

HOW TO USE THIS VERSION OF ORCHIDEE

The philosophy of the new version :

Dear User, thank you for being a tester of this new version of Orchidee, called the « externalised » version. It is based on the AR5 version. In this version, you can easily change the values of the parameters without having to recompile the code. Almost all parameters can be read directly from the file « orchidee.def ». These parameters can be scalar values. You can change one, two, three, etc or all the values from an array. You can also create new PFTs in order to have a better representation of the variability of the vegetation, specially useful for local studies.

The rest of the document is divided in five sections. The first section reminds you the specifications of « orchidee.def ». The second section explains how you can add a new pft or modify the values for the pfts parameters. The third section focuses on the manner you can control the values you imposed after your job. The fourth section gives you the restrictions and rules you have to follow for this version. The last section deals with some secondary aspects of the externalised version.

I. Global specifications :

The file « orchidee.def » contains the options and parameters that you can modify for any job. You will find it in the folder *config/ORCHIDEE_OL/Job_Name/PARAM* where Job_Name designates your own job . Each parameter externalised in ORCHIDEE is defined by a key word. This key word is written in capital letters in the « orchidee.def ». You will find some comments (lines beginning by #) for the parameters externalised in the AR5 version. For a scalar parameter, you should indicate the key word in capital letters and the new value.

For example, open « orchidee.def » in the folder :

.../modipsl/config/ORCHIDEE_OL/Job_Name/PARAM/

and set exactly :

| |
|-----------------------------|
| HCRIT_LITTER = 0.075 |
|-----------------------------|

(the standard value for this parameter is 0.08)

II. Specific requirements for PFTs parameters :

Unlike the other versions of ORCHIDEE, the number of PFTs is not fixed to 13 in the ORCHIDEE externalised version. You have the possibility to change the parameter representing the number of PFTs 'NVM'. You can easily modify the values of the parameters associated to each PFT (88 in this version). You know that the PFTs are described by a set of common parameters of which values were fixed in the previous versions of ORCHIDEE. In this ORCHIDEE externalised version, you can choose and change the values of parameters for each PFT. It is possible by using an object called metaclass (=MTC). Each PFT is associated to a MTC. Here we give the list of the MTCs :

| MTC name : |
|-------------------------------------|
| 1. bared ground |
| 2. tropical broad-leaved evergreen |
| 3. tropical broad-leaved raingreen |
| 4. temperate needleleaf evergreen |
| 5. temperate broad-leaved evergreen |

| |
|---------------------------------------|
| 6. temperate broad-leaved summergreen |
| 7. boreal needleleaf evergreen |
| 8. boreal broad-leaved summergreen |
| 9. boreal needleleaf summergreen |
| 10. C3 grass |
| 11. C4 grass |
| 12. C3 agriculture |
| 13. C4 agriculture |

As you notice, the list of the MTCs matches exactly the list of PFTs in the ORCHIDEE AR5 version. Each PFT is associated to one MTC, corresponding to the usual PFTs in the previous versions. Each MTC is described by the same parameters than the PFTs. The values of the MTC-parameters are fixed and are the default values of the system. By consequence, we implemented 13 MTCs in the code of the externalised version.

All you have to do is to associate a PFT to a MTC. When you link a PFT to a MTC, the PFT parameters will take automatically the default values given for the corresponding MTC. With this technique, you can modify the values and the parameters of interest for you. In practice, go the folder *modipsl/config/ORCHIDEE_OL/Job_Name/PARAM/* and open « orchidee.def ». In the first step, specify your number of PFTs (called NVM) like this :

NVM = 13

In the second step, you will declare the array PFT_TO_MTC. You have two ways for the declaration of arrays :

- 1) Either you can declare directly like this :

PFT_TO_MTC = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 , 11, 12 , 13

- 2) or component by component :

PFT_TO_MTC__01 = 1
PFT_TO_MTC__02 = 2
PFT_TO_MTC__03 = 3
PFT_TO_MTC__04 = 4
PFT_TO_MTC__05 = 5
PFT_TO_MTC__06= 6
PFT_TO_MTC__07 = 7
PFT_TO_MTC__08 = 8
PFT_TO_MTC__09 = 9
PFT_TO_MTC__10 = 10
PFT_TO_MTC__11 = 11
PFT_TO_MTC__12 = 12
PFT_TO_MTC__13 = 13

In both cases, you have to declare ALL the NVM values taken by **PFT_TO_MTC** you intend to use for your job. In this example, we associate each PFT to one MTC corresponding to the PFTs of the AR5 configuration of ORCHIDEE.

! **Syntax advice** : for the second method, there is a double underscore symbol between the name and the component of the vector (**PFT_TO_MTC__01**). « orchidee.def » recognize also the type of the parameter : for a real put **1.** (like in fortran), for an integer **1**, for a logical **TRUE** or **FALSE**.

Look at « sechiba_parameters_default_values.txt » and « stomate_parameters_default_values.txt » in order to know the type of the parameter.

Except **PFT_TO_MTC**, the other parameters are optional.

If you want ORCHIDEE not to read some values, put a comment by using the character # at the beginning of the line.

Examples :

1)Change the value for an already-existing PFT-parameter :

```
LAI_MAX__10 = 3.5
```

(the standard value is 2.5)

2)Create a new pft that differs only by one parameter value. For that, follow these two steps :

a) Associate your new PFT to one MTC using the array called **PFT_TO_MTC**. If you want your 14th PFT to possess the carateristics of the C3 grass , set exactly in « orchidee.def » :

```
NVM = 14
```

```
PFT_TO_MTC__14 = 10
```

(the 10th MTC is C3 grass according to the previous array)

At this point, your new pft will share automatically all the standard values taken by the parameters of the C3 grass.

b) Now, your new PFT is not exactly the C3 grass. It differs from the C3 grass by one value. For example, let's suppose that the **LAI_MAX** parameter for your PFT is different from the standard value. All you have to do is to set in the « orchidee.def »:

```
LAI_MAX__14 = 3.5
```

These two examples are not similar : in the first case, you change the **LAI_MAX** value from the default one for the PFT 10 (C3 grass). In the second case, you introduce a new specie of C3 grass (PFT 14).

You have just seen an example with one parameter. You are not limited to one : you can change the values for any given set of parameters. You can also change **NVM** in the other way ; you could set **NVM** less than 13 and considering only a given number of MTCs (4 or 5 for example).

Let's suppose you only want temperate vegetation and natural grass for your job. All you have to do is to set :

```
PFT_TO_MTC_01 = 1
PFT_TO_MTC_02 = 4
PFT_TO_MTC_03 = 5
PFT_TO_MTC_04 = 6
PFT_TO_MTC_05 = 10
PFT_TO_MTC_06 = 11
```

NB : Concerning the other arrays parameters

For the other arrays parameters, you have to set directly the new values. For example, let's consider the parameter called `albsoil_vis`. His key word is **ALBSOIL_VIS** and it is of dimension 9, corresponding to the dimensioning parameter for the soil color numbers and their albedo. Its standard values are :

albsoil_vis = (/0.18, 0.16, 0.16, 0.15, 0.12, 0.105, 0.09, 0.075, 0.25/)

If you want to change the third value, set in « orchidee.def » :

```
ALBSOIL_VIS__03 = 0.25
```

(for example). You could change the values of the third, sixth and ninth components. Just set :

```
ALBSOIL_VIS__03 = 0.25
ALBSOIL_VIS__06 = 0.120
ALBSOIL_VIS__09 = 0.27
```

This last example is available for all parameters-arrays (PFT or others), except **PFT_TO_MTC**.

Warning : all parameters depend on the different flags of ORCHIDEE like `river_routing`, `ok_co2`, `ok_stomate`, etc.... For example, the parameters used by stomate will be read only if the option `STOMATE_OK_STOMATE` is set to `TRUE` in « orchidee.def ». You will find the list of the flags and the corresponding parameters in « `sechiba_parameters_default_values.txt` » and « `stomate_parameters_default_values.txt` ».

III. How can you control your values after your job :

If you want to see if ORCHIDEE takes into account your modifications, you must go in a first step to the following folder : `...../IGCM_OUT/OL2/JOB_JobName/OOL/Debug/`

You will find three types of files (this is a my own example, replace `SECHSTOM` by the name of your job) :

```
SECHSTOM_19820101_19820131_out_orchidee_ol
SECHSTOM_19820101_19820131_run.def
SECHSTOM_19820101_19820131_used_run.def
.....
```

The file « `SECHSTOM_19820101_19820131_run.def` » is a copy of your initial « `orchidee.def` ».

In this one, you could find of course the values you imposed.

The file « SECHSTOM_19820101_19820131_used_run.def » contains the list of all the parameters read by the driver in « orchidee.def ». For the scalar parameters read by ORCHIDEE, you will find this comment for example in « SECHSTOM_19820101_19820131_used_run.def » :

```
# Values of LIMIT_NORTH comes from run.def
```

Thanks to that comment, you know that the **LIMIT_NORTH** parameter has been imposed by the user.

Unfortunately, in the case of the arrays, it will give you only the values which were default or imposed. For example, you could find :

```
# Values of LAI_MAX are all defaults.
```

or

```
# Values of ALBSOIL_VIS are a mix of run.def and defaults
```

but it won't tell you which value is default or not. Normally, it would be sufficient to know if ORCHIDEE has correctly read your « orchidee.def ».

The file « SECHSTOM_19820101_19820131_out_orchidee_ol » will list only the default values :

```
USING DEFAULTS : ALBSOIL_VIS__00001 = 0.1800000000000000  
USING DEFAULTS : ALBSOIL_VIS__00002 = 0.1600000000000000  
USING DEFAULTS : ALBSOIL_VIS__00004 = 0.1500000000000000  
USING DEFAULTS : ALBSOIL_VIS__00005 = 0.1200000000000000
```

You notice that **ALBSOIL_VIS__03** is missing. Comparing the files

« SECHSTOM_19820101_19820131_used_run.def » and

« SECHSTOM_19820101_19820131_run.def » to be absolutely sure to know whether ORCHIDEE has taken into account your new values.

IV. Some rules to respect for this version :

As the world is not perfect, this version is not perfect. There are few rules you have to respect for the moment (discussions are needed with the people involved to solve these restrictions) :

- **First rule** : you have **always to fill the array PFT_TO_MTC** in « orchidee.def ». It is very important because without it, the code won't be able to make the link between PFT and MTC. The dimension of PFT_TO_MTC is NVM. Even the parameter NVM is optional : if you don't set it, NVM will take the default value 13.
- **Second rule** : For the moment, the veget map can only deal with **NVM=13**. If you want to set a number of PFTs different to 13, you have to set the option **IMPOSE_VEG** in the « run.def » to TRUE.

- **Third rule** : the first PFT has to be associated to the first MTC (ie **PFT_TO_MTC__01 = 1**) corresponding to the bare soil. Moreover, you can't have two PFTs corresponding to the bare soil . To sum up, the only PFT associated to the bare soil (the first MTC) has to be the first PFT.

V. Other aspects :

- In order not to have a « big » file filled with 300 parameters and all their default values, the new « orchidee.def » file contains no default values. It lists only the description of the parameters and their key words. You find an example here :

```
# Vegetation distribution within the mesh (0-dim mode)
# If IMPOSE_VEG
# The fraction of vegetation is read from the restart file.
# If it is not found there we will use the values provided
here.
# SECHIBA_VEG
```

The goal is that « orchidee.def » will be filled only by the values the user imposed. You could easily see the new values of the parameters. (see « How can you control your values after your job »). You will find all the default values and their types by two different ways : there are files called « orchidee_run_parameters.txt », « sechiba_parameters_default_values.txt » and « stomate_parameters_default_values.txt » which list all the default values used in the code. This is the « user's » way. The second is the « developer's » way : look at the files called « constantes.f90 » and « constantes_mtc.f90 » in the src_parameter folder.

PS : The next objective will be to divide « orchidee.def » in three parts : « orchidee_run.def » which will contain all the global options for a job, « sechiba_parameters.def » and « stomate_parameters.def » «containing respectively the specific parameters for sechiba and for stomate.

- Only for the PFT parameters, you can use an option called **IMPOS_PARAM**. If you set it to FALSE, all the new values you put to the PFT parameters won't be read. If you make many changes for one job and you want to compare with the default values, you don't need to comment the values you impose. We think about the same mechanism for the other parameters (you could find all them in constantes.f90).
- Except for **PFT_TO_MTC** and a few exceptions, the values of the parameters you will set are not controlled. (We can't put 300 barriers) Read the different « .txt » files before making some changes.

Advice for developers only :

Our strategy was to centralize all the parameters in src_parameters. We think that this is better for the maintenance. If some people for their developpements want to add a new permanent metaclass or a new pft parameter (like Q10), they only have to modify the file constantes_mtc.f90 and add all the new values (don't forget to change the parameter 'nvmc' representing the number of MTCs).

If you find some bugs or have some comments or suggestions about this version or this tutorial, send an email to Didier.Solyga@lsce.ipsl.fr or open a ticket on the wiki : <http://forge.ipsl.jussieu.fr/orchidee/report>

