

Quick Start Tutorial PCRaster

Using the Windows Command Prompt

Version september 2009
Dr. Ir. WPA van Deursen
PCRaster Environmental Software
Carthago Consultancy
wvandeursen@carthago.nl
willem@pcraster.nl

@2003..2009 WPA van Deursen

Permission is granted to copy this document, provided that the entire document is copied (including this copyright statement) and that the document is not modified or altered. To modify or alter this document, or use individual parts of this document, written permission should be obtained from the author

Rotterdam, The Netherlands,
September 2009

Installation of PCRaster



This manual assumes you installed PCRaster and the Quickstart tutorial. I assume PCRaster is installed in c:\program files\pcraster and the Quickstart tutorial and dataset is in My Documents\PCRasterData\Quickstart.

Information about PCRaster and the associated tools can be accessed through www.pcraster.nl. The PCRaster main software can be downloaded from pcraster.geo.uu.nl/pcrwin32. Some new features are used in these Quickstart exercises that are currently not part of the default PCRaster package. These features should make it into the default PCRaster package fairly soon, but information on the Quickstart version can be obtained from Willem van Deursen (Email: willem@pcraster.nl or wvandeursen@carthago.nl).

You may need system administrator rights to install PCRaster. If you don't have these administrator rights, check with your system administrator or computer department.

This manual describes the use of PCRaster using the Windows Command Prompt. Since PCRaster is a scripting environment, a command prompt is the basic mechanism for controlling PCRaster. By default, the Windows operating system offers a system command prompt for this kind of purposes. Although this mechanism for issuing commands through a command prompt seems to be outmoded, it gives flexibility not available in the normal Windows Explorer.

Note that there are other ways to obtain a command prompt to PCRaster. The various windows command shells, such as Total Commander (www.ghisler.com) offer a useful environment. Furthermore, the Nutshell graphical interface to PCRaster is mentioned (<http://www.itc.nl/lisem/nutshell>). This site also hosts a map editor for PCRaster (<http://www.itc.nl/lisem/mapedit>)

Exercise 1

Finding your way in PCRaster

Evaluating ecosystem competition

This section is about the general principles of PCRaster. It teaches you how to recognise PCRaster maps, how to obtain information on them and how to open them for visualisation. An environmental model is executed, and the results are evaluated.

This first exercise uses a simulation model for heather ecosystems as an example. This simulation model analyses the competition between heather and grasses, and analyses the impact of eutrophication on this system.

Introducing the scene



In the Netherlands, many heathlands formerly dominated by *Calluna* have become dominated by grasses (*Deschampsia*). The environmental policy of the Netherlands regards this replacement of heather by grasses as unfavourable. It has been suggested that eutrophication of these heathlands may be a significant factor in the replacement of heather by grasses and field. The heather – grassland system is further influenced by the heather beetle or heather bug. During heather beetle outbreaks it is observed that *Calluna* is more severely affected by heather beetles in the more heavily fertilized vegetation. This suggests that the competition between heather and grasses is not simply due to the higher potential growth rate of the grasses, but may also be caused by a higher susceptibility of the fertilised *Calluna* to the heather beetle.



It is hypothesized that the outbreaks of the beetle are stimulated by nitrogen enrichment, and become more severe and occur more frequently under eutrophic conditions. No outbreaks are observed in young *Calluna* stands (< 5 years), and we assume that the canopy cover of *Calluna* must be more than approximately 50% before outbreaks occur.



During an outbreak of this beetle plague, *Calluna* plants die off almost completely over large area. The opening up of the dwarf shrub canopy enhances the growth of grasses due to an elimination of the competition from *Calluna*.

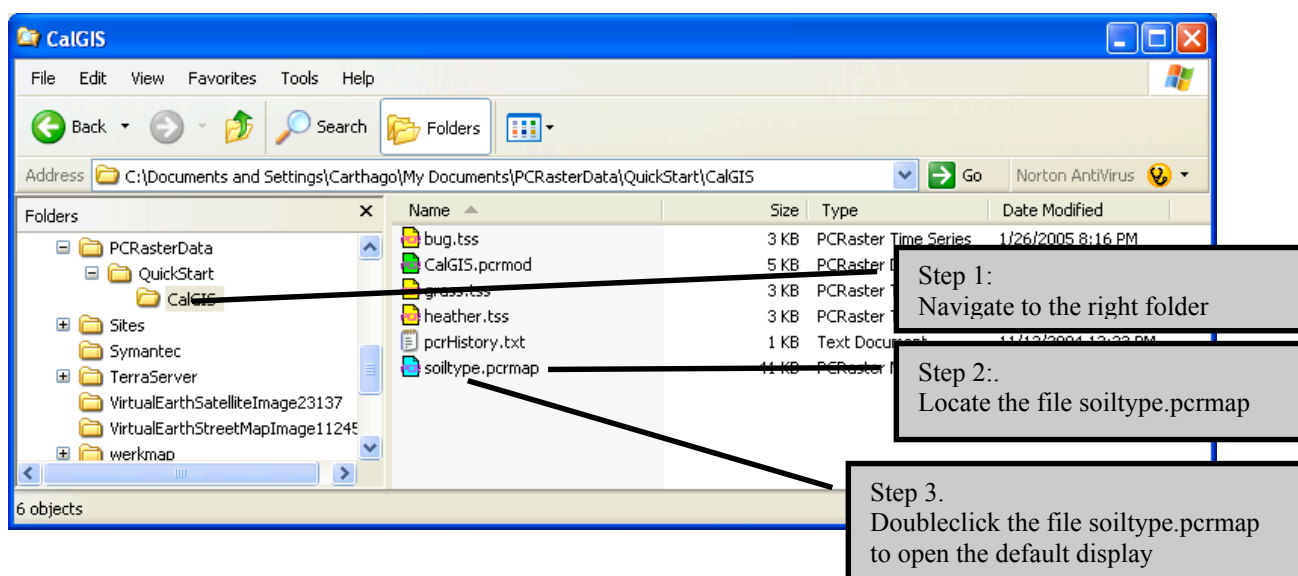
The CalGIS model contains a simplified process model to experiment with these processes. The CalGIS model is based on the vegetation research of Dr.

Gerrit Heil of the University of Utrecht. It has been one of the first models ever to be build in PCRaster. The pictures included are also courtesy of Dr Gerrit Heil. More information can be found in Van Deursen W.P.A. and G.W. Heil, 1994: Analysis of heathland dynamics using a spatial distributed GIS model. Scripta Geobotanica 21: 17-27; or on www.bio.uu.nl/~boev/staff/personal/gwheil/gwheil.html

CalGIS (Calluna GIS model) simulates the heather-grasses competition in the Netherlands using the soil map as the basic input map. CalGIS very optimistically assumes a sandy soils in the Netherlands to be potential heather habitat. This version does not include the quite substantial disturbances on this habitat in such a densely populated region as the Netherlands.

Exploring the CalGIS model

Start Windows Explorer and navigate to the folder with the QuickStart exercises (usually 'My Documents\PCRasterData\QuickStart\CalGIS') and open the folder CalGIS. In this folder there should be a file called soiltype.pcrmap. Doubleclick on this file to open the standard display for this map.



This will start the display-application which is used for displaying PCRaster map files. The map shown is a generalised soil type map of the Netherlands. For the demo, it is assumed that all sandy soils in the Netherlands are potential candidates for the heather and grass competition.

Close the display-window and locate the CalGIS.pcrmod file. This file contains the model script for the CalGIS model. Open this file by double-clicking the CalGIS.pcrmod file. This will start the default editor for PCRaster scripts.

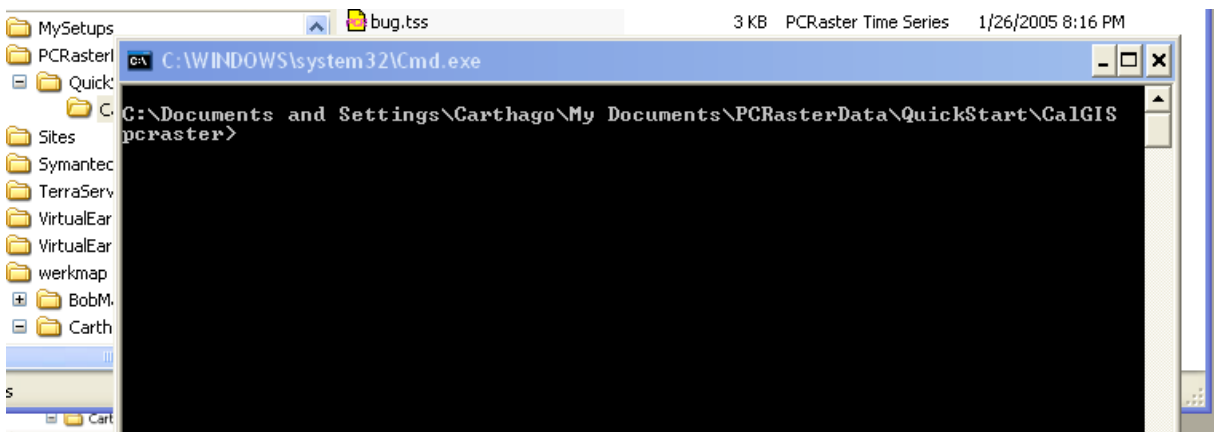
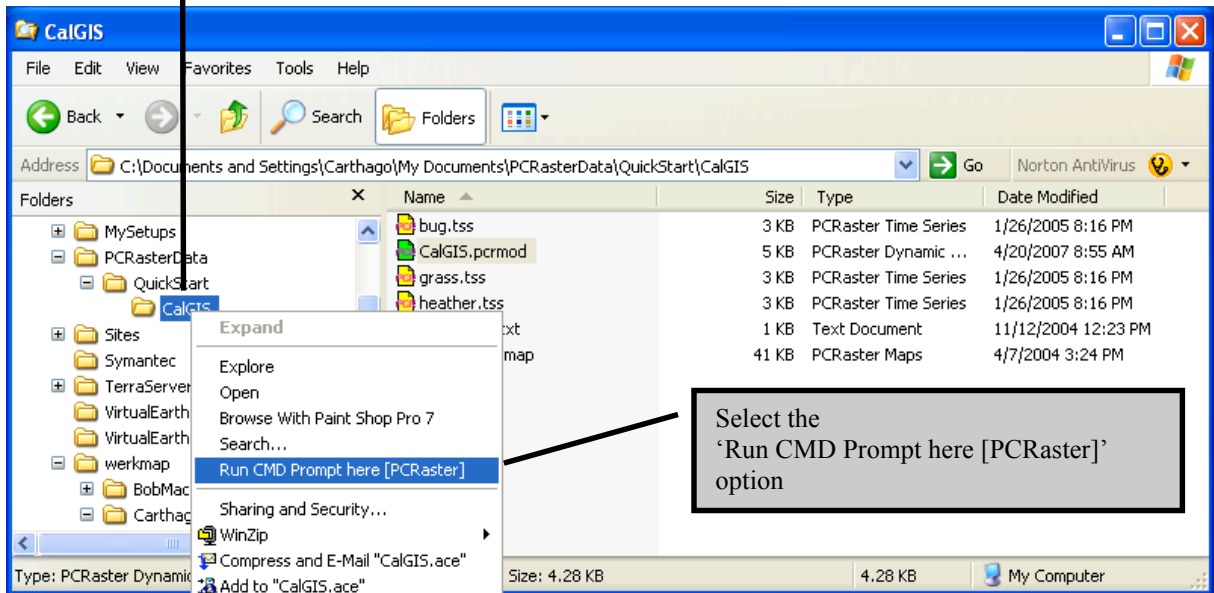
The first part of the script contains remarks and a short explanation of the model. All lines which start with an # are comment lines, meant for the users of the script and serving as documentation and explanation of the script. The lines not starting with an # contain commands for PCRaster. As you can see, even the PCRaster commands can be read (and understood) by humans.

Close the editor.

Start the Windows Command prompt by rightclicking the folder with your demodata (usually 'My Documents\PCRasterData\QuickStart\CalGIS') in the Folders Pane (left pane) of Windows Explorer. This will open a command window (also known as a CMD box) for the current folder.

Rightclick the CalGIS folder

Select the 'Run CMD Prompt here [PCRaster]' option



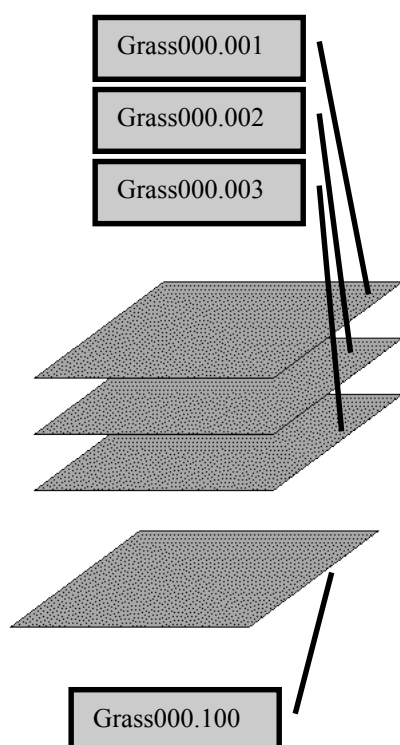
Run the script by entering the command

```
pcrcalc -f CalGIS.pcrmod<Return>
```

in the command window. Note that you do not type the phrase "<Return>", but this is to indicate that you have to press the Return key or the Enter key on your keyboard. This should run the model.

You will see PCRCalc in the Command Window counting to 100 and then finishing. If PCRCalc didn't complain about any errors, you just finished your first PCRaster run. Congratulations!

Leave the Command Window open during this entire exercise. You will switch quite a few times between the Command Window and Windows Explorer. Your folder (shown in Windows Explorer) should now contain quite a few more files. These files are the results of the CalGIS model run. As you can see, quite a few files consist of a name combined with a number. In the CalGIS folder there is a whole bunch of files called grass000.001, grass000.002, .. up to grass000.100. This entire set of files is called a dynamic map stack or, for short, a map stack. We will refer to this dynamic map stack as the Grass dynamic map stack. Combined these files contain the dynamic behaviour of the grass variable of the CalGIS model. Similarly, there is a heather dynamic map stack and a beetle dynamic map stack, named heather0.001 up to heather0.100 and bug00000.001 up to bug00000.100.



Type

```
aguila -2 grass000.001<Return>
```

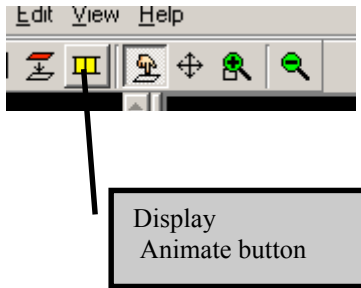
in your command window. This opens the file named grass000.001 in the aguila module. Aguila is the main visualisation module of the PCRaster package. This map shows the status of the grass variable for the first timestep. Leave this aguila window open and type

```
aguila -2 grass000.100<Return>
```

in your command window. This shows the status of the grass variable at timestep 100. As you can see, the grass has grown substantially in these 100 years. Close the two aguila windows.

Type in the command window (no spaces around the + sign!):

```
aguila -2 grass000.001+100<Return>
```



This will load the aguilá program with the entire Grass dynamic map stack. Now press the 'Animate map stack' button on the aguilá interface. Aguilá will now show you the contents of the entire Grass dynamic map stack. Close all aguilá windows and try to display the heather dynamic map stack (heather0.001 up to heather0.100) and the beetle dynamic map stack (bug00000.001 up to bug00000.100).

(Note the "-2" on the aguilá command line. This forces aguilá to open in a 2D mapview. Aguilá offers also a 3D map view. In earlier versions of aguilá, the 3D mapview was default, in later versions the 2D map view is default. For the default view, you can omit the "-2" or "-3". See the aguilá manual for more information)

You can also display the entire stack by double-clicking on the first stack item in your windows explorer. Double-click the file grass000.001. The entire stack is loaded in the aguilá program, and by clicking the Animate button you see the animation of the entire stack.

Animating dynamic map stacks in the aguilá map display is one way of analysing the results of your simulation. Other ways include producing timeplots of your runs. Locate and select the file heather.pcartss (timeseries of the dynamic behaviour of heather). Now type in the command

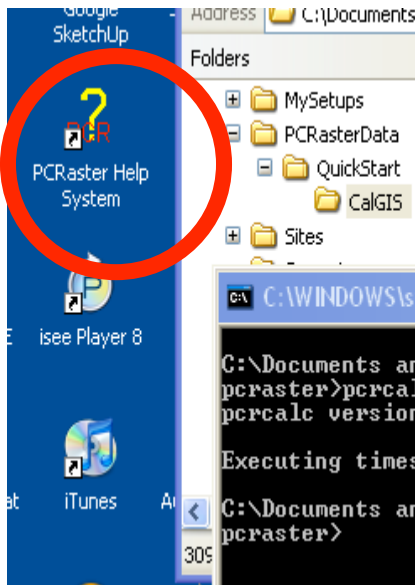
```
timeplot heather.pcartss<Return>
```

This will execute the timeplot module for visualisation of timeseries, showing the spatially averaged development of the heather over the 100 timesteps. Notice that after an initially very favourable start the growth of the heather population is severely limited by the beetle and the competition with the grasses.

Analyse also the timeplots of bug.pcartss and of grass.pcartss. Close all aguilá windows after finishing the analysis.

Using PCRCalc from the command window

Getting help for PCRaster modules and the PCRCalc functions



PCRaster includes a number of modules for which help is available. You have already used the display-module for visualising the maps. The display module offers much more functionality and features than covered in this Quickstart tutorial. More information about the display-module can be obtained through the PCRaster Help shortcut that was created during the installation of PCRaster. This shortcut directs you to the help system of PCRaster.

Doubleclick the 'PCRaster Help System' shortcut. In the 'PCRaster Visualisation Commands' table, click the 'Yes' hyperlink for the manual of aguila. This will direct you to the aguila manual. Read the help pages for aguila carefully. Open a aguila window for the file soiltype.pcrmap and try some of the more advanced functions of aguila, such as zooming.

The core of the PCRaster package is the PCRCalc module. This is your main raster map calculations engine. Essentially it provides a large number of map functions and map operators. These functions and operators take maps as their input and return maps as their results. Examples are the operators +, -, * and /, which add two maps together, subtract a map from another map, multiply two maps or divide two maps. PCRCalc map functions include ln(), log10(), max(), min() and a whole range of other functions.

The functionality of PCRCalc can be directly accessed through the command window. Type:

```
pcrcalc diffheather.pcrmap = heather0.100 - heather0.001<Return>
```

In normal words, this means: use the pcrcalc command to calculate the difference between the heather at timestep 100 and the heather at timestep 1. Put the result in the file diffheather.pcrmap. Note that diffheather is a short name for difference in heather coverage. Read the PCRCalc messages and type:

```
aguila diffheather.pcrmap<Return>
```

In normal words: use the aguila command to show the file we just created: the file diffheather.pcrmap. The first command uses PCRCalc as a mapcalculator. Results of these calculations are written to your hard disk. In

this case a file named 'diffheather.pcrmap' was created as the mathematical difference between the cells in the map heather0.100 and heather0.001.

It is a good habit to maintain some structure in naming these files. Although PCRaster or PCRCalc doesn't enforce it, a good habit is to give each map an explanatory filename and use a file extension .pcrmap (dot pcrmap). Most of the time you are not responsible for the filename-extensions of the dynamic stack maps, and PCRaster names them with the extensions .001, .002, .003 etc. Timeseries should be named with a .pctss extension and models should have a pcrmod extension. If you maintain these conventions, it will be easier to maintain your PCRaster files.

Now try the command (note the intentional error in the second mapname: the file heather0.pp1 probably does not exist on your system)

```
pcrcalc differror.pcrmap = heather0.100 - heather0.pp1<Return>
```

PCRCalc issues an error message, and the resulting map is not created.

PCRaster uses what is called a data-typing mechanism for its maps. This means that PCRaster stores information in the map on whether the values in the map should be interpreted as numerical values (scalar values), true-or-false values (boolean values), classvalues (nominal or ordinal values) or special values (ldd values, discussed in the next exercise). PCRaster enforces strict type-checking on your maps, meaning that it doesn't allow you to use class values as if they were numerical values. If you try to add a numerical value (scalar value) to your soil map (nominal values), PCRaster gives you an error message, telling you this is not a valid operation on a soilmap. Try the command

```
pcrcalc myresult.pcrmap = soiltype.pcrmap + 10<Return>
```

Notice the error message produced by PCRCalc, and also notice that the map myresult.pcrmap is not created. In general, this type checking mechanism is transparent to the user, and if PCRCalc complains about data types that are incorrect, it most of the time means the logic in your analysis is not correct. In emergencies, there are the functions scalar(), nominal(), boolean() etc. that will overrule the types associated with the maps. Try

```
pcrcalc myresult.pcrmap = scalar(soiltype.pcrmap) + 10<Return>
```

and see that PCRCalc now accepted this command. However, it becomes very difficult to try to interpret the resulting map.

PCRCalc functions and operators can be combined, that is, we can nest operators and functions to perform complex operations. As an example from the CalGIS model, the following statements are valid PCRCalc statements within a model script (only examples, not meant to be executed here!)

```
h_growth= (0.005*nutrient*heather+0.22*heather)*(100-grass-heather)/100;
h_death = 0.12*heather;
heather = min(max(heather+h_growth-h_death,0),100);
```

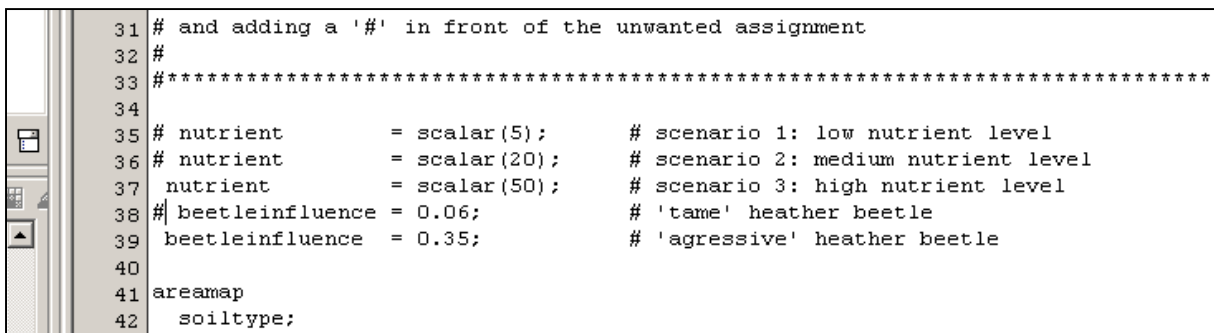
You can access the Help for the individual functions through the PCRaster help shortcut and clicking on the 'list of PCRCalc functions'.

Analysis of different heather beetle characteristics and eutrophication scenario's

Reanalyse the bug pattern that is simulated by the current version of the model by displaying the bug dynamic map stack. Notice the distribution of the bugs through the landscape, once the bugs have affected a certain location.

Now, open the script for the heather-grass-beetle model again by selecting the script CalGIS.pcrmod and double-clicking the file. If you read through the first part of the model you will see some lines that are valid PCRCalc statements but that do have a # (comment sign) in front of them. This means they are treated by PCRCalc as comments and that they are not executed.

Edit line 38 and 39 of your model: add the '#' comment sign in front of line 38 ('comment away' line 38) and remove the '#' of line 39 ('activate' line 39).



```
31 # and adding a '#' in front of the unwanted assignment
32 #
33 #*****
34
35 # nutrient      = scalar(5);      # scenario 1: low nutrient level
36 # nutrient      = scalar(20);     # scenario 2: medium nutrient level
37 nutrient        = scalar(50);     # scenario 3: high nutrient level
38 # beetleinfluence = 0.06;          # 'tame' heather beetle
39 beetleinfluence = 0.35;           # 'agressive' heather beetle
40
41 areamap
42 soiltype;
```

Now save the model and run it by entering the appropriate command from the command window (The appropriate command to run a model is:

```
pcrcalc -f modelname,
```

in this case

```
pcrcalc -f CalGIS.pcrmod. )
```

If PCRCalc doesn't return any error messages, you now finished simulating the competition between the heather, the grasses and the beetle with a much more aggressive beetle. Watch the beetle taking over by the command:

```
aguila bug00000.001+100
```

and

```
aguila bug.pcrtss
```

Analyse the animation and the timeplots resulting from this simulation.

Try to run the simulation for different eutrophication scenarios by selecting another level for the nutrient-variable. If you are finished with this exercise, close your CMD window.

A few final notes on the use of the Windows Command prompt. The CMD window was initially started in a specific folder of your file system. This folder is the default folder for the CMD window, and any command you issue through the CMD window is executed in this folder. If you change folders, because you are moving to another project, another dataset or whatever reason, your CMD window is NOT AUTOMATICALLY referring to the new folder. In these situations, it is advised to close the old CMD window, and use the techniques described above to open a new CMD window at the new location. People familiar with DOS can also use the cd-command to move to the new folder.

A final warning on the use PCRCalc. As you might have noticed, PCRCalc overwrites existing files without any warning. You better get used to this behaviour, and carefully operate PCRCalc. Thoughtless use of PCRCalc will eventually result in overwriting important files you created earlier.

Although initially this behaviour of PCRCalc seems awkward, but this is a deliberate design of PCRCalc. Creating dynamic simulation models (the main

purpose of PCRaster) is a process in which you are constantly overwriting files and results of earlier experiments. The 'awkward behaviour' of overwriting existing files is actually a very powerful mechanism to become very productive. The user is in full control and PCRCalc executes anything the user wants. But the user better knows what he wants, and be aware of this.

For more Quickstart Exercises: contact Willem van Deursen at willem@pcraster.nl or wvandeursen@carthago.nl