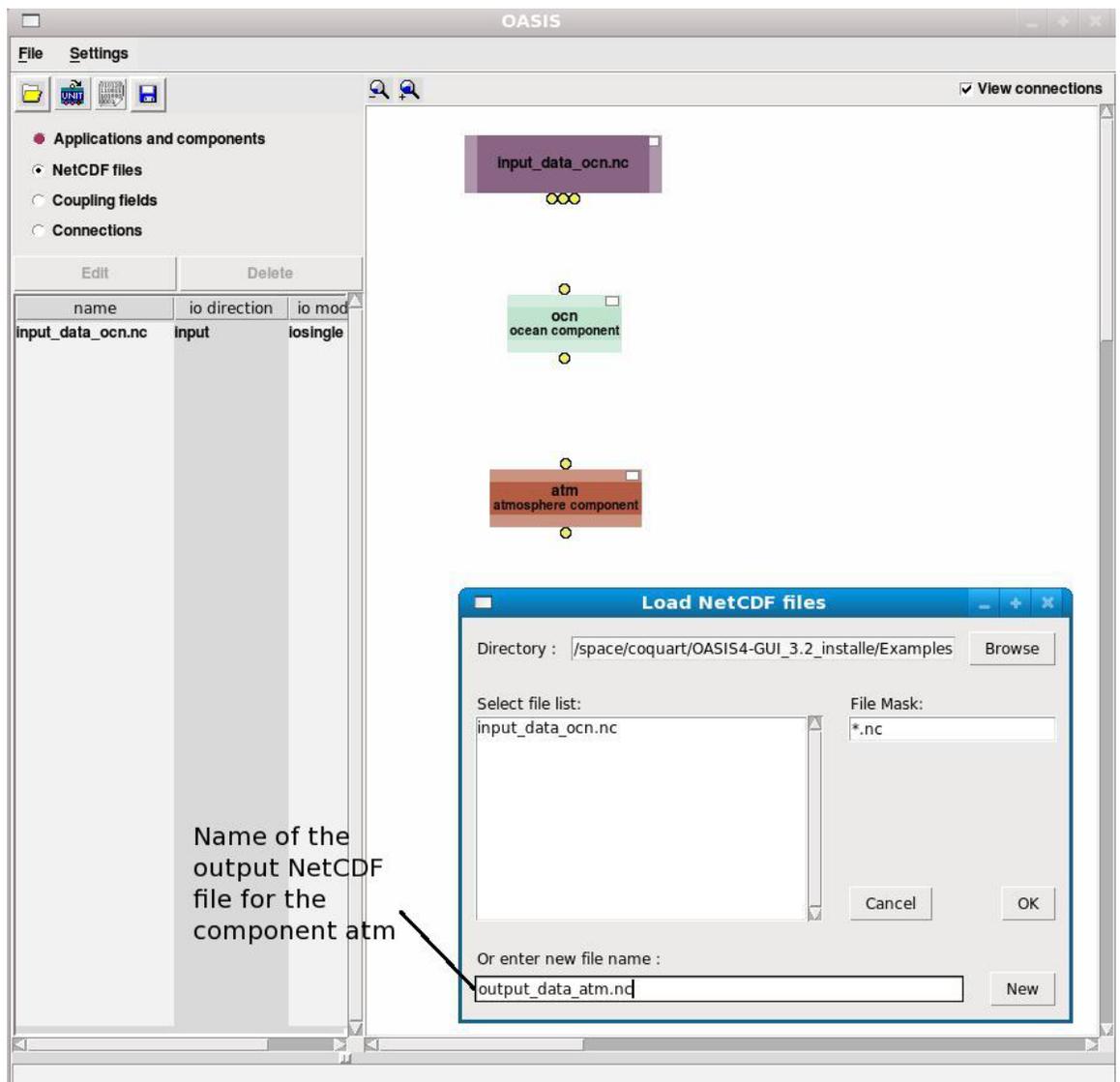


Figure 5.7-1: Defining the name of the output NetCDF file for the component atm

When clicking on the **New** button to validate your choice, another window appears plotted on the **Figure 5.7-2**, where you must enter the name of the field contained in the NetCDF file:

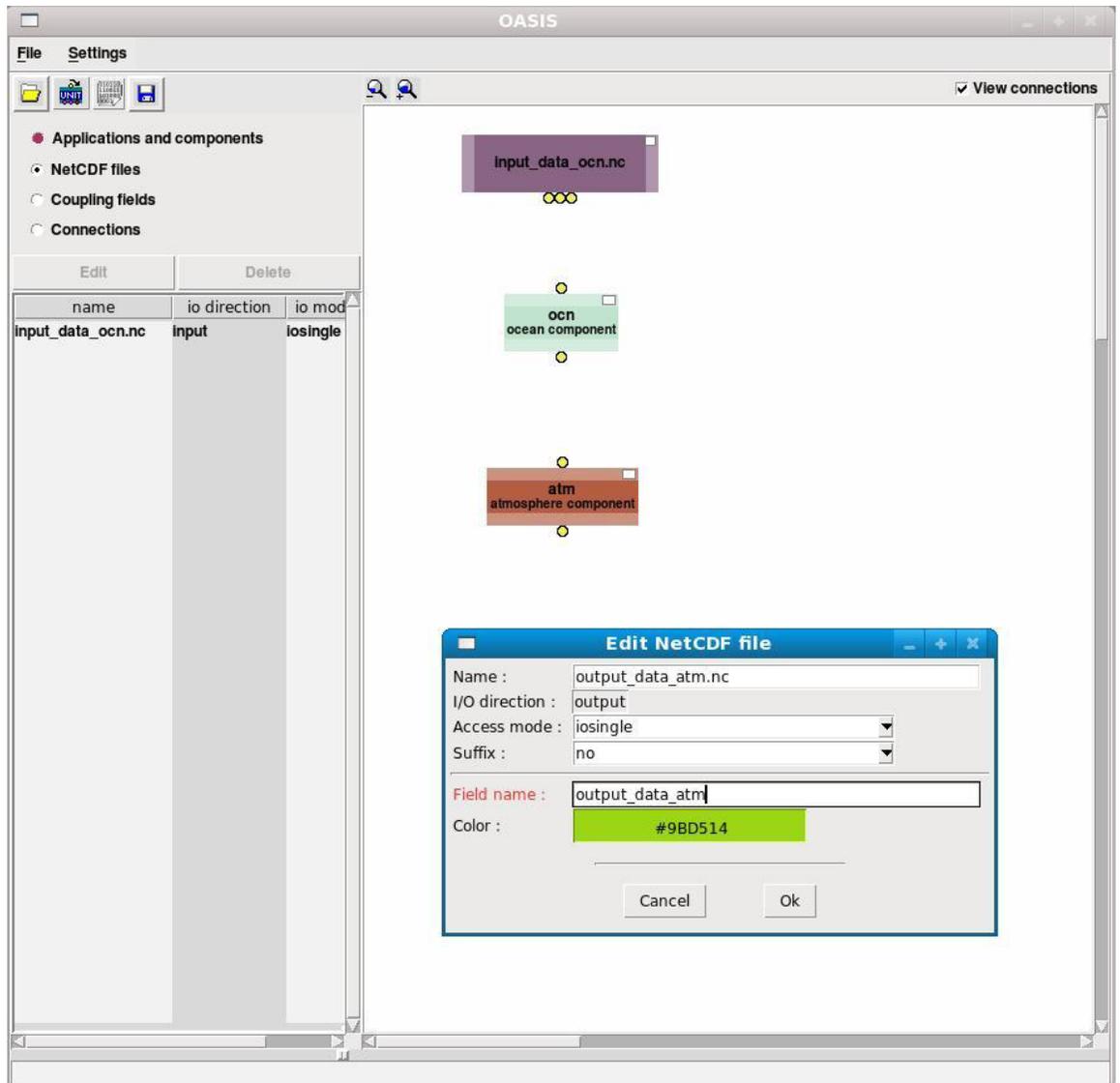


Figure 5.7-2: Characteristics of the output NetCDF file for the component atm

By clicking on the **NetCDF files** item, below the **Applications and Components** radio button, the different NetCDF files involved in the coupling appear in the left window. You can **Delete** them or **Edit** them if necessary. Finally you obtain, *Figure 5.7-3*:

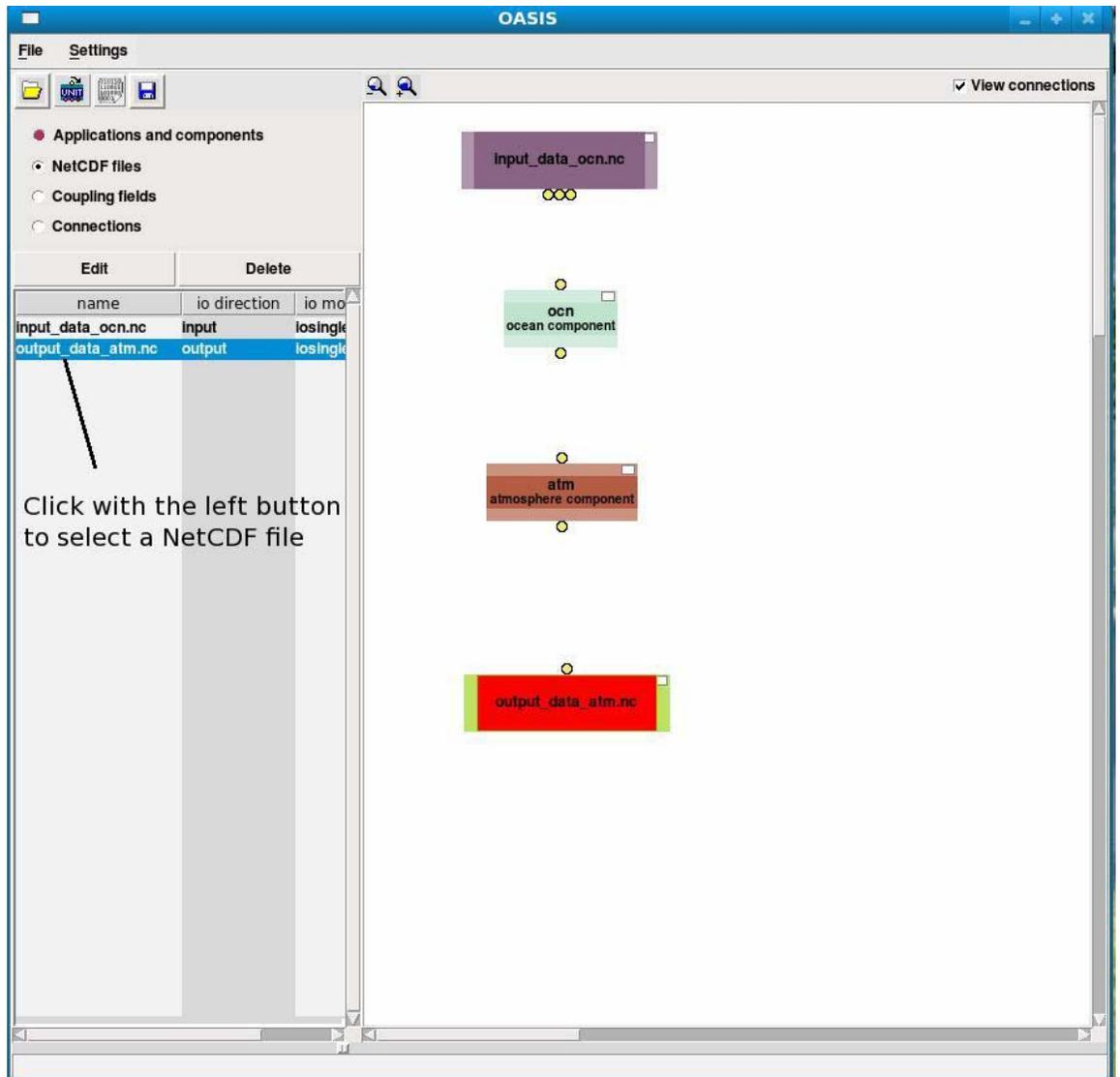


Figure 5.7-3: Boxes after loading components and NetCDF files

5.8. Coupling fields

Once the different applications, components and NetCDF files have been loaded, you must define the connections between the coupling fields.

When clicking on the radio button **Coupling fields** on the top-left pane c (see the *Figure 5.1-1*), all the fields involved in the coupling appear in the left window (see the *Figure 5.8-1*). Their names were obtained from the PMIOD XML files.

a- Establishing a connection between a component and an input NetCDF file

When connecting the component ocn to the NetCDF file input_data_ocn.nc, the yellow bullets become red and the system opens a window to let the user characterize the connection between the coupling fields as illustrated on the *Figure 5.8-1*:

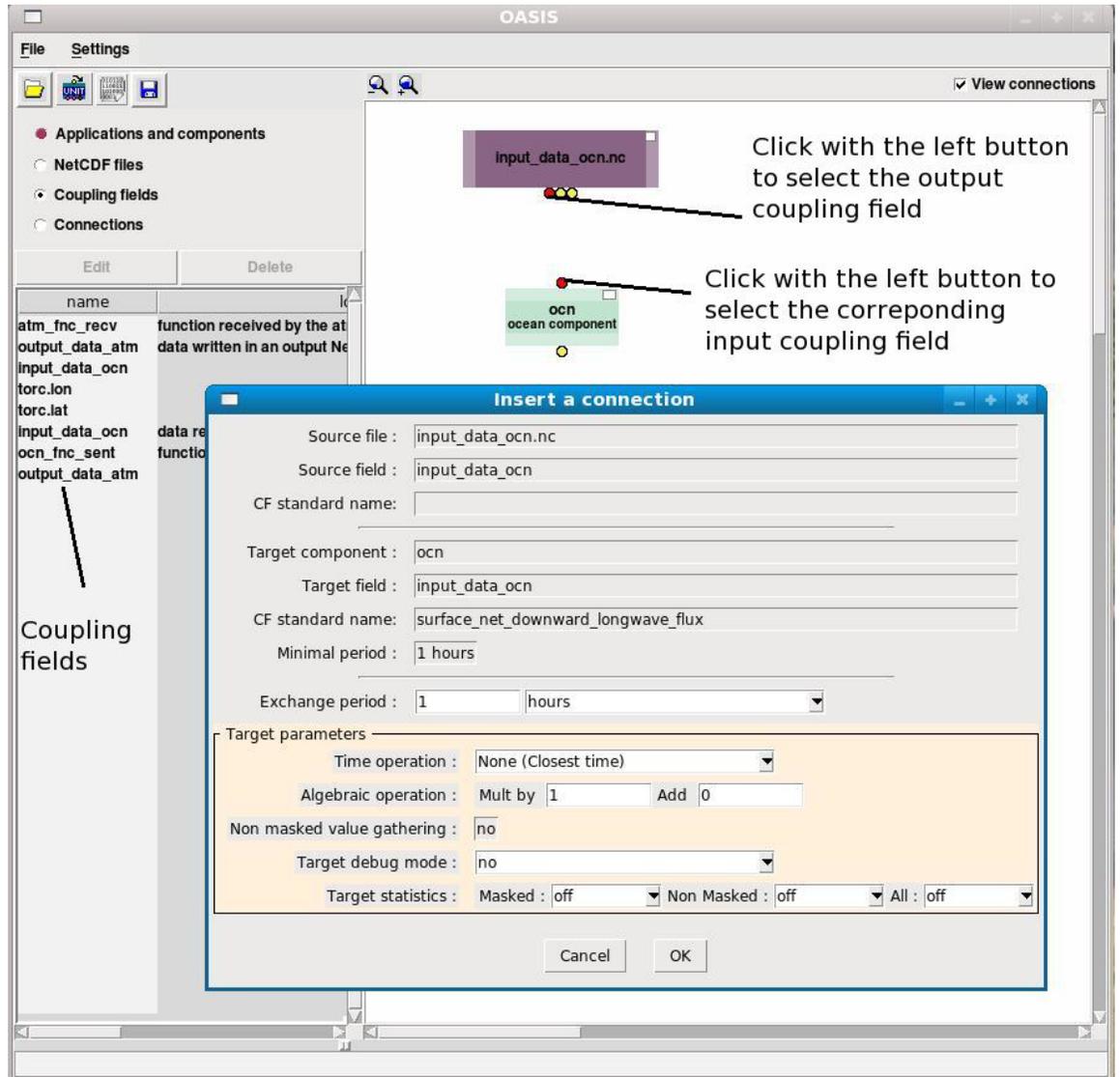


Figure 5.8-1: Connection between the component ocn and the input_data_ocn.nc file

Then you have a connection that appears in the main window, and in the left plane, as illustrated below on the *Figure 5.8-2*:

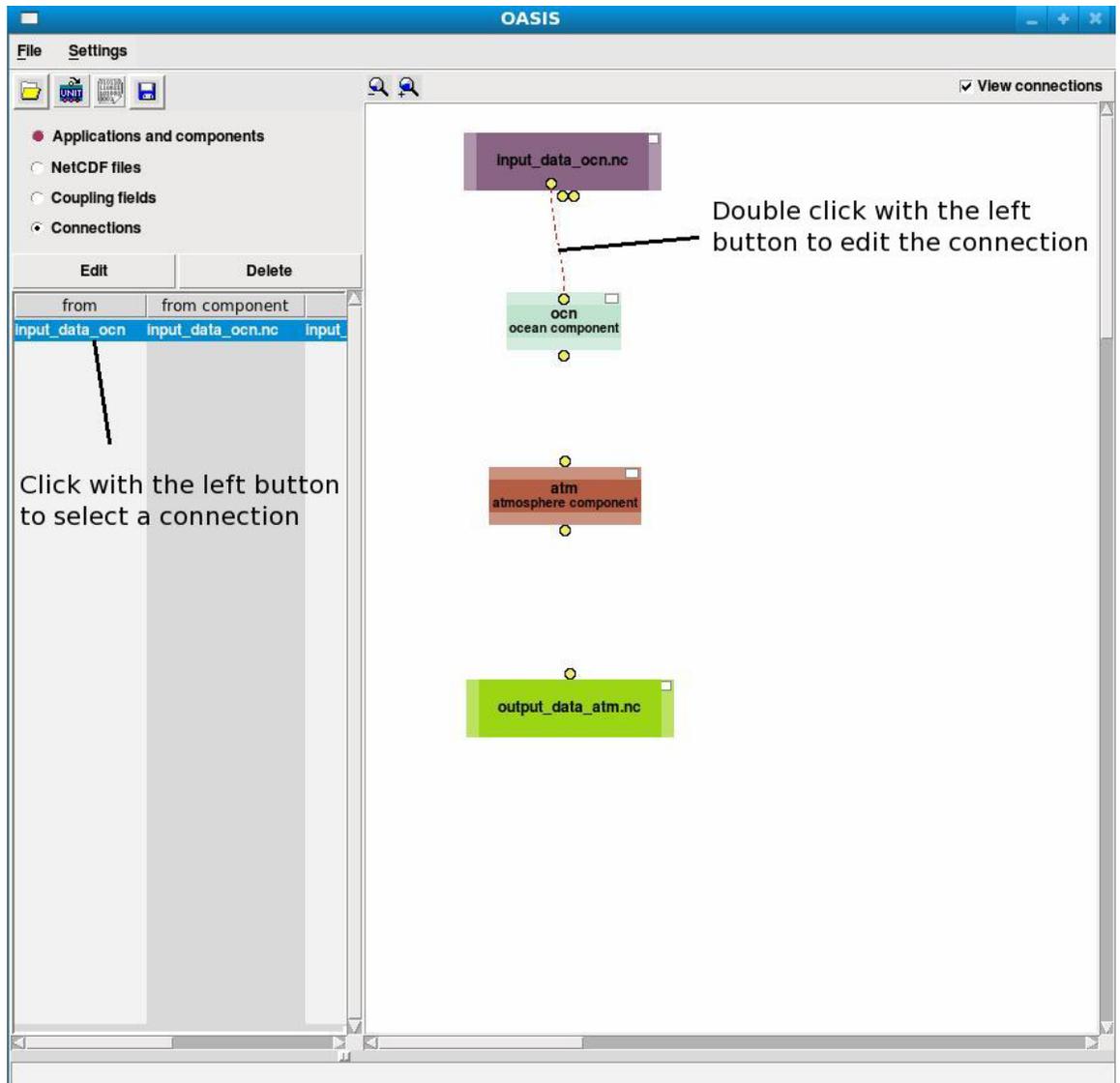


Figure 5.8-2: Graphical connection

Once you have selected a connection in the left window, you can **Edit** or **Delete** it. You can also double click on the graphical representation of it to edit it.

b- Establishing a connection between two components

When connecting the ocn component to the atm component, a new window appears as on the *Figure 5.8-3*:

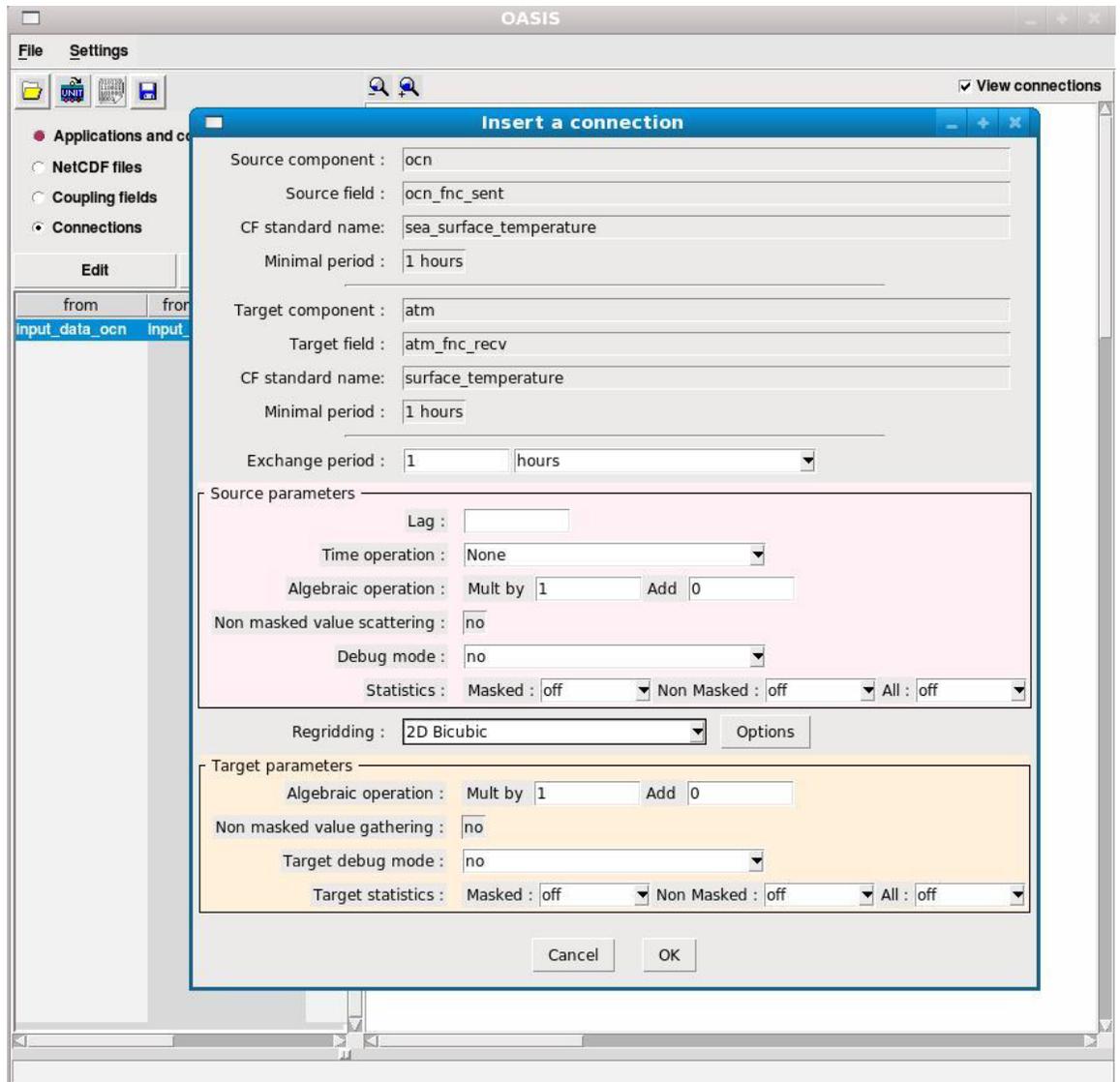


Figure 5.8-3: Connection between the component ocn and the component atm

You must enter the coupling period and define if there is a lag (see the OASIS4 User Guide). If the lag was equal to one, the coupling field `ocn_fnc_sent` would be read from a restart file at the beginning of the experiment and you would have had to define it as in the **section 5.8-a**.

You must also define the **Regridding** (type of interpolation) with its options by clicking on the right **Options** button.

You can also add a scalar or multiply the target field by a constant.

By default there is no lag (and then no restart files) and no operation on the target field.

The exchanged period is the period at which the coupling will be effectively performed.

b- Establishing a connection between a component and an output NetCDF file

When connecting the atm component to the output NetCDF file `output_data_atm.nc`, the window **Figure 5.8-4** appears:

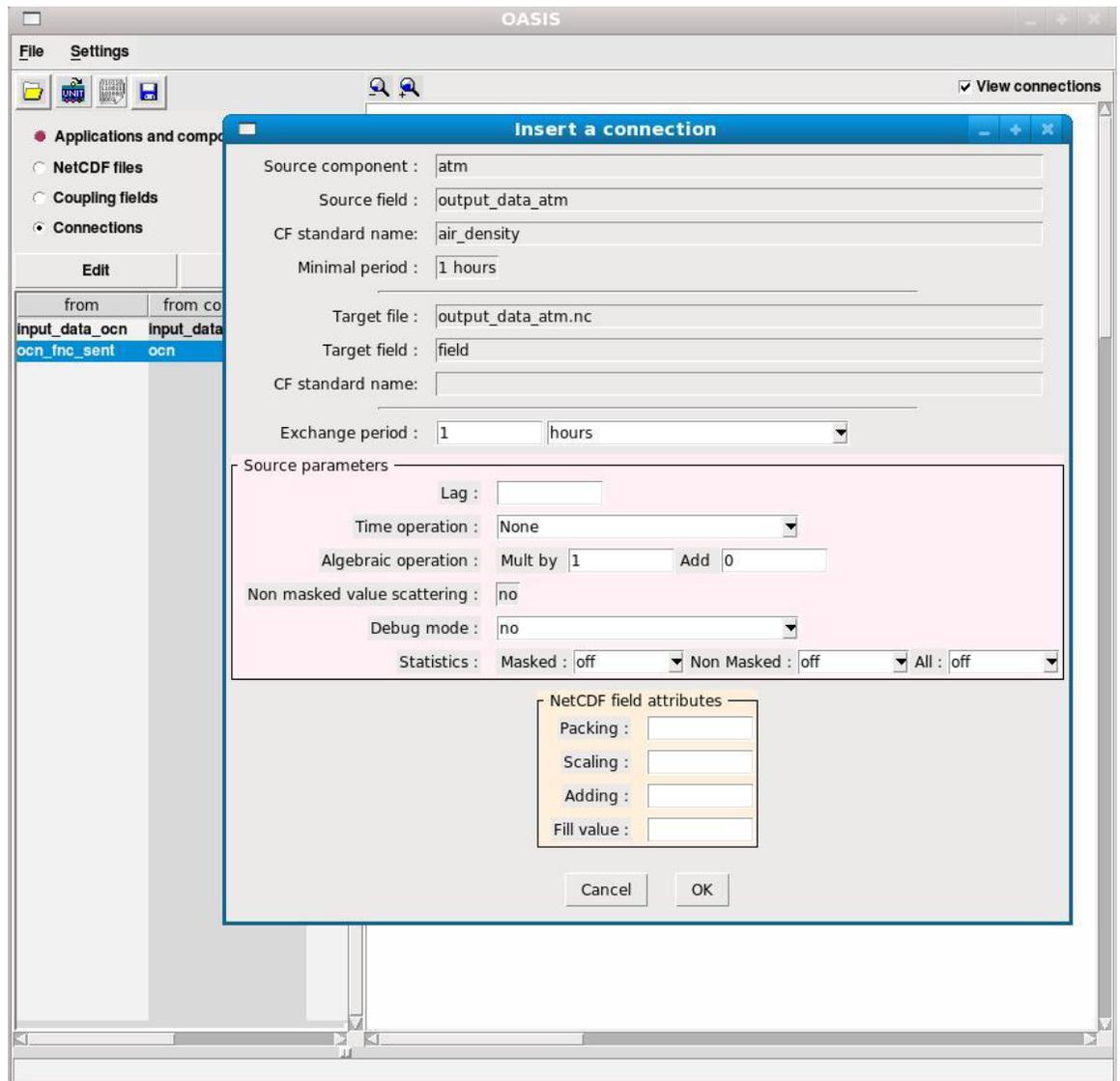


Figure 5.8-4: Connection between the component atm and the output_data_atm.nc file

The lag is only used in the stand alone mode for the creation of restart files (see the **section 6** and the OASIS4 User Guide).

Finally you get the graphical coupling configuration **Figure 5.8-5**:

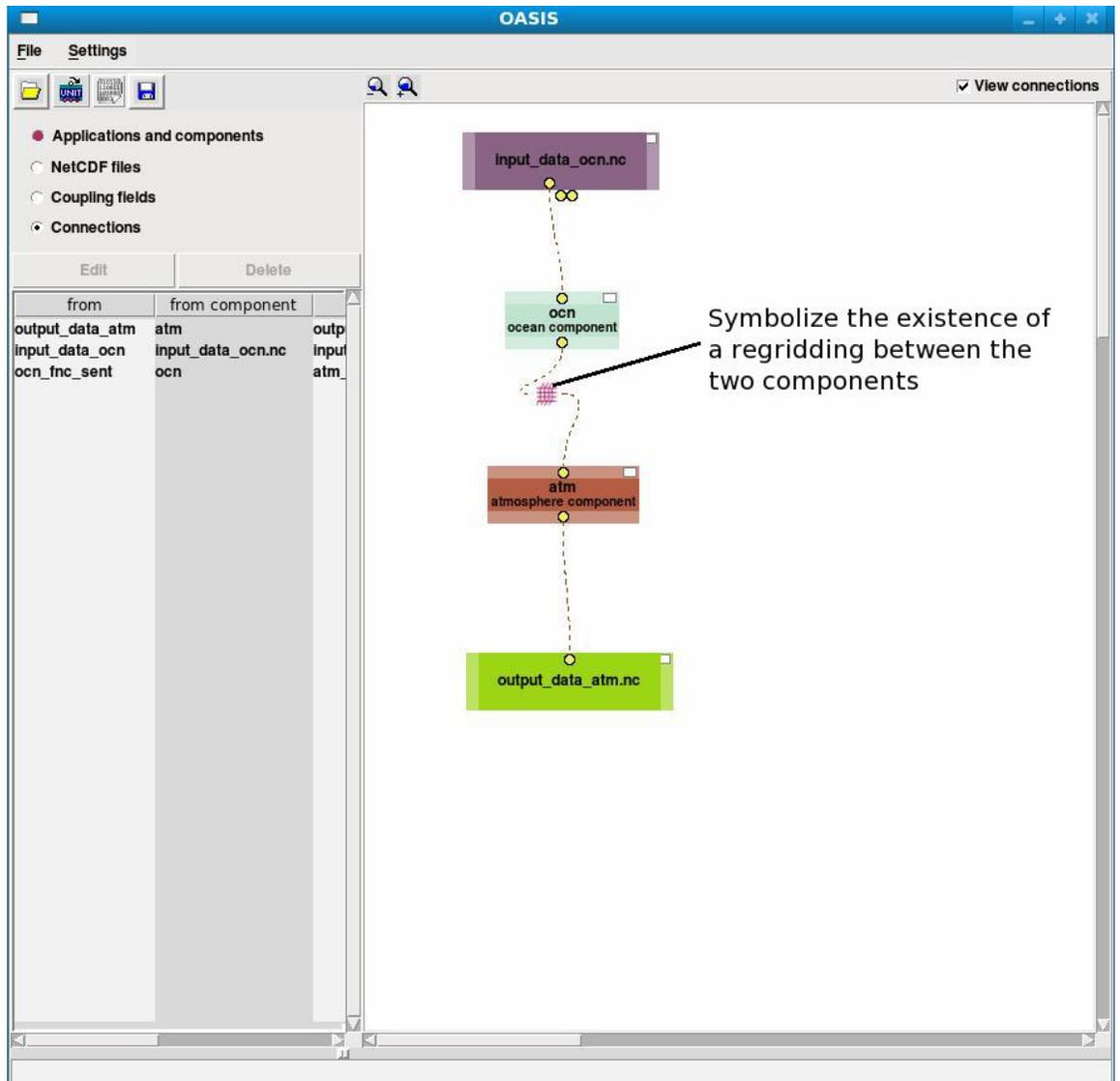


Figure 5.8-5: Graphical coupling configuration

To be able to save the SCC and SMIOC configuration XML files, you must fulfill the experiments settings that will be written in the SCC XML file as described now.

5.9. Experiments Settings

It is mandatory to fill up the experiment settings by using the menu: **Settings/Experiment settings**

A new window appears as illustrated on the *Figure 5.9-1*:

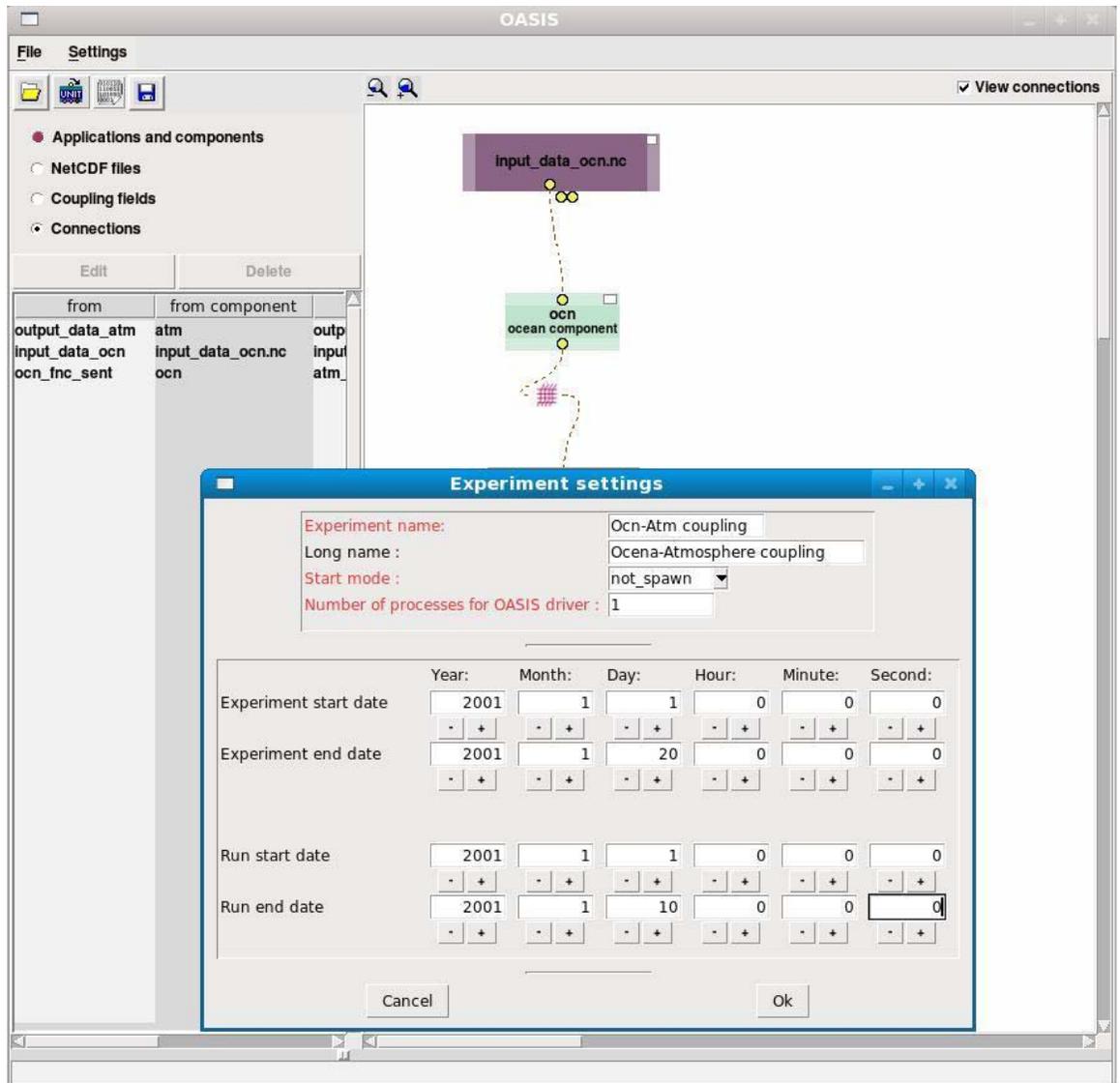


Figure 5.9-1: Experiment settings

The entry **Experiment name** is mandatory.

You have to precise the mode in which the application will be started (spawn in MPI2 or not spawn in MPI1) from what is allowed in the AD XML file and the number of processes for the driver.

You must finally define the Experiment dates and the Run dates (included in these experiment dates, see the OASIS4 User Guide).

5.10. Saving and/or loading a coupling configuration

There are different ways to save the SCC and SMIOC configuration XML files:

*a- By using the **Save** button (see the section 5.2) or the menu **File/Save all***
 In this case the SCC and SMIOC XML files are saved at the same time. A window appears (see the **Figure 5.10-1**) and you can choose the directory where you want to save these files:

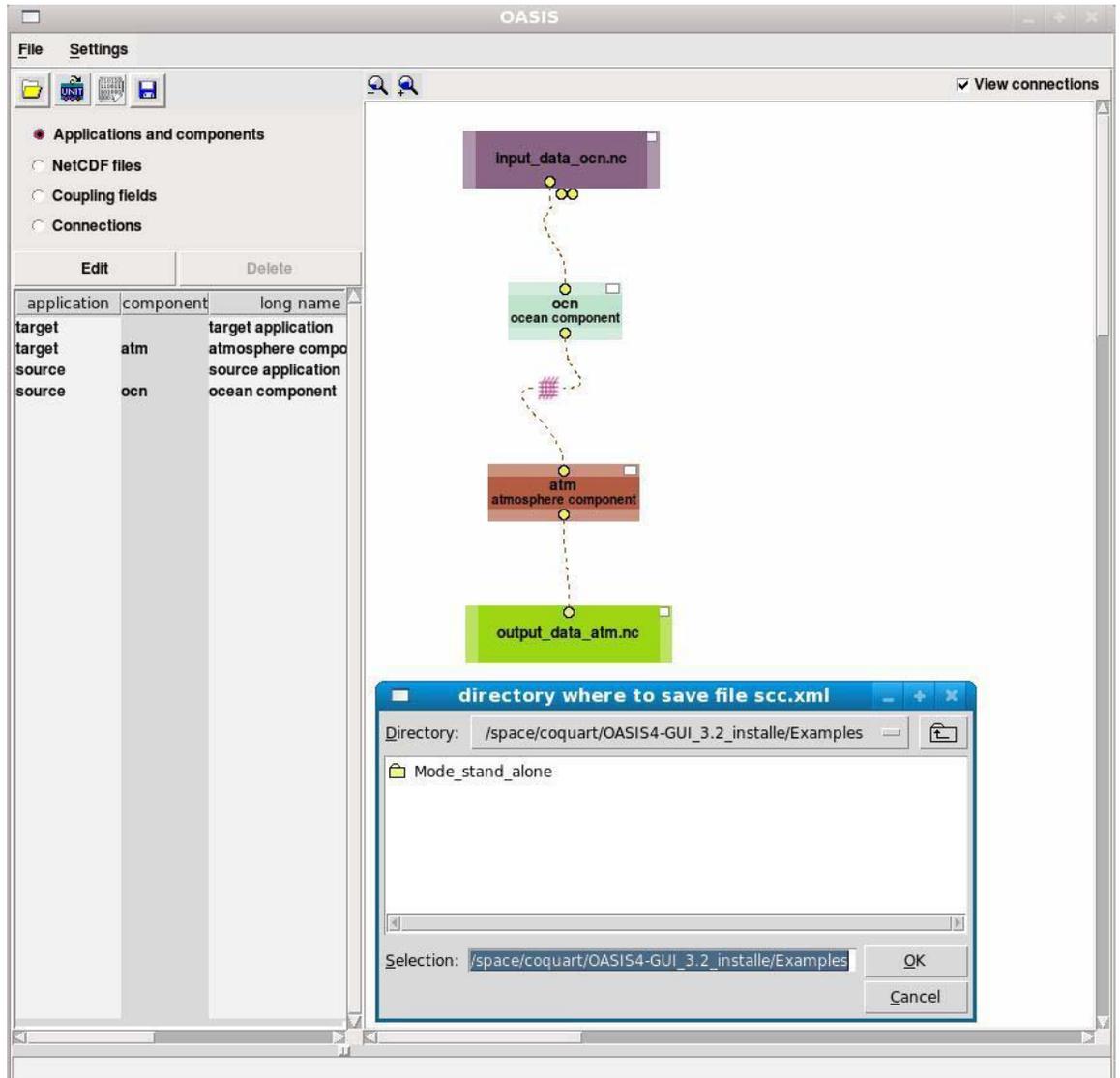


Figure 5.10-1: Saving the SCC and the SMIOC XML files

b- By using the file menu

You can also save separately the SCC file and the SMIOC XML files by using the menu File/Save SCC file or the menu File/Save SMIOC files .

When saving the SCC XML file the « geometry » of the coupling in the main window is stored in the file Graphics.xml.

If you want to load an old configuration created with the interface, you must have all the XML files, AD, PMIOD, SCC and SMIOC and Graphics.xml in the same directory.

You can also load an existing SCC XML file if only the AD and PMIOD XML files are present in the directory.

6. The mode standalone

In this case the application is running alone, without exchanging data with another component through the driver. This particular mode can be used to create restart files with OASIS4 thanks to the routine **prism_put_restart** (see the OASIS4 User Guide).

In the following, we describe how to use the graphical interface to create the configuration files that will be used to create a restart file with OASIS4 for the component ocn of the application source. We suppose that we have already created the AD and PMIOD XML files with the wizard.tcl program (see the **section 4**). You can find these files in the directory

Examples/Mode_stand_alone.

You must begin by loading the AD and PMIOD configuration XML files of the application source, as shown below on the **Figure 6-1** (see also the **section 5.3**):

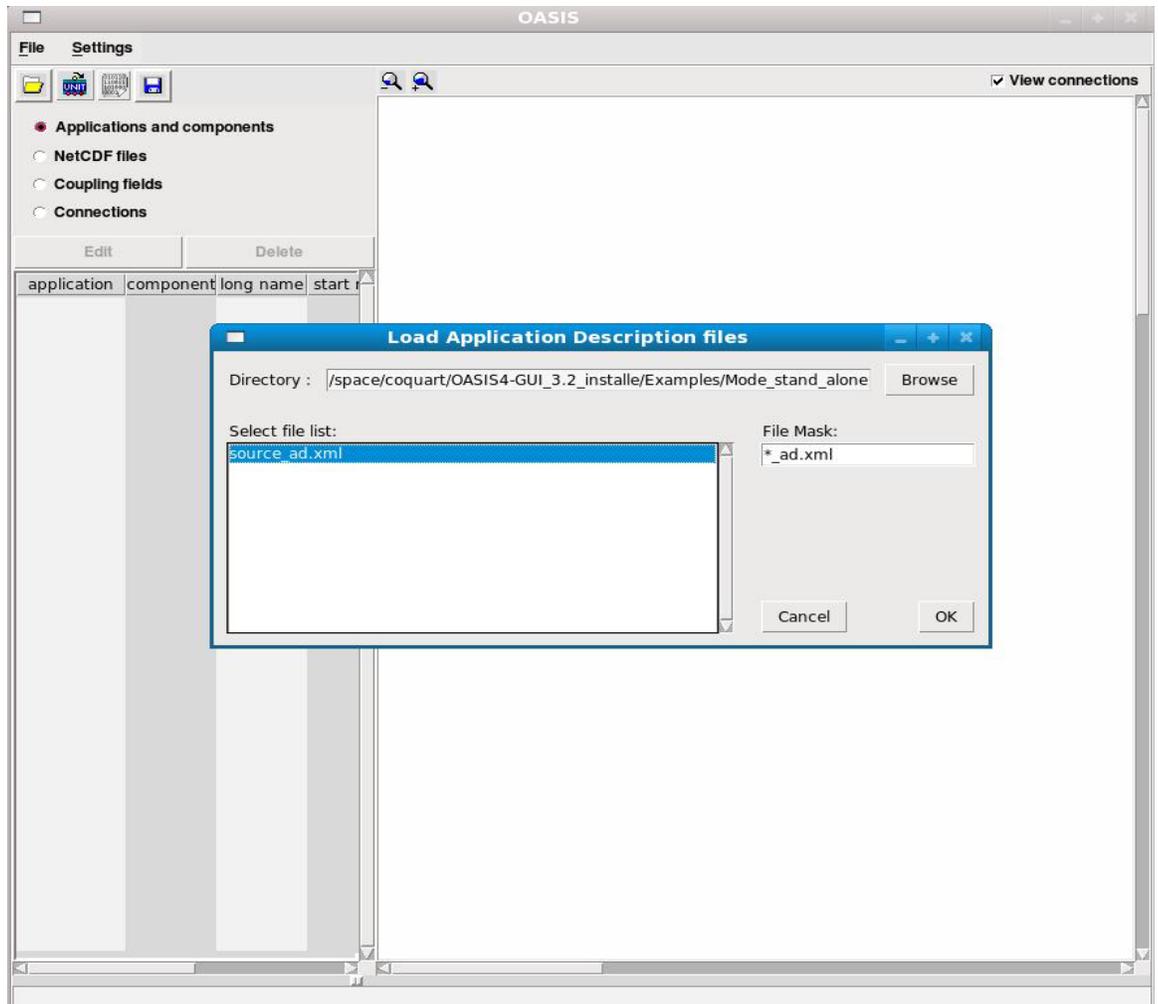


Figure 6-1: Loading the AD and PMIOD files for the application source

You have then to define the output NetCDF restart file where the component ocn will write its data, as plotted below on the *Figure 6-2* (see also the **section 5.7**):

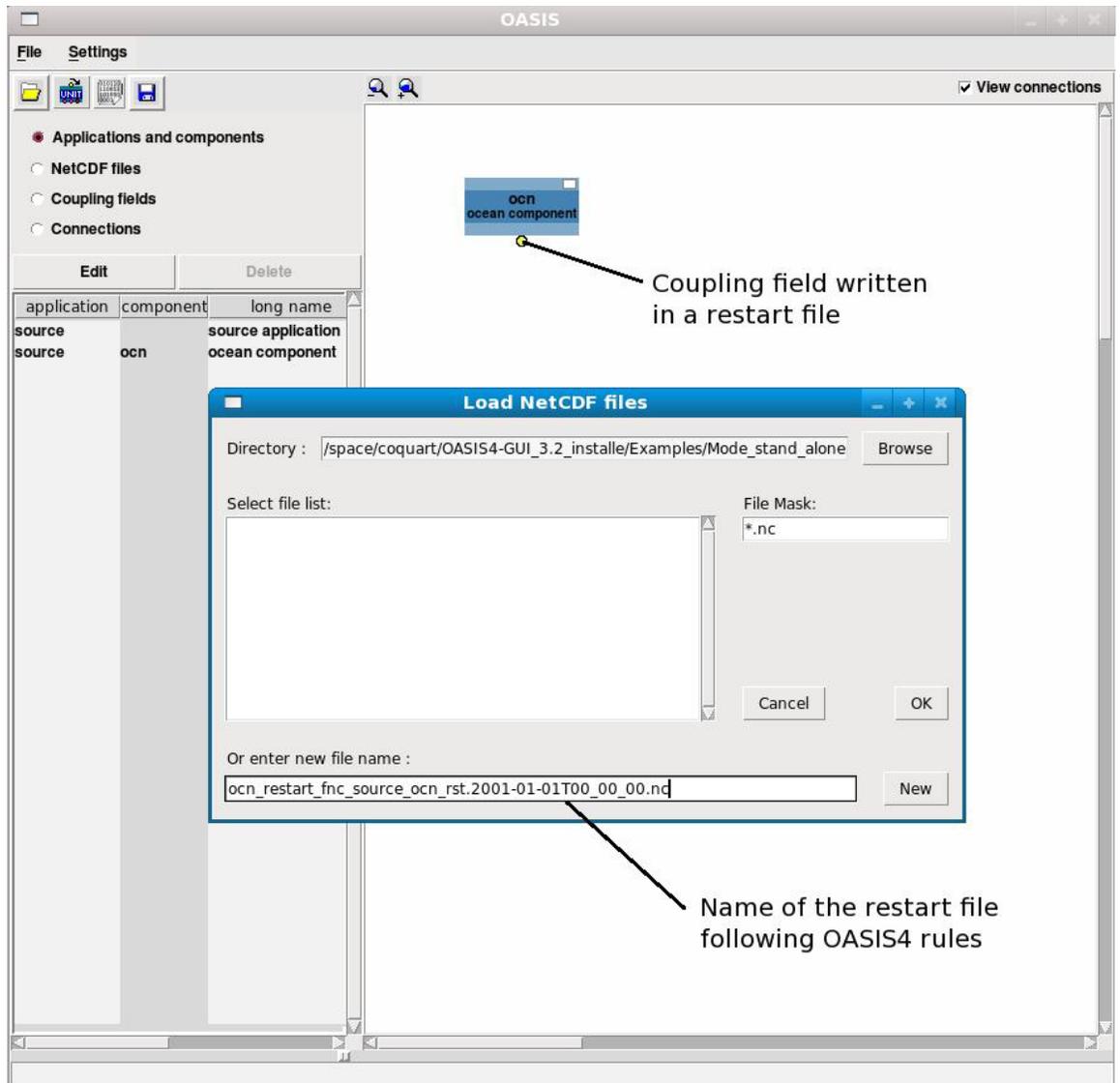


Figure 6-2: Defining the output restart file of the component ocn

When connecting the component ocn to the NetCDF file, you have to put the lag equal to zero, see the **Figure 6-3** below:

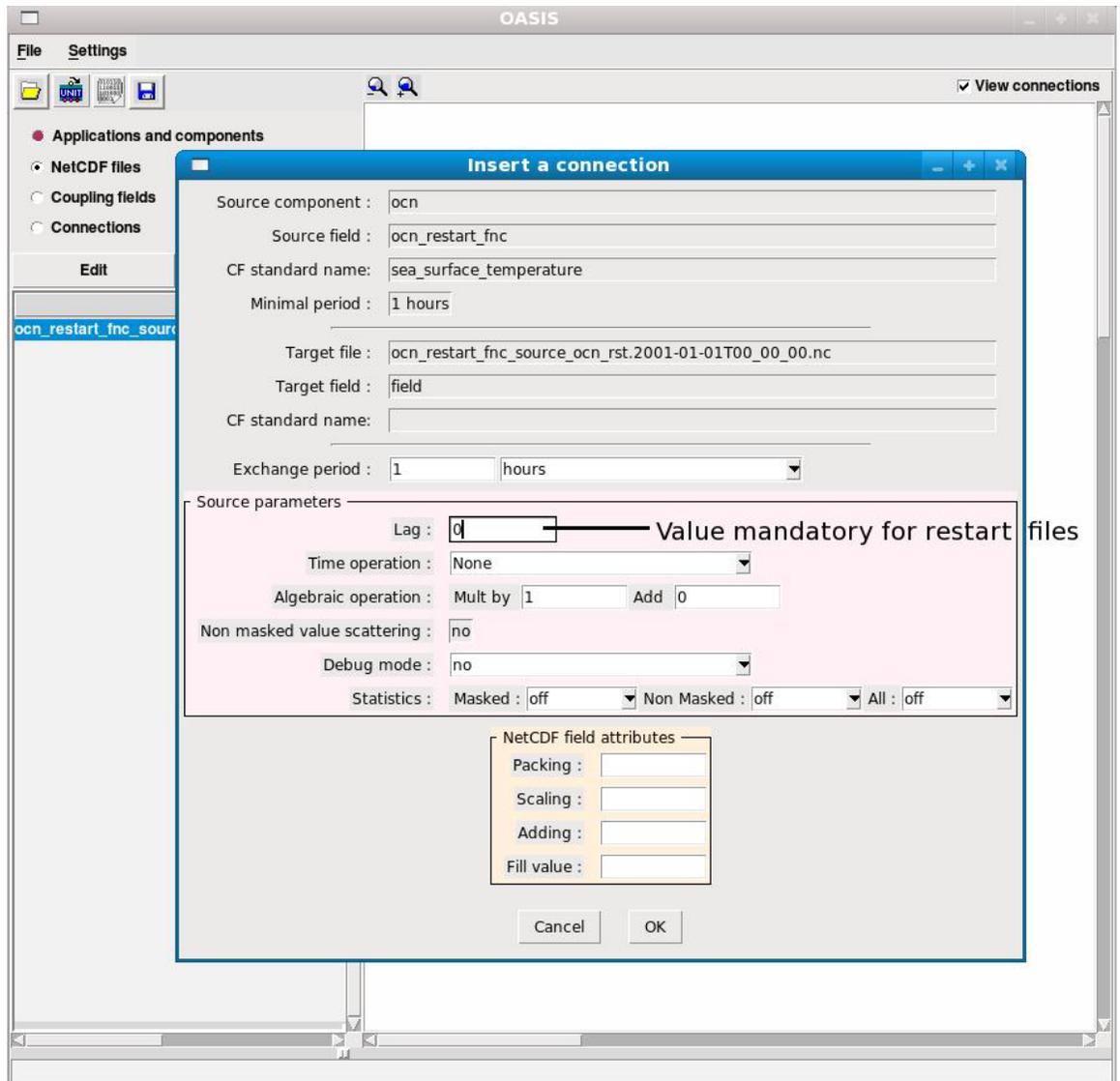


Figure 6-3: Connection between the component ocn and its restart file

When entering the Experiments settings, you must put the number of processors of the driver to zero and put **Run start date = Run end date = Experiment start date** (= Your Experiment start date). This date must also correspond to the one written in the name of the NetCDF restart file (see the *Figure 6-2*):

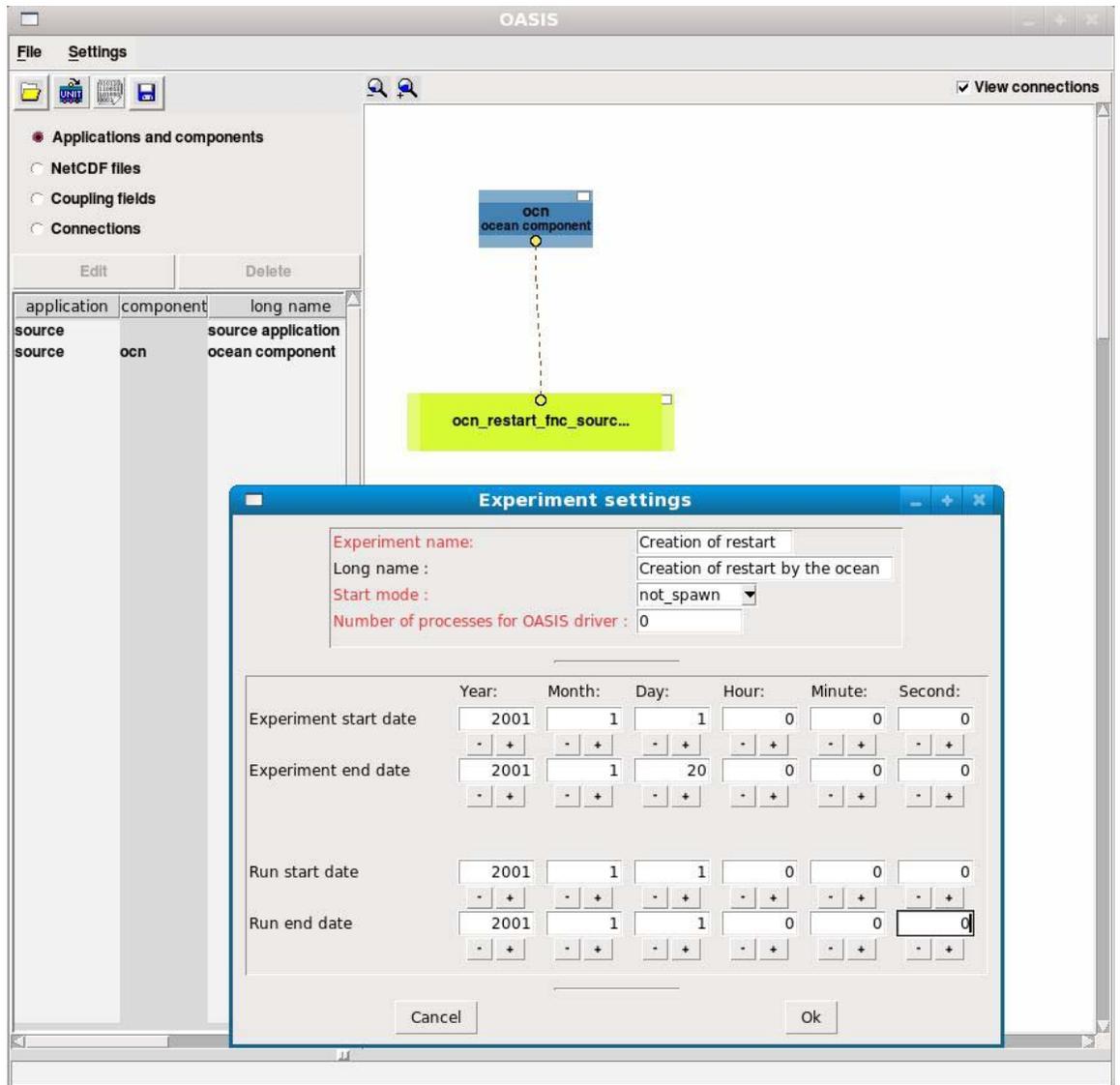


Figure 6-4: Experiments settings

You finally save the configuration as for a coupling case, except that you will have only one SMIOC XML file (see the **section 5.10**).

7. Acknowledgement

We thank Moritz Hanke (DKRZ), René Redler (Max-Planck-Institute) and Sophie Valcke (CERFACS), who tested the GUI and re-read this document, for their remarks and improvements.