Daisy Program Reference Manual

Per Abrahamsen <abraham@dina.kvl.dk>

February 15, 2011

University of Copenhagen
Department of Basic Sciences and Environment
Environmental Chemistry and Physics
Thorvaldsensvej 40
DK-1871 Frederiksberg C Tel: +45 353 32300
Fax: +45 353 32398

Contents

Part I

The Simulation

Chapter 1

Introduction

Welcome to the Daisy Program Reference Manual. This manual provides you with a reference to the Daisy software. That is, it describes the command line version of Daisy, and the format of the setup files. As a start, let's look at what this manual isn't.

- It is not an introduction to the Daisy simulation model itself. See [Hansen, 2002], [Hansen et al., 1990] and [Hansen et al., 1991] instead.
- It is not a reference manual for the mathematical or physical aspects of the various models in Daisy. The models are listed together with their parameters, but not described in any detail. Some models are explained in [Hansen, 2002], [Hansen et al., 1990] and [Hansen et al., 1991].
- It is not a tutorial for the Daisy software. The manual is intended as a reference for people who already know the system. A tutorial can be found in [Abrahamsen, 2003].
- It is not a programmers manual. See [Abrahamsen, 1997] for how to add new models to Daisy, or [Abrahamsen, 1998] for a description of how to add Daisy to larger systems. An overview of the program design can be found in [Abrahamsen and Hansen, 2000].

Enough negative statements, what this manual *does* provide is a complete reference to all components and models provided by Daisy at the time the manual was generated, as well as all the parameters, state variables and log variables supported by these models. These are collected in part II of the manual. The reason why we can state with absolute confidence that the listing is complete, is that it is generated automatically from the code. Which means, if Daisy accepts a parameter in the setup files, it will also be listed in the manual. It also means that the components, models and parameters cannot be organized into logical groups with helpful descriptions of each. Instead they are presented in alphabetical order. This manual is generated from the version of Daisy listed at page 449.

Part I of the manual is written by humans though. Here, we will try to describe enough of the concepts, format, and nomenclature to be able to use part II efficiently.

1.1 Concepts

In order to use the Daisy model, as well as this manual, you must already possess a fairly good understanding of the agronomical and physical processes that are modelled by Daisy. However, when describing the organizing and principles behind the

Daisy software, we have to introduce some new concepts, which are used throughout this manual. This section is somewhat abstract and dry. The best way to read it might be to read it once, without worrying too much about understanding it, then get some experience with the Daisy program and the setup files, and then read this section again.

1.1.1 Components, Models, and Parameterizations

The two most significant concepts behind the organization of the Daisy, are *components* and *models*. Typically, a component corresponds to some physical process, and a model is a particular way to describe (or simulate) the process. For example, water transport in the unsaturated zone is a process, while Richards Equation is a particular way to describe that process. In Daisy, water transport is a component (described in chapter 77), while Richards Equation is a model (described in section 77.3).

Just about everything in Daisy are either components or models, not just what you naturally think of as physical processes. For example, crops (chapter 22), management (chapter 9), and even log files (chapter 39) are all components, each having several different models. Which leads us to the next issue, *fixed* vs. *library* components.

The general component/model framework makes it very easy to add new models to existing Daisy components. All that is required is a C++ file with an implementation of the model. No other code has to be changed. To activate the model, the user just selects it in the setup file. However, for some components we only support a single model. This makes them slightly simpler, for example the user will not have to specify a model in the setup file, as there can only be one. The cost is less flexibility. We call the first kind of components, which can (theoretically) can have many different models, for *library components*. Each library component has its own chapter in part II, with a section for each model of that component. The second less flexible kind of components is called *fixed components*. All the fixed components are described in chapter 85.

Parameterizations

Library components have another important feature, which fixed components lack. They allow the user to save standard parameterizations of the model in text files, and refer to these parameterizations by name. Daisy is distributed with a large number of such standard parameterizations. The most obvious example where they are useful is parameterization of crop models. The default crop model (22.1) is sufficiently generic to be able to describe crops ranging from grass to potatoes, but it takes lot of parameters to do so. This is why Daisy comes with already parameterized versions of the model, with names like “Grass” and “Potato”.

All in all, this gives a three layered hierarchy, with library components in top. For each library component, there can be one or more models implemented in C++. For each model, there can be one or more named parameterizations.

1.1.2 Model Attributes

A parameterization can set *parameters* (as the name implies) for the model, as well as initial values for the models *state variables*. The parameters are constant during the simulation, while state variables can change as the simulation progresses. A typical parameter could be the hydraulic conductivity for saturated soil, or the maximum height for a crop. Typical state variables could then be the water content in the soil, or the current height of a crop.

Optional and Default Parameters

In general, all the parameters and state variables must be specified before a given model can be used in the simulation. There are two exceptions to that rule though.

First, some parameters and state variables are marked *optional*. This can mean one of two things. Sometimes the model will be able to infer the value of one variable (or parameter) from the values of other variables or parameters. For example, the 'default' chemical model (section 17.1) has both an 'M' state variable, indicating the total chemical content in the soil, and a 'C' state variable, indicating the chemical concentration in the soil water. Given one of these, and information about the amount of water in the soil, the 'default' chemical component is able to calculate the other.

The other exception is parameters or state variables which have a *default value*. Many models have parameters that the typical user will rarely change, because the default value is well established, but where the parameter is still provided, for those people who want to fiddle with the model. For state variables, there can also be reasonable values provided by default. For example the 'crops' vegetation model (section 79.1) will by default start with an empty list of crops, i.e. there will be no crops in the field at the start of the simulation. This is what most people will expect.

Partial Parameterizations

One can say that the model itself can provide a *partial parameterization* of itself this way, i.e. a parameterization of some, but not necessarily all, parameters in the model. The user can also create partial parameterizations, which is a third example where you don't need to specify all the parameters and state variables. A partial parameterization cannot be used directly in the simulation without specifying the missing parameters and state variables, but can be used as a *base* for other *specialized* parameterizations. For example, the standard fertilizer library in the Daisy distribution includes a partial parameterization named 'slurry'. This parameterization is used as the base for two specialized parameterizations, namely 'pig_slurry' and 'cattle_slurry', and contains the parameters that are common for both kinds of slurry. With partial parameterizations, the user can build a hierarchy of parameterizations with the most general on top, and the most specialized (complete) parameterizations at the bottom.

Log Variables

A model can also support a second kind of variables, namely *log variables*. Log variables are computed during the simulation, but are not part of the model state, i.e. they are not stored between time steps. This also means that it makes no sense to initialize them, so they cannot appear in a parameterization of the model. However, log variables and state variables can be written to output files (or other external media) by the various 'log' components (chapter 39). Thus, we have parameters and state variables in the input files, and state variables and log variables in the output files.

In the Daisy vocabulary, parameters, state variables, and log variables are three different *categories* of model *attributes*. One can think of the attributes in general as the users mean to interact with the model, and each of the three categories as specifications of how the particular attribute can be used, for input, for output, or for both.

1.1.3 Attribute Types

Beside having a category, attributes also have a *type* and a *size*.

A type is the set of legal values. For example, an attribute whose type is 'integer' (see section 1.4.3) cannot hold a string (see section 1.4.5) value. This prevents you from setting the attribute '`NoOfIntervals`' to '`"Gnat"`' in the input file. For log variables, the type will give you an idea of what kind of values to expect in the output files. The various types are described in section 1.4.

The attribute size specifies the number of values to expect. If the attribute is a *singleton*, only a single value can appear. If the attribute is a *sequence*, any number of values can appear. The attribute can also have an integer size, which means exactly that number of values should appear.

Each attribute has an associated description, which hopefully will give some idea of what the attribute is used for. Attributes with the type 'number' (see section 1.4.4) will also have a dimension, so you for example will know whether the length should be given in millimeter or kilometer. This can make a difference. Trust me on that one.

Ordered Attributes

Finally, some attributes may be *ordered* within the model. This order is used by the 'file' parser model (section 47.1) for parsing the setup files. Normally, you will have to specify both the name and the value of the attributes in the setup files. However, ordered attributes are identified by the order they appear in the setup file, so you will not have to specify their names. In fact, you are not allowed to specify their names. Ordered attributes are typically used for "obvious" parameters, for example the 'file' parser model has a single parameter, 'where', the name of the file to parse. That parameter is ordered, so you can write

```
(input file "filename.dai")          ; Correct.
```

instead of

```
(input file (where "filename.dai")) ; Wrong.
```

as you would have had to write, if the 'where' attribute had been unordered.

1.2 Nomenclature

Having covered the concepts, it is now possible to explain the nomenclature used in part II of the manual. As already explained, each library component has its own chapter, while all the fixed components are collected in chapter 85. All the library component chapters start with a short explanation of the purpose of that component.

Each model (including those models that are part of a fixed component) has its own section. Here, we will explain how to read the content of these sections. This is best done with an example. Now, keep a finger here, and find section 9.48 (that's on page 83). Section 9.48 is a description of the 'fertilize' model, which is part of the action component. You specify this model when you want the manager to fertilize the field.

1.2.1 Name and Description

The name of the section is 'fertilize', which is the name you have to specify in the setup file when you want to refer to the model. The text in the start of the section is a brief description of the model and what it does.

1.2.2 Sample

The third element in the section is a *sample*, which shows the format recognized by the file parser model when it reads the setup files. The sample starts with '<' and ends with '>', these should not be included in the setup files, they are only there to show the start and end of the sample. The sample text written in a normal typewriter should be included directly in the setup files, while the text written in slanted text should be replaced with the parameter (or state variable) values, as listed after the sample.

All parameters and state variables will be listed in the sample. Log variables will not be listed, since they cannot be set in input files.

Ordered attributes will be listed first, in order, since that is how they must appear in the setup files. In the example (that's section 9.48, your finger should still be there!), *am* is such an ordered attribute. You will note that the text *am* only occurs once in the sample, while the texts *to* and *from* both occur twice. This is because you only have to specify the value for the 'am' component in the setup files, since it is ordered, while you have to specify both the name and value for the unordered 'to' and 'from' attributes.

For the unordered attributes, it doesn't matter what order they appear in the setup files. In the sample, they will be shown in alphabetic order, 'from' will appear before 'to'. However, there is no reason (except good taste) why you shouldn't set the 'to' parameter before the 'from' parameter in the setup files.

All three attributes ('am', 'from', and 'to') are singletons (reread section 1.1.3 if you have forgotten what that means). If any of the attributes had been sequences, they would have been followed by the string '...'. This does not mean you should write '...' in the setup files, but that you can write any number of values at that point, separated by whitespace. Section 9.20 has an example of that.

If an unordered attribute has a simple value, it will be printed directly in the sample.

1.2.3 The Attribute List

After the sample comes a list with information about each attribute in the model. The parameters and state variables are listed in the same order as in the sample, that is, first the ordered attributes (in the specified order), and then the rest of the parameters and state variables in alphabetic order. After that, the log variables are listed, if any, in their own section. The 'fertilize' management action model doesn't have any log variables, so look at the default bioclimate model (section 13.1) instead. It has plenty.

The documentation for each attribute is divided in three parts, which are explained below.

The Type Line

The first line (the line with the bullet) contains the name and the type (section 1.1.3, remember?) of the attribute. The name is written in slanted italic, to match the way it is written in the sample (section 1.2.2). Two types are treated specially: Numbers have a dimension, which is written in bold in square brackets; Components have their name written in bold, and a reference to the chapter (or section, for fixed components) in the manual where they are described. Other types (such as integers and strings) have a reference to the subsection of section 1.4 where they are explained.

The Category Line

The second line contains the following information:

1. Whether the attribute is optional. If so, the line will start with the string 'Optional'.

2. The category of the attribute, that is, whether it is a parameter or a state variable. The category line is not included for log variables.

3. The default value, if any. If the value is something simple (like a number or a very short string), the value will be shown on the line. Otherwise, the manual will just state that the attribute has a default value, but not what it is.

Sometimes the category line is left out. It is never included for log variables, since their category it is obvious from context, and they never have default values. The category line is also left out for components, unless the component is optional or has a default value. This is because components can contain both state and log variables, as well as parameters. So the category of a component is not a very useful concept. If a component does a default value or is optional, the category line will be printed. However, instead of 'parameter' or 'state variable', the attribute will be categorized as a 'component'.

The Textual Description

On the third line (or second, if the category line is missing) there is a brief textual description of the attribute. The description may continue on the following lines.

1.2.4 Submodels

Sometimes it is useful to combine attributes in a model into groups. One example of this is the 'cond' management action model (section 9.20), where we use the 'clauses' group to emphasize the strong association between each condition, and the list of actions to perform when the condition is true. Another example is the 'default' crop model (section 22.1), where there simply are so many attributes that grouping them logically becomes a necessity.

If you have looked at the two examples, you will see the attribute groups are categorized and typed as 'submodel' in the manual. After the textual description of the submodel follows a sample illustrating the formatting of the submodel attributes. The submodel sample follows the same conventions as the model sample, described in section 1.2.2. After the sample follows a nested attribute list, documenting attributes of the submodel in the exact same way as the attributes of the main model are documented, as described in section 1.2.3. You can even have submodels nested inside submodels, ad infinitum.

1.3 The Daisy Model and Named Parameterizations

Try to look at section 52.12, which describes a program model named 'Daisy'. That model is actually the top level description of the simulation. All you need to do in a setup file, is to specify values for the specified attributes. Then you will be ready to run it.

The main attribute is 'column', which describes the physical model itself. To run the model, you also need to specify 'time', which is the starting time of the simulation, 'manager', which specifies what management operations should be done

to the column, and 'weather' which provides the weather during the simulation. Furthermore, the 'output' attribute should be set to specify what data you want logged during the simulation. You can also use the 'input' command to split the setup description into multiple files.

In this section we will describe the individual Daisy attributes in more detail. It helps having an example to look at, the Daisy distribution (see section 1.5) contains a sample setup file named "`test.dai`". We do change that file from time to time, but the content shouldn't be too different from what you see here.

1.3.1 Input Files

The 'input' attribute isn't really an attribute. It is a command to read some more parameterizations or, more rarely, set some of the top level attributes. However, the format is the same as for singleton parser components (see sections 47, 1.1.3 and 1.4.8), except that you can have multiple 'input' statements, all taking effect.

The parser component is supposed to read parameterizations or top level attributes from an external source. The 'file' parser model (section 47.1) does this by reading a file in exactly the same format as the main setup file. It has a single (ordered) attribute. In the example below, we use the input command to read files containing parameterizations of management actions (tillage), crops, output (log) file formats, and soil types (horizons). The three first contain parameterizations that are used in most simulations.

```
(input file "tillage.dai")
(input file "crop.dai")
(input file "log.dai")
(input file "test-horizons.dai")
```

These parameterizations will be used when setting the attributes described in the rest of this section.

The file parser will search for files in the directories specified by the `DAISYPATH` environment variable (see 1.5), or by the 'path' attribute (section 52.12.

1.3.2 Simulation Time

The simplest attribute is 'time', which is a state variable holding the current time in the simulation. The value you specify in the setup file will be its value at the start of the simulation. However, if you output the time attribute value in a log file, you will see that time changes constantly during the simulation in steps of one hour.

The type of the 'time' attribute is a Time fixed component, as described in section 85.21. You specify four integer values separated by whitespace after the attribute name (time), which indicate the year, the month, the day in the month, and the hour (from 0 to 23) in that sequence.

```
(time 1986 12 1 1)
```

1.3.3 Weather

The type of the 'weather' attribute is a singleton library component, so the format is the same as for 'input' (section 1.3.1).

There are several weather models, but the main one is named 'default', and read files containing daily weather data, as described in appendix A. The first (ordered) attribute specifies the name of the weather file; "weather.dwf" is a file distributed together with Daisy, containing many years worth of weather data from KVL's research farm in Taastrup, near Copenhagen.

```
(weather file "weather.dwf")
```

1.3.4 Field

The value of the 'column' attribute should be a sequence of column components. The 'column' component is the main component of Daisy, most of the physical processes modelled by the simulation are attributes of the 'default' column component (see section 19.1). The 'default' column model is the only column model implemented at the time of writing.

Conventionally, the columns are specified by first defining a named parameterization for the default column, and then using that parameterization when setting the 'column' attribute. This is also what we do in the example.

First we define a parameterization named 'JB1_Andeby' of the 'default' column model. The command 'defcolumn' is used for this. The first argument to the 'defcolumn' is the name of the new parameterization (in this case: 'JB1_Andeby'), and the next argument is the name of the column model to parameterize (in this case 'default'). The second argument could also be a name of an existing parameterization, which you wanted to change or further refine. If there are no ordered attributes, the optional third argument is a documentation string. Otherwise, the second argument is followed by the ordered attributes (the default column model has no ordered attributes). Finally comes the unordered attributes. All separated by whitespace.

```
(defcolumn JB1_Andeby default
  "Data for the farm of B.And, Andeby, as collected by F.Guf, 2002."
  (Groundwater static)
  (Soil (horizons (-20 [cm] top_soil) (-250 [cm] bottom_soil))
        (MaxRootingDepth 60.0 [cm])
        (dispersivity 8.0 [cm]))
  (OrganicMatter (initial_SOM (-20  [cm] 4.86 [kg C/m^2])
                              (-40  [cm] 2.92 [kg C/m^2])
                              (-60  [cm] 1.46 [kg C/m^2])
                              (-80  [cm] 0.49 [kg C/m^2])
                              (-100 [cm] 0.16 [kg C/m^2])
                              (-120 [cm] 0.03 [kg C/m^2]))))
```

You can define named parameterizations for all components this way. The name of the command to define a new parameterization starts with 'def', followed by the name of the component. For example, the 'defweather' command can be used for defining named weather parameterizations.

The default column model has many attributes, namely one for each major physical process in the model. However, only two attributes have to be specified in the setup file, the rest have default or optional values.

- You need to specify a groundwater model. In this case we have specified static (unchanging) groundwater conditions with free drainage.
- You need to specify the physical properties of the soil.
  - Here we claim that the soil is divided in two horizons, the top horizon reaching down to 20 cm and using the physical properties specified by the 'top_soil' horizon parameterization, and the second reaching from 20 cm down to 250 cm, using the 'bottom_soil' horizon parameterization. You will usually have to define the horizon parameterizations yourself, the real '`test.dai`' file should give you an example of how to do that.

Technically, the 'horizons' attribute is a sequence of submodels, each submodel having two ordered attributes. The first attribute specifies the endpoint of that horizon (a number), while the second attribute specifies the horizon parameterization to use.

  - The two remaining attributes are physical properties that relate to the soil as a whole, rather than the individual horizons.

- In the example, we also specify 'initial_SOM' for the 'OrganicMatter' fixed component. The initial SOM is another sequence of submodels with two ordered attributes. In this case the first number is the depth in cm, and the second number is the soil carbon content in kg/m^2. Each submodel specifies a soil layer, with the first attribute indicating the layer's end point, and the second attribute the layer's carbon content. If the initial SOM is not explicitly specified, Daisy will use the humus content specified in the horizon to deduce it.

After defining the column parameterization, we can set the column attribute itself.

```
(column JB1_Andeby)
```

If we had defined two column parameterizations which we wanted to run concurrently, we could have set the column attribute like this:

```
(column JB1_Andeby JB1_Gaaseroed)
```

The most common case is to run a single column though, and at the time of writing very few of the standard log component parameterizations support logging more than a single column.

1.3.5 Management

The value of the 'manager' attribute should be a single parameterization of the 'action' library component. What makes the 'manager' attribute special, is that the 'action' library component contains a large number of primitive action models like 'sow' or 'fertilize', and a number of generic models which can be used for creating crop rotations out of these primitive actions.

The most common way to specify a manager is to use the 'activity' model (section 9.31). The 'activity' action model has a single ordered attribute, whose value is a sequence of actions. Each action will be performed in each time step, until done. Most actions are done after a single time step, but the 'wait' action is only done when the corresponding condition is true.

A common condition is 'at' (section 21.22), which is fulfilled only when the time of the simulation is equal to the time value specified in the condition. However, in the example we also use 'crop_ds_after' (section 21.7), which is fulfilled after the specified crop in the field has reached the specified development stage. The value specified here, 2.0, is reached when a crop is ripe. The action we then perform is to immediately harvest the crop, which means the condition will only be fulfilled once.

```
(manager activity
  (wait (at 1987 3 20 1))
  (plowing)
  (wait (at 1987 4 4 1))
  (fertilize (mineral
              (weight 100.0 [kg N/ha])
```

```
            (NH4_fraction 0.5 [])))
  (wait (at 1987 4 5 1))
  (progn
    (sow "Grass")
    (sow "Spring Barley"))
  (wait (crop_ds_after "Spring Barley" 2.0))
  (harvest "Spring Barley")
  (wait (at 1987 10 10 1))
  (harvest "Grass"
           (stub 15 [cm])                    ;Leave 15 cm stub.
           (stem 1.00 []))                  ;Harvest everything above stub.
  (wait (at 1987 9 8 1))
  (fertilize (mineral
              (weight 80.0 [kg N/ha])
              (NH4_fraction 0.5 [])))
  (wait (at 1987 10 10 1))
  (harvest "Grass"
           (stub 8. [cm])                    ;Leave 08 cm stub.
           (stem 1.00 []))                  ;Harvest everything above stub.
  (wait (at 1988 4 1 1))
  (stop))
```

Read chapter 9 and 21 to learn more about the available action and condition models.

1.3.6 Log File Generation

The value of the 'output' attribute should be a sequence of log component parameterizations. Each parameterization should specify some state or log variables to be output during the simulation.

The '`log.dai`' file contains a number of standard log parameterizations, for example 'Crop Production', which will write information about the development and production of any crops in the field to a file named '`crop_prod.dlf`' located in the directory specified by the 'directory' attribute, or, by default, the directory where daisy was started. This file will contain a header with some meta-information about the simulation, a line of dashes, and then a large number of tab separated columns, with each column describing one aspect of the crop development (e.g. the stem nitrogen content), and each row a specific time step. The two first rows are special, the first will contain a label identifying the column (e.g. NStem), and the second row the dimension used for the numbers in that column (e.g. kg N/ha).

You can obviously also define your own parameterizations. If you wish to do so, the 'table' log model (see section 39.5) is far the most useful. In the example, we specify two standard log parameterizations ('Crop Production' and 'Harvest') and one home-made parameterization ('Pond').

```
(deflog Pond table
  (where "pond.dlf")
  (when daily)
  (entries (number (path time year))
           (number (path time month))
           (number (path time mday))
           (number (path column "*" Surface pond)
                   (dimension "mm"))))
```

The 'table' log model has a number of parameters, most of these have default values. The three listed here are 'where', which specifies the name of the file to log the results in, 'when', which specifies how often to log the results, and 'entries', which specifies what to log. The 'entries' attribute is a sequence of 'select' (see section 62) models, where each entry specifies one column in the log file. The first entry listed will correspond to the first column in the log file.

Each log entry model has a large number of attributes, allowing you to accumulate results, extract values from the soil array, specify the tag and dimension printed in the two first rows of the column, and more. Most of these have default values, and only the fourth (and last) log entry submodel specifies one of these attributes, namely the dimension string to print in the second row.

The most important log entry parameter is 'path', which specifies the state or log variable we want to include in the log file. The value is a sequence of attribute names, which should be read as follows: Start with the Daisy model (section 52.12), and select an attribute in that model. That should be the first name listed in the path sequence.

- If the attribute is a time value, as in the three first log entries, you list one of 'year', 'month', 'mday', or 'hour' next in the path. The table log model will then log the selected part of the time value.
- If the selected attribute is a component sequence, as in the fourth log entry in the example, you should list the name of the model or parameterization you want to log next in the path list. In this case 'JB1_Andeby' would do, but instead we use the special value '*', which will match *all* entries in the sequence. However, their values will be accumulated into a single number in the log file, which is why we say most of the standard log models don't work well with multiple columns.
- After listing the name of the model or parameterization in the sequence to log, we must list an attribute of the model. In this case we list the 'Surface' fixed component.
- As long as the selected attribute itself is a submodel or fixed component, we keep listing a name of an attribute in *that* model next.
- In this case, it ends with the 'pond' attribute of the 'Surface' fixed component. The value of the 'pond' attribute is a single number, which will be logged in the fourth column of the '`pond.dlf`' log file.

That is the general strategy for writing log file entry paths. Keep listing names of nested attributes, until you reach a simple value, which will then be logged.

1.4 Types and Formatting

There are a few general rules for the formatting accepted by the 'file' parser.

1. Elements are separated by whitespace.
2. Grouping is done with parentheses. You can always (but need not) have whitespace before and after parentheses.
3. Whitespace is linefeeds, spaces, and tabs. You can have any number (greater than one) of these anywhere you can have whitespace.
4. Comments start with the character ';' and ends at the end of the line. Comments are legal anywhere you can have whitespace.

5. All names (attribute names, component names, model names, and parameterization names) follow the same conventions as string values, described in section 1.4.5.

In this section, we will describe how each type of attribute values should be formatted for each of the attribute types. The formatting does also depend slightly on whether the attribute is a singleton or a sequence, and whether the attribute is ordered or not, so we will cover that subject first.

1.4.1 Order and Size

Booleans, integers, numbers, and strings have a common format though. If *val* denotes the attribute value as described in sections 1.4.2, 1.4.3, 1.4.4, and 1.4.5, and *name* is the attribute name, then the format for the attributes in the setup files depends on whether they are ordered or not, and whether they are singletons or sequences, as illustrated in table 1.1.

	singleton	sequence
ordered	*val*	(*val* ...)
unordered	(*name val*)	(*name val* ...)

Table 1.1: Format for booleans, integers, numbers, and strings.

PLF's, submodels and fixed components are different though. Unlike the simpler types described above, these already contain several parts that are white space separated. Therefore, it is necessary to mark their start and end with parentheses except for unordered singletons, when there can be no ambiguity. Again, *name* is the attribute name, and *value* is the attribute value in the format described in sections 1.4.6 and 1.4.7.

	singleton	sequence
ordered	(*val*)	((*val*)...)
unordered	(*name val*)	(*name* (*val*)...)

Table 1.2: Format for time values, PLF's, submodels, and fixed components.

In general, library component values are written using the formatting conventions in table 1.2. However, there are two exceptions.

1. If you just specify the name of a complete parameterization, as in section 1.3.4, you do not need the extra parentheses, and can use the format conventions from table 1.1.

2. If there are no unordered attributes, then you do not need to put parentheses around the last attribute in the order. You can see an example of that in section 1.3.5. The action sequence is the last attribute in the 'clauses' submodel in the 'cond' action model.

1.4.2 Booleans

The simplest attribute type is booleans. A boolean value can be either '`false`' or '`true`', spelled exactly that way, with small letters.

1.4.3 Integers

Integers should start with a non-zero digit, and be followed by a sequence of digits. Negative numbers should start with a '`-`' sign, followed immediately with the digits. No whitespace is allowed. The only integer which should start with '`0`' is the integer zero itself, don't use leading zeros for padding. Use space instead. Otherwise, Daisy might misinterpret the integer on some systems.

1.4.4 Numbers

The general format for writing numbers is

<*sign*><*integer-part*>.<*fraction*>e<*sign*><*exponent*>

All of these can be left out, except for the integer part. There can be no whitespace between any of the parts.

After the number, you may optionally specify the dimension inside square brackets, like this `[cm]`. Fractions and other dimensionless numbers are specified as `[]`. Daisy will check that the specified dimension is also the dimension is expects, or a simple convertion thereof (like `[m]` instead of `[cm]`). If the deimension is listed as "unknown" in this manual, Daisy will be unable to check or convert it. It can still be specified for informational purposes, but the string must start with a question mark, like this `[?cm]`.

For lists of numbers, the dimension applies to all the number before it.

1.4.5 Strings

Daisy accepts two different string formats. The first format is called 'identifier'. Identifiers must start with a letter or an underscore, and can be followed by letters, digits or underscores only. They are conventionally used for attribute names.

The other format is called 'quoted strings', and are most often used for attribute values. Quoted strings start and begin with a double quote ('`"`'). Any characters in between the two double quote signs are part of the string, with these exceptions:

- To include a '`"`' in the string, you must write '`\"`'.
- To include a '`\`', you must write '`\\`'.
- If you put a backslash as the last character on the line, neither it, nor the following newline character will not be part of the string.
- A '`\n`' will be replaced by a newline character.

It is an error to put any other character after backslash.

Case is always significant.

File Names

Under some operating systems, file names are conventionally written like this:

```
C:\daisy\lib\log.dai
```

However, since '`\`' is treated specially in Daisy setup files, you will have to double all the backslashed, like this:

```
(input file "C:\\daisy\\sample\\test.dai")
```

Since this looks rather awkward, Daisy also supports ordinary slashes in file names:

```
(input file "C:/daisy/sample/test.dai")
```

1.4.6 PLF

PLF is the name used in Daisy for piecewise linear functions. In the setup files, such a function is specified as a sequence of `(x y)` pairs, where the x value must be monotonically increasing. The value of the PLF for any point in the x axis is then calculated by linear interpolation between the two closest points. The value *before* the first specified point will be the same as the value *at* the first specified point. Similarly, The value *after* the last specified point will be the same as the value *at* the last specified point.

An example might help. If 'f' is a unordered singleton PLF, we can specify its value like this:

```
(f (1.0 1.0) (3.0 2.0) (5.0 4.0))
```

We then get the values in table 1.3.

x	$f(x)$
0.0	1.0
1.0	1.0
2.0	1.5
3.0	2.0
4.0	3.0
5.0	4.0
6.0	4.0

Table 1.3: Interpolated values for f

1.4.7 Submodels and Fixed Components

A submodel or fixed component is basically a set of attribute names, with a value associated to each value. Some of the attributes are ordered, when specifying the value of a submodel you must specify the values for all the ordered attributes separated by whitespace, followed by the values for each of those unordered attributes you wish to set. The unordered attributes can, as the name implies, come in any order. How to write the attributes in the file depends on whether they are ordered or not, their type, and whether they are sequences or singletons. This is all described in section 1.4, in particular 1.4.1.

The fixed components are found in chapter 85, and the submodels are found with their parent models throughout part II. In the description of each submodel or fixed component, there is a *sample* showing the attributes you can specify, and how they should be placed. The sample is explained in section 1.2.2.

Submodels and fixed components are special compared to the other attribute values, in that they do not replace the original value. Instead, the original value will be amended with the individual attributes you specify. If you don't specify the value for a particular attribute, the original value will be used.

1.4.8 Library Components

A library component value should always start with the name of model or parameterization. If you want to overwrite some of the attribute values of the model or parameterization you just specified, you should write all the values of all the ordered attributes immediately after the name of the model or parameterization, separated by whitespace, and then the unordered attributes, just like you would for a submodel or fixed component (see section 1.4.7).

Each section of each chapter in part II describes a specific model. Part of that description is a *sample*, which shows how to specify that model in a setup file, and how to set the parameters for that model.

If you want to make partial modifications to the original value of the library component, you can specify the special keyword "original" instead of a name of a model or parameterization, followed by a list of those attribute names and value you want to overwrite.

The format for library component values described here is also used when defining named parameterizations. In the setup file, you write:

`(def`*component name value*`)`

where *component* is the name of the component you want to define a named parameterization for, *name* is the name of the new parameterization you want to define, and *value* is a library component value as described in this section (1.4.8).

1.5 The Daisy Distribution

In this section we describe the executable files which are part of the Daisy distribution, followed by a brief overview of the content of each of the standard parameterization files. Please note that these changes frequently, so the information here is unlikely to be complete and current.

1.5.1 The Daisy Program

The main Daisy executable is called '`daisy.exe`' on win32 systems, or just '`daisy`' on Unix. This is the command line version of Daisy.

To run a simulation, you should start the daisy executable with a setup file as an argument. A sample setup file named '`test.dai`' is distributed together with the sample parameterizations. You run it by typing

```
daisy test.dai
```

from the command line. Please note that you *must* include the full name of the file, that is include '`.dai`' file suffix. Setup files can be named anything, the '`.dai`' is only there to make them easier to recognize for the user.

The command above will only work if you are located in the directory containing the '`test.dai`' file, and all the parameterization files used by '`test.dai`'. However, we recommend that you place all the standard parameterization files in one directory, and use another for the setup files you create yourself. The `DAISYPATH` environment variable should then contain a list of directories which Daisy will search for directories. Under win32, the directories should be separated by semicolons, while under UNIX they should be separated by colons. How to set the environment variable depends on the system, here are some examples:

- Unix, using the C shell, in '`.cshrc`'.

  ```
  setenv DAISYPATH .:$HOME/daisy/sample:$HOME/daisy/lib
  ```

- Unix, using the Bourne shell, in '`.profile`'.

  ```
  DAISYPATH=.:$HOME/daisy/sample:$HOME/daisy/lib; export DAISYPATH
  ```

- Windows 95/98, in '`autoexec.bat`'.

  ```
  set DAISYPATH=.;C:\daisy\sample;C:\daisy\lib
  ```

- Under Windows NT, use `regedit` to add the environment variable.

Alternatively, you can specify both where to run the executable (and thus, where to create output files) and the path for searching for files directly in the beginning of the setup file, with the *directory* and *path* commands, like this:

```
(directory "C:/daisy/jsmith/sim1")
(path "." "C:/daisy/sample" "C:/daisy/lib")
```

Here we specify that all files generated must be put in the "sim1" directory, and that Daisy should look for files there first, followed by the "sample" and then the "lib" directories. Note that you must use slash (/) and not backslash (\) to separate directory components.

1.5.2 `tillage.dai`

The '`tillage.dai`' file contains parameterizations for the tillage operations listed below, whose nature should be apparent from their names.

- plowing
- rotavation
- disk_harrowing
- stubble_cultivation
- seed_bed_preparation

1.5.3 `crop.dai`

The '`crop.dai`' file contains crop parameterizations.

- "Grass"
- "Maize"
- "Pioneer Maize"
- "Ikuwala Maize"
- "Silage Maize"
- "Pea"
- "Potato"
- "Rye"
- "Spring Barley"
- "Spring Rape"
- "Sugar Beet"
- "Spring Wheat"
- "Winter Barley"
- "Winter Rape"
- "Winter Wheat"

The above parameterizations all refer to the default crop model, we also have some parameterizations of the old crop model. Please note that these cannot be used with multicrop systems.

- "Beet; V1"
- "Grass; V1"
- "Pea; V1"
- "Potato; V1"
- "Spring Barley; V1"
- "Spring Rape; V1"
- "Winter Barley; V1"
- "Winter Rape; V1"
- "Winter Wheat; V1"

1.5.4 `fertilizer.dai`

The parameterizations in the '`fertilizer.dai`' file are mostly intended as examples, as the parameters for organic fertilizer vary a lot.

- pig_slurry
- cattle_slurry
- pig_manure
- cattle_manure
- horse_manure

The '`fertilizer.dai`' file does also contain a partial parameterization named 'slurry', which contains the parameters common to the 'pig_slurry' and 'cattle_slurry' parameterizations, and partial parameterization named 'manure', which contains the parameters common to the 'pig_manure', 'cattle_manure' and 'horse_manure' parameterizations

1.5.5 `log.dai`

The '`log.dai`' file contains a number of standard log file descriptions. With the exception of 'Harvest', they only make sense for single column fields. Also with the exception of 'Harvest', they all use a tabular format, where the name and dimensions of each column are written in the two first rows.

- "Total Soil Content"
  This log file format will print information about the total soil content at midnight each day.
- "Soil Nitrate Concentration"
  Midnight nitrate concentration in all numeric layers.
- "Soil Temperature"
  Midnight soil temperature in all numeric layers.

- "Soil Water Potential"
  Midnight water potential in all numeric layers.
- "Surface Water Balance"
  Enough information to calculate a daily surface water balance.
- "Root Zone Water Balance"
  Enough information to calculate a daily root zone water balance.
- "N Balance"
  Enough information to calculate a daily nitrogen balance.
- "Soil Chemicals"
  Daily information about the fate of the soil chemicals.
- "Surface Chemicals"
  Daily information about the fate of the surface chemicals.
- "Crop Production"
  Daily crop development numbers for the default crop model.
- "Old Crop Production"
  Daily crop development numbers for the old crop model.
- "Bioclimate"
  All the information you ever wanted about the bioclimate, and more.
- "harvest"
  Information about all harvests during the simulation.

1.5.6 Error Messages

Sometimes the simulation will fail to run for some reason or another. In these cases, Daisy will attempt to give an error message, which hopefully will give some indication of what went wrong. There are zillions of situations where Daisy might give some error message, in the following we describe some of them.

Parser errors

If the format of the setup files is wrong, for example if there are missing parentheses or unknown attributes, Daisy will give you information about which file the error was found, the line number, the column number, and what the problem was.

```
test.dai:15:6: Unknown attribute 'kurt'
```

This is the "best" kind of errors, since it will lead you directly to the place where the error occurred.

If you run Daisy with '`M-x compile RET`' inside the Emacs text editor, you can go directly to the place the error occurred by clicking on the error message with the mouse.

Completeness checks

After parsing the setup files, Daisy will check that all the required parameters and state variables are defined, and make some very simple consistency checks. Errors during this phase looks as follows:

```
* Daisy
time missing
```

Consistency checks

After checking that all the required attributes are there, Daisy will create the simulation objects, and then perform a more throughout consistency check. Errors found during these phases will be reported like the one below.

```
* column
** Andeby
*** SoilHeat
You have 20 intervals but 60 T values
```

Runtime exceptions

Now Daisy is ready to run the simulation. However, some errors are too hard to find before the simulation has started. If Daisy finds one of those during the simulation, it will throw an exception, which normally causes the program to exit.

```
Exception: Cannot restrict already restricted field
```

Assertion failures

During debugging, a lot of 'assert' statements have been added to the code. Each of these statements will check if a certain condition is true, and if not, exit the program with a message like

```
field.C:71: failed assertion 'false'
```

A failed assertion message will always print the file name, the line number, and the assertion that failed, but the format may vary between different platforms.

The difference between assertion failures and runtime exceptions is that the runtime exceptions are normally due to an error in the setup files, while the assertion may also be caused by bugs in the program. The printed information is a great help for debugging those. So failed assertions should be reported to KVL.

Program errors

Sometimes we can get errors that aren't even caught by the assertions, they can manifest themselves as "Bus error" or "Segmentation violation" errors, or weird pop-up windows under win32. Or even worse, as bogus numbers in the log files. These can be due to program errors, hardware errors, too little memory, or errors in the operating system. They can also be caused by errors in the setup files, but in this case it is also a program error, as such errors should be caught by one of the mechanisms above.

Part II

The Components

Chapter 2

ABAproduction

The 'ABAproduction' component calculates the prod of ABA in soil.
Used by RootSystem @ ABAprod (see 85.13, page 423) .

< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

2.1 none

No ABA production.
Used by RootSystem @ ABAprod (see 85.13, page 423) .

2.2 soil

ABA production based on soil location.

- *expr*: **number** component (see chapter 45)
  Expression to evaluate to ABA uptake [g/cm^3/h]. The symbol 'h' will be bound to the water pressure [cm]. The symbol 'L' will be bound to the root density [cm/cm^3]. The symbol 'S' will be bound to the water uptake [cm^3/cm^3/h].

2.3 root

ABA production based on production in roots.
The assumptions are that that each length of root will produce ABA with a rate that depends solely on the water pressure in that cell, and that all the ABA will be included in the water uptake.

- *expr*: **number** component (see chapter 45)
  Expression to evaluate to ABA production per root length [g/cm/h]. The symbol 'h' will be bound to the water pressure [cm].

2.4 uptake

ABA production based on concentration in water uptake.

The assumption is water uptake from roots in specific region of the soil comes with a specific ABA concentration, which depends solely on the water pressure in that region.

- *expr*: **number** component (see chapter 45)
  Expression to evaluate to ABA concentration in water uptake [g/cm^3]. The symbol 'h' will be bound to the water pressure [cm].

Chapter 3

AOM

A single Added Organic Matter pool.

Used by Bioincorporation @ AOM (see 85.3, page 411) , and Harvesting @ Root (see 85.16, page 431) .

```
< component   (N N ...)
              (description description)
              (cite)
              (C C ...)
              (heat_factor heat_factor)
              (water_factor water_factor)
              (C_per_N C_per_N ...)
              (turnover_rate turnover_rate)
              (turnover_halftime turnover_halftime)
              (efficiency efficiency ...)
              (fractions fractions ...)
              (initial_C_per_N initial_C_per_N)
              (initial_fraction initial_fraction)
              (top_C 0 [g C/cm^2])
              (top_N 0 [g N/cm^2]) >
```

- *N*: number [**g N/cm**^{3}] soil cells
  Optional state variable
  Nitrogen in each soil interval.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *C*: number [**g C/cm**^{3}] soil cells
  Optional state variable
  Carbon in each soil interval.

- *heat_factor*: plf [**dg C** → <**none**>]
  Optional parameter
  Heat factor. If empty, use default from 'OrganicMatter'.

- *water_factor*: plf [**cm** → <**none**>]
  Optional parameter
  Water potential factor. If empty, use default from 'OrganicMatter'.

- *C_per_N*: number [**(g C/cm^3)/(g N/cm^3)**] soil cells
  Optional state variable
  The carbon/nitrogen ratio.

- *turnover_rate*: number [**h**$^{-1}$]
  Optional parameter
  Fraction converted to other pools each hour. You must specify either this or 'turnover_halftime'.

- *turnover_halftime*: number [**h**]
  Optional parameter
  Time until half had been converted to other pools. You must specify either this or 'turnover_rate'.

- *efficiency*: number [<**fraction**>] sequence
  Parameter
  The efficiency this pool can be digested by each of the SMB pools.

- *fractions*: number [<**fraction**>] sequence
  Parameter
  How this pool is divided into other pools. The first numbers corresponds to each of the SMB pools, the next number to the SOM buffer, and any remaining numbers to each of the DOM pools. The length of the sequence should thus be the number of SMB pools plus 1 plus optionally the number of DOM pools.

- *initial_C_per_N*: number [**g C/g N**]
  Optional state variable
  The initial C/N ratio when this pool is created. Negative numbers mean unspecified.

- *initial_fraction*: number [<**fraction**>]
  Optional parameter
  The initial fraction of the total available carbon allocated to this pool for AOM. One pool should be left unspecified.

- *top_C*: number [**g C/cm^2**]
  State variable (default 0)
  Carbon on top of soil.

- *top_N*: number [**g N/cm^2**]
  State variable (default 0)
  Nitrogen on top of soil.

3.1 AOM-SLOW

Slow AOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]

Used by crop simple root_am (see 22.56, page 150) , vegetation permanent litter_am (see 79.2, page 383) , am root om (see 11.45, page 96) , and Harvesting @ Root (see 85.16, page 431) .

3.2 AOM-SLOW-BIOINCORPORATION

A 'AOM-SLOW' model (see 3.1, page 52) build into Daisy.
Lower C/N ration for bioincorporated matter. See also [Bruun et al., 2003]
Used by Bioincorporation @ AOM (see 85.3, page 411) .

3.3 AOM-SLOW-OLD

A 'AOM-SLOW' model (see 3.1, page 52) build into Daisy.
Original parameterization of the slow AOM pool. See also [Müller et al., 1997]

3.4 CROP-SLOW

A 'AOM-SLOW' model (see 3.1, page 52) build into Daisy.
Parameterization used for slow pool of some crop residuals. See also [Bruun et al., 2003]

3.5 Ryegrass-SLOW

A 'CROP-SLOW' model (see 3.4, page 53) defined in 'ryegrass.dai'.
Slow AOM pool modified for ryegrass. See also [Bruun et al., 2003]

3.6 Wclover-SLOW

A 'CROP-SLOW' model (see 3.4, page 53) defined in 'wclover.dai'.
Slow AOM pool modified for Wclover. See also [Bruun et al., 2003]

3.7 AOM-FAST

Fast AOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]
Used by crop simple root_am (see 22.56, page 150) , vegetation permanent litter_am (see 79.2, page 383) , am root om (see 11.45, page 96) , Bioincorporation @ AOM (see 85.3, page 411) , and Harvesting @ Root (see 85.16, page 431) .

3.8 CROP-FAST

A 'AOM-FAST' model (see 3.7, page 53) build into Daisy.
Parameterization used for fast pool of some crop residuals. See also [Bruun et al., 2003]

Chapter 4

ClayOM

Find the effect of clay on organic matter processing.

4.1 old

Traditional clay influence on organic matter.
Used by organic default ClayOM (see 46.2, page 243) .

- *factor*: plf [<**fraction**> → <**none**>]
  Parameter (has default value with 3 points)

  ```
  (factor (0 1) (0.25 0.5) (1 0.5))
  ```

  Parameter description:
  Function of clay content, multiplied to the maintenance and turnover rates of SMB1 and all SOM pools.

4.2 biomod

Clay influence on organic matter from BIOMOD project. All SMB pools are affected, but not the SOM pools. Additionally, the ration between maintenance and turnover is also clay dependent.

- *factor*: plf [<**fraction**> → <**none**>]
  Parameter (has default value with 3 points)

  ```
  (factor (0 1) (0.25 0.5) (1 0.5))
  ```

  Parameter description:
  Function of clay content, multiplied to the maintenance and turnover rates of the SMB pools.

- *a*: number (dimensionless)
  Parameter
  Maintenance parameter.

- *E_SMB*: number [<**fraction**>]
  Parameter
  SMB efficiency in processing organic matter. Note that you must set the 'efficiency' parameter for all OM pools to this number for the BIOMOD clay response model to work correctly.

- *f_SMB1*: number [<**fraction**>]
  Parameter
  Fraction of AOM pools goind to SMB1. Only the fraction of AOM going to a SMB pool count, so this is really a fraction of the fraction coing to the SMB pools. Note that you must set the 'fraction' parameter of all AOM pools to reflect this for the BIOMOD clay response model to work correctly.

Chapter 5

MV_Crop

Description of a crop for use by the MARKVAND model.

5.1 default

Standard MARKVAND crop model.

- *S_F*: number [**dg C d**] sequence
  Parameter
  Temperature sum for each phase.
- *A_F*: number [<**fraction**>] sequence
  Parameter
  Allowable water deficit for each phase before irrigation.
- *L_gv*: number (dimensionless)
  Parameter
  Green leaf area index at emergence / growth start.
- *L_ge*: number (dimensionless)
  Parameter
  Green leaf area index at the time where growth rate become exponential.
- *L_gx*: number (dimensionless)
  Parameter
  Maximum green leaf area index.
- *L_gm*: number (dimensionless)
  Parameter
  Green leaf area index at maturity.
- *L_ym*: number (dimensionless)
  Parameter
  Yellow leaf area index at maturity.
- *S_Le*: number [**dg C d**]
  Parameter
  Temperature sum when green LAI growth turn exponential.
- *S_Lx*: number [**dg C d**]
  Parameter
  Temperature sum maximum green LAI.

- *S_Lr*: number [**dg C d**]
  Parameter
  Temperature sum for start of yellow leaves.

- *S_Lm*: number [**dg C d**]
  Parameter
  Temperature sum at maturity.

- *z_0*: number [**mm**]
  Parameter
  Root depth before emergence (growth start).

- *z_v*: number [**mm**]
  Parameter
  Root depth at emergence (growth start).

- *z_xA*: number [**mm**]
  Parameter
  Maximum root depth for this crop.

- *z_m*: number [**mm**]
  Parameter
  Root depth at maturity.

- *c_r*: number [**mm/d**]
  Parameter
  Root penetration rate.

Chapter 6

MV_Soil

Description of a soil for use by the MARKVAND model.

6.1 default

Standard MARKVAND soil model.

- *z_o*: number [**mm**]
  Parameter
  Depth of top soil.
- *z_xJ*: number [**mm**]
  Parameter
  Max rooting depth.
- *Theta_fo*: number [<**fraction**>]
  Parameter
  Field capacity, topsoil.
- *Theta_wo*: number [<**fraction**>]
  Parameter
  Wilting point, topsoil.
- *Theta_fu*: number [<**fraction**>]
  Parameter
  Field capacity, subsoil.
- *Theta_wu*: number [<**fraction**>]
  Parameter
  Wilting point, subsoil.
- *C_e*: number [**mm**]
  Parameter
  Capacity of evaporation reservoir.
- *c_e*: number [<**fraction**>]
  Parameter
  Basic evaporation factor.
- *c_T*: number [**mm**]
  Parameter
  Transpiration constant.

- *k_qr*: number (dimensionless)
  Parameter
  Drainage constant root zone.

- *k_qb*: number (dimensionless)
  Parameter
  Drainage constant subsone.

Chapter 7

SMB

A single Soil MicroBiological pool.

```
< component   (N N ...)
              (description description)
              (cite)
              (C C ...)
              (heat_factor heat_factor)
              (water_factor water_factor)
              (C_per_N C_per_N ...)
              (turnover_rate turnover_rate)
              (turnover_halftime turnover_halftime)
              (efficiency efficiency ...)
              (fractions fractions ...)
              (initial_C_per_N initial_C_per_N)
              (maintenance maintenance) >
```

- *N*: number [**g N/cm**^{3}] soil cells
  Optional state variable
  Nitrogen in each soil interval.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *C*: number [**g C/cm**^{3}] soil cells
  Optional state variable
  Carbon in each soil interval.

- *heat_factor*: plf [**dg C** → <**none**>]
  Optional parameter
  Heat factor. If empty, use default from 'OrganicMatter'.

- *water_factor*: plf [**cm** → <**none**>]
  Optional parameter
  Water potential factor. If empty, use default from 'OrganicMatter'.

- *C_per_N*: number [**(g C/cm^3)/(g N/cm^3)**] soil cells
  Optional state variable
  The carbon/nitrogen ratio.

- *turnover_rate*: number [**h**$^{-1}$]
  Optional parameter
  Fraction converted to other pools each hour. You must specify either this or 'turnover_halftime'.

- *turnover_halftime*: number [**h**]
  Optional parameter
  Time until half had been converted to other pools. You must specify either this or 'turnover_rate'.

- *efficiency*: number [<**fraction**>] sequence
  Parameter
  The efficiency this pool can be digested by each of the SMB pools.

- *fractions*: number [<**fraction**>] sequence
  Parameter
  How this pool is divided into other pools. The first numbers corresponds to each of the SMB pools, the next numbers corresponds to the SOM pools, and the last numbers to each of the DOM pools. The length of the sequence should thus be the number of SMB pools plus the number of SOM pools plus the number of DOM pools.

- *initial_C_per_N*: number [**g C/g N**]
  Optional state variable
  The initial C/N ratio when this pool is created. Negative numbers mean unspecified.

- *maintenance*: number [**h**$^{-1}$]
  Parameter
  The fraction used for staying alive each hour.

7.1 SMB-SLOW

Slow SMB pool parameterization.
Used by organic default smb (see 46.2, page 243) .

7.2 SMB-FAST

Fast SMB pool parameterization.
Used by organic default smb (see 46.2, page 243) .

Chapter 8

SOM

A single Soil Organic Matter pool.

```
< component  (N N ...)
             (description description)
             (cite)
             (C C ...)
             (heat_factor heat_factor)
             (water_factor water_factor)
             (C_per_N C_per_N ...)
             (turnover_rate turnover_rate)
             (turnover_halftime turnover_halftime)
             (efficiency efficiency ...)
             (fractions fractions ...)
             (initial_C_per_N initial_C_per_N) >
```

- *N*: number [**g N/cm**^{3}] soil cells
  Optional state variable
  Nitrogen in each soil interval.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *C*: number [**g C/cm**^{3}] soil cells
  Optional state variable
  Carbon in each soil interval.

- *heat_factor*: plf [**dg C** → <**none**>]
  Optional parameter
  Heat factor. If empty, use default from 'OrganicMatter'.

- *water_factor*: plf [**cm** → <**none**>]
  Optional parameter
  Water potential factor. If empty, use default from 'OrganicMatter'.

- *C_per_N*: number [**(g C/cm**^{3}**)/(g N/cm**^{3}**)**] soil cells
  Optional state variable
  The carbon/nitrogen ratio.

- *turnover_rate*: number [$\mathbf{h}^{-1}$]
  Optional parameter
  Fraction converted to other pools each hour. You must specify either this or 'turnover_halftime'.

- *turnover_halftime*: number [**h**]
  Optional parameter
  Time until half had been converted to other pools. You must specify either this or 'turnover_rate'.

- *efficiency*: number [<**fraction**>] sequence
  Parameter
  The efficiency this pool can be digested by each of the SMB pools.

- *fractions*: number [<**fraction**>] sequence
  Parameter
  How this pool is divided into other pools. The first numbers corresponds to each of the SMB pools, the next numbers corresponds to the SOM pools, and the last numbers to each of the DOM pools. The length of the sequence should thus be the number of SMB pools plus the number of SOM pools plus the number of DOM pools.

- *initial_C_per_N*: number [**g C/g N**]
  Optional state variable
  The initial C/N ratio when this pool is created. Negative numbers mean unspecified.

8.1 SOM-SLOW

Slow SOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]
Used by organic default som (see 46.2, page 243) .

8.2 SOM-SLOW-OLD

A 'SOM-SLOW' model (see 8.1, page 64) build into Daisy.
Original parameterization of the slow SOM pool. See also [Müller et al., 1997]

8.3 SOM-FAST

Fast SOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]
Used by organic default som (see 46.2, page 243) .

8.4 SOM-FAST-OLD

A 'SOM-FAST' model (see 8.3, page 64) build into Daisy.
Original parameterization of the fast SOM pool. See also [Müller et al., 1997]

8.5 SOM-INERT

Inert SOM pool parameterization.
Used by organic default som (see 46.2, page 243) .

Chapter 9

action

The 'action' component represents management on different abstraction levels, from a single tillage operation to strategies of how to manage a farm. Typically, but not necessarily, the high level management strategies are build by combining low level management operations.

9.1 set_surface_detention_capacity

Set amount of ponding the surface can retain.

- *height*: number [**cm**]
  Parameter
  Max ponding height before runoff.

9.2 spray

Spray a chemical (typically a pesticide) on the field.

- *chemical*: string (see section 1.4.5)
  Parameter
  Name of pesticide to spray.

- *amount*: number [**g/ha**]
  Parameter
  Amount of pesticide to spray.

9.3 with-column

Perform actions on a specific column.

- *column*: string (see section 1.4.5)
  Parameter
  Name of column to perform actions on.

- *actions*: **action** component (see chapter 9) sequence
  Actions to perform on the specified column.

9.4 emerge

Force a crop to emerge.

- *crop*: string (see section 1.4.5)
  Parameter (default 'all')
  Name of the crop to emerge. If you specify 'all', all crops will emerge. If there are no crop on the field with the specified name, nothing will happen.

9.5 harvest_base

Common parameters for harvest operations.

- *crop*: string (see section 1.4.5)
  Parameter (default 'all')
  Name of the crop to harvest or cut. If you specify 'all', all crops will be harvested. If there are no crop on the field with the specified name, nothing will happen.

- *combine*: boolean (see section 1.4.2)
  Parameter (default false)
  Set this to 'true' in order to combine all crop parts into stem in the harvest log files. This is mostly useful for silage.

- *stub*: number [**cm**]
  Parameter (default 0)
  Leave stem and leafs below this height on the field.

- *stem*: number [<**fraction**>]
  Parameter (default 1)
  Fraction of stem (above stub) to harvest.

- *leaf*: number [<**fraction**>]
  Parameter (default 1)
  Fraction of leafs (above stub) to harvest.

- *sorg*: number [<**fraction**>]
  Parameter (default 1)
  Fraction of storage organ to harvest.

9.6 harvest

A 'harvest_base' model (see 9.5, page 66) build into Daisy.
Harvest a crop.

9.7 cut

A 'harvest_base' model (see 9.5, page 66) build into Daisy.
Cut a crop.

9.8 pluck

Pluck a crop. Unlike the 'harvest' operation, this allows you to pluck selected parts of the above ground dry matter without killing the crop. It is intended for crops like tomatoes, that are harvested multiple times.

- *crop*: string (see section 1.4.5)
  Parameter (default 'all')
  Name of the crop to pluck. If you specify 'all', all crops will be plucked. If there are no crop on the field with the specified name, nothing will happen.

- *stem*: number [<**fraction**>]
  Parameter (default 0)
  Fraction of stem to pluck.

- *leaf*: number [<**fraction**>]
  Parameter (default 0)
  Fraction of leaves to pluck.

- *sorg*: number [<**fraction**>]
  Parameter (default 1)
  Fraction of storage organ to pluck.

9.9 mix

Mix soil content down to the specified depth. The effect is that nitrogen, water, temperature and such are averaged in the interval.

- *depth*: number [**cm**]
  Parameter
  How far down to mix the soil (a negative number).

- *penetration*: number [<**fraction**>]
  Parameter (default 1)
  Fraction of organic matter on surface that are incorporated in the soil by this operation.

9.10 rotavation

A 'mix' model (see 9.9, page 67) defined in 'tillage.dai'.

9.11 disk_harrowing

A 'mix' model (see 9.9, page 67) defined in 'tillage.dai'.

9.12 stubble_cultivation

A 'mix' model (see 9.9, page 67) defined in 'tillage.dai'.

9.13 seed_bed_preparation

A 'mix' model (see 9.9, page 67) defined in 'tillage.dai'.

9.14 swap

Swap two soil layers. The top layer start at the surface and goes down to 'middle', and the second layer starts with 'middle' and goes down to 'depth'. After the operation, the content (such as heat, water, and organic matter) will be averaged in each layer, and the bottom layer will be placed on top of what used to be the top layer.

- *depth*: number [**cm**]
  Parameter
  The end of the second layer to swap.

- *middle*: number [**cm**]
  Parameter
  The end of the first layer and the start of the second layer to swap.

9.15 plowing

A 'swap' model (see 9.14, page 68) defined in 'tillage.dai'.

9.16 set_porosity

Set the porosity of the horizon at the specified depth. To get useful results, you need to use a hydraulic model that supports this.

- *depth*: number [**cm**]
  Parameter (default 0)
  A point in the horizon to modify.

- *porosity*: number [<**fraction**>]
  Parameter
  Non-solid fraction of soil.

9.17 nil

This action does nothing, always done.
Used by action if else (see 9.21, page 69) .

9.18 t

This action does nothing, never done.

9.19 progn

Perform all the specified actions in the sequence listed. All the actions will be performed in the same time step.

- *actions*: **action** component (see chapter 9) sequence
  List of actions to perform.

9.20 cond

Perform the actions associated with the first true condition in the list.

- *clauses*: submodel (see section 1.4.7) sequence
  Each clause consist of a condition and a sequence of actions. The first clause whose condition is true, will have its actions activated.

  < *condition actions...* >

  - *condition*: **condition** component (see chapter 21)
    Condition for performing the actions.
  - *actions*: **action** component (see chapter 9) sequence
    Actions to perform when condition is meet.

9.21 if

If the condition is true, perform the first action, otherwise perform the second action.

- *if*: **condition** component (see chapter 21)
  Condition determining which action to perform.
- *then*: **action** component (see chapter 9)
  Action to perform if the condition is true.
- *else*: **action** component (see chapter 9)
  Component (default 'nil')
  Action to perform if the condition is false.

9.22 irrigate_base

Shared parameter for irrigate actions.

- *flux*: **number** component (see chapter 45)
  Amount of irrigation applied.
- *solute*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Solutes in irrigation water.

  < *name value* >

  - *name*: string (see section 1.4.5)
    Parameter
    Name of chemical.
  - *value*: number [**ppm**]
    Parameter
    Value for chemical.
- *hours*: integer
  Optional parameter
  Irrigate this number of hours. By default, irrigate 1 hour if days is 0, and 0 hours plus the specified number of days else.
- *days*: integer
  Parameter (default 0)
  Irrigate this number of days.

- *remaining_time*: number [**h**]
  Optional state variable
  Irrigate this number of hours.Setting this overrides the 'days' and 'hours' parameters.

- *temperature*: number [**dg C**]
  Optional parameter
  Temperature of irrigation (default: air temperature).

9.23 irrigate_overhead

A 'irrigate_base' model (see 9.22, page 69) build into Daisy.
Irrigate the field from above.

9.24 irrigate_surface

A 'irrigate_base' model (see 9.22, page 69) build into Daisy.
Irrigate the field directly on the soil surface, bypassing the canopy.

9.25 irrigate_subsoil

A 'irrigate_base' model (see 9.22, page 69) build into Daisy.
Irrigate the field directly into the soil. Currently, the 'temperature' parameter is ignored.

- *flux*: **number** component (see chapter 45)
  Amount of irrigation applied.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to add irritaion.

- *from*: number [**cm**]
  Optional parameter
  Height where you want to start the incorporation (a negative number). OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Optional parameter
  Height where you want to end the incorporation (a negative number). OBSOLETE: Use (volume box (bottom TO)) instead.

9.26 stop

Stop the simulation.

9.27 sow

Sow a crop on the field.

- *crop*: **crop** component (see chapter 22)
  Crop to sow.

- *seed*: number [**g w.w./m**^{2}]
  Optional parameter
  Amount of seed applied. By default, initial growth will be governed by 'typical' seed amounts.

- *row_width*: number [**cm**]
  Parameter (default 0)
  Distance between rows. Specify zero to spread equally over the area (no rows).

- *plant_distance*: number [**cm**]
  Optional parameter
  Distance between plants.

  Setting this will overrule 'row_width'. The only purpose of this paramater is to provide the user with a more intuitive name for 'row_width' for the situation where you have a 2D simulation, where the x axis is parallel with the actual rows in the field, rather than ortogonal to the rows as is otherwise assumed by Daisy.

- *row_position*: number [**cm**]
  Parameter (default 0)
  Position of plant row on x-axes. Specify zero to spread equally over the area (no rows).

- *plant_position*: number [**cm**]
  Optional parameter
  Position of plant on x-axes.

  Setting this will overrule 'row_position'. The only purpose of this paramater is to provide the user with a more intuitive name for 'row_position' for the situation where you have a 2D simulation, where the x axis is parallel with the actual rows in the field, rather than ortogonal to the rows as is otherwise assumed by Daisy.

9.28 plant

A 'sow' model (see 9.27, page 70) build into Daisy.
'plant' is anothor name for 'sow'

9.29 crop

Manage a specific crop or multicrop.

- *spray*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  List of chemicals to apply.

  < *month day name amount* >

  - *month*: integer
    Parameter
    Month in the year.
  - *day*: integer
    Parameter
    Day in the month.

  - *name*: string (see section 1.4.5)
    Parameter
    Name of chemichal to spray.
  - *amount*: number [**g/ha**]
    Parameter
    Amount of chemichal to spray.

- *secondary*: submodel (see section 1.4.7)
  Optional submodel
  Secondary crop, if any.

```
<  (crop crop)
   (date date)      ; Has partial value.
   (done false) >
```

  - *crop*: **crop** component (see chapter 22)
    Crop to sow.
  - *date*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(date <missing month> <missing day>)
```

    Parameter description:
    Date to sow.

```
<  month day
   (hour 8) >
```

    * *month*: integer
      Parameter
      Month in the year.
    * *day*: integer
      Parameter
      Day in the month.
    * *hour*: integer
      Parameter (default 8)
      Hour in the day.
  - *done*: boolean (see section 1.4.2)
    State variable (default false)
    True iff the crop has been sowed.

- *primary*: submodel (see section 1.4.7)
  Submodel (has partially specified default value)

```
(primary )
```

  Parameter description:
  Primary crop.

```
<  (crop crop)
   (date date)      ; Has partial value.
   (done false) >
```

  - *crop*: **crop** component (see chapter 22)
    Crop to sow.
  - *date*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(date <missing month> <missing day>)
```

Parameter description:
Date to sow.

< *month day*
(hour 8) >

* *month*: integer
  Parameter
  Month in the year.
* *day*: integer
  Parameter
  Day in the month.
* *hour*: integer
  Parameter (default 8)
  Hour in the day.

– *done*: boolean (see section 1.4.2)
State variable (default false)
True iff the crop has been sowed.

- *harvest_annual*: submodel (see section 1.4.7)
Optional submodel
Harvest parameters for annual crops.

```
<  (done false)
   (loss loss)
   (remove_residuals remove_residuals)
   (latest latest)                        ; Has partial value. >
```

– *done*: boolean (see section 1.4.2)
State variable (default false)
True iff the crop has been sowed.

– *loss*: number [<**fraction**>]
Parameter
Fraction lost during harvest.

– *remove_residuals*: boolean (see section 1.4.2)
Parameter
Remove residuals at harvest.

– *latest*: submodel (see section 1.4.7)
Submodel (has partially specified default value)

```
(latest <missing month> <missing day>)
```

Parameter description:
Latest harvest date. If the crop is ripe before this date, it will be harvested at that point.

< *month day*
(hour 8) >

* *month*: integer
  Parameter
  Month in the year.
* *day*: integer
  Parameter
  Day in the month.
* *hour*: integer
  Parameter (default 8)
  Hour in the day.

- *harvest_perennial*: submodel (see section 1.4.7)
  Optional submodel
  Harvest conditions for perennial crops.

  ```
  <  (fertilize fertilize ...)
     (DS DS)
     (end end)                                  ; Has partial value.
     (DM DM)
     (seasons seasons)
     (year_of_last_harvest year_of_last_harvest)
     (fertilize_index 0)
     (fertilize_rest fertilize_rest ...)
     (fertilize_rest_index 0)
     (fertilize_year fertilize_year) >
  ```

  - *fertilize*: **am** component (see chapter 11) sequence
    Optional component
    Fertilizer applications after harvest first season. First season is defined as the year where the first harvest occurs.
  - *DS*: number (dimensionless)
    Parameter
    Development stage at or above which to initiate harvest.
  - *end*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

    ```
    (end <missing month> <missing day>)
    ```

    Parameter description:
    End management this date the last season.

    ```
    <  month day
       (hour 8) >
    ```

    * *month*: integer
      Parameter
      Month in the year.
    * *day*: integer
      Parameter
      Day in the month.
    * *hour*: integer
      Parameter (default 8)
      Hour in the day.
  - *DM*: number [**kg DM/ha**]
    Parameter
    Dry matter at or above which to initiate harvest.
  - *seasons*: integer
    Parameter
    Number of seasons to harvest crop. The crop will be harvested whenever the specified DS or DM are reached. The first season is the year the crop management starts.
  - *year_of_last_harvest*: integer
    Optional state variable
    Year of last season.
  - *fertilize_index*: integer
    State variable (default 0)
    Next entry in 'fertilize' to execute.

  - *fertilize_rest*: **am** component (see chapter 11) sequence
    Optional component
    Fertilizer applications after harvest remaining seasons. If missing, use the same fertilizer as first season.
  - *fertilize_rest_index*: integer
    State variable (default 0)
    Next entry in 'fertilize_rest' to execute.
  - *fertilize_year*: integer
    Optional state variable
    Year last fertilization was applid.

- *fertilize_at*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Fertilizer application by date.

  < *month day what* >

  - *month*: integer
    Parameter
    Month in the year.
  - *day*: integer
    Parameter
    Day in the month.
  - *what*: **am** component (see chapter 11)
    Fertilizer to apply.

- *fertilize_at_index*: integer
  State variable (default 0)
  Next entry in 'fertilize_at' to execute.

- *fertilize_incorporate*: boolean (see section 1.4.2)
  Parameter (default false)
  Incorporate organic fertilizer in plowing zone.

- *tillage*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  List of tillage operations to apply.

  < *month day operation* >

  - *month*: integer
    Parameter
    Month in the year.
  - *day*: integer
    Parameter
    Day in the month.
  - *operation*: **action** component (see chapter 9)
    Tillage operation.

- *tillage_index*: integer
  State variable (default 0)
  Next entry in 'tillage' to execute.

- *spray_index*: integer
  State variable (default 0)
  Next entry in 'spray' to execute.

- *irrigation*: submodel (see section 1.4.7)
  Optional submodel
  Irrigation model for first season. If missing, don't irrigate.

```
<   (flux 2 [mm/h])
    (amount amount)
    (from from)                ; Has partial value.
    (to to)                    ; Has partial value.
    (potential potential) >
```

  - *flux*: number [**mm/h**]
    Parameter (default 2)
    Water application speed.
  - *amount*: number [**mm**]
    Parameter
    Amount of water to apply on irrigation.
  - *from*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(from <missing month> <missing day>)
```

    Parameter description:
    Start of irrigation period.

```
<   month day
    (hour 8) >
```

    * *month*: integer
      Parameter
      Month in the year.
    * *day*: integer
      Parameter
      Day in the month.
    * *hour*: integer
      Parameter (default 8)
      Hour in the day.
  - *to*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(to <missing month> <missing day>)
```

    Parameter description:
    End of irrigation period.

```
<   month day
    (hour 8) >
```

    * *month*: integer
      Parameter
      Month in the year.
    * *day*: integer
      Parameter
      Day in the month.
    * *hour*: integer
      Parameter (default 8)
      Hour in the day.
  - *potential*: number [**cm**]
    Parameter
    Soil potential at which to irrigate.

- *irrigation_rest*: submodel (see section 1.4.7)
  Optional submodel
  Irrigation model for remaining seasons. If missing, use the same model as first season.

```
< (flux 2 [mm/h])
  (amount amount)
  (from from)                ; Has partial value.
  (to to)                    ; Has partial value.
  (potential potential) >
```

  - *flux*: number [**mm/h**]
    Parameter (default 2)
    Water application speed.
  - *amount*: number [**mm**]
    Parameter
    Amount of water to apply on irrigation.
  - *from*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(from <missing month> <missing day>)
```

    Parameter description:
    Start of irrigation period.

```
< month day
  (hour 8) >
```

    - *month*: integer
      Parameter
      Month in the year.
    - *day*: integer
      Parameter
      Day in the month.
    - *hour*: integer
      Parameter (default 8)
      Hour in the day.
  - *to*: submodel (see section 1.4.7)
    Submodel (has partially specified default value)

```
(to <missing month> <missing day>)
```

    Parameter description:
    End of irrigation period.

```
< month day
  (hour 8) >
```

    - *month*: integer
      Parameter
      Month in the year.
    - *day*: integer
      Parameter
      Day in the month.
    - *hour*: integer
      Parameter (default 8)
      Hour in the day.

  - *potential*: number [**cm**]
    Parameter
    Soil potential at which to irrigate.

- *irrigation_year*: integer
  Optional state variable
  Year management started. Negative number means it hasn't started yet.

- *irrigation_delay*: integer
  Optional state variable
  Hours we test for irrigation again. This is set at each irrigation, to avoid multiple applications.

9.30 sequence

Perform all the specified actions in the sequence listed. Each action is performed until done. At most one action can be performed at each time step.

- *do*: **action** component (see chapter 9) sequence
  Component (default: an empty sequence)
  Sequence of actions to perform.

9.31 activity

A 'sequence' model (see 9.30, page 78) build into Daisy.

9.32 wait

Wait until the specified condition is true.

- *condition*: **condition** component (see chapter 21)
  Condition to wait for.

9.33 wait_period

Waits the specified period.

- *hours*: integer
  Parameter
  Wait this number of hours.

- *days*: integer
  Parameter
  Wait this number of days.

- *end_time*: **Time** fixed component (see section 85.21)
  Optional submodel
  Wait until this date.Setting this overrides the 'days' and 'hours' parameters.

9.34 wait_days

A 'wait_period' model (see 9.33, page 78) build into Daisy.
Waits the specified number of days.

9.35 wait_hours

A 'wait_period' model (see 9.33, page 78) build into Daisy.

Waits the specified number of hours.

9.36 wait_mm_dd

Wait until a specific month and day in the year.

- *month*: integer
  Parameter
  Wait until this month.

- *day*: integer
  Parameter
  Wait until this day in the month.

- *hour*: integer
  Parameter (default 8)
  Wait until this hour.

9.37 while

Perform all the specified actions in the sequence listed, but in the same timestep. The 'while' action is done when the first action in the list is done.

- *actions*: **action** component (see chapter 9) sequence
  List of actions to perform.

9.38 table

Read management actions from a Daisy data file.

After the ddf header, the following column tags are recognized (with the dimension for the dimension linein square brackets).

Date [date]: The date for fertilization or irrigation.

Planting [date]: The content should be a date in yyyy-mm-dd format, where the operation specified by the 'sow' attribute will be perfomed.

Emerging [date]: The content should be a date in yyyy-mm-dd format, where the operation specified by the 'emerge' attribute will be perfomed.

Harvest [date]: The content should be a date in yyyy-mm-dd format, where the operation specified by the 'harvest' attribute will be perfomed.

Irrigate [mm]: The content should be an irrigation amount, that will be applied as overhead irrigation for the date specified in the 'Date' field. You can disable it with the 'enable_irrigation' attribute.

Fertilize [kg N/ha]: The content should be an amount of nitrogen fertilizer to be applied on the date specified in the 'Date' field. The fertilizer type will be either the one specified in the 'Fertilizer' column, or the 'fertilizer' attribute. You can disable it with the 'enable_fertilization' attribute.

Fertilizer [name]: The type of fertilizer to be applied.

- *emerge*: **action** component (see chapter 9)
  Optional component
  Emerge action.

- *harvest*: **action** component (see chapter 9)
  Optional component
  Harvest action.

- *sow*: **action** component (see chapter 9)
  Optional component
  Sow action.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to add irritaion.

- *flux*: number [**mm/h**]
  Parameter (default 2)
  Water application speed.

- *from*: number [**cm**]
  Parameter (default -5)
  Height where you want to start the incorporation (a negative number). OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Parameter (default -25)
  Height where you want to end the incorporation (a negative number). OBSOLETE: Use (volume box (bottom TO)) instead.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *fertilizer*: **am** component (see chapter 11)
  Optional component
  The fertilizer you want to apply.

- *enable_irrigation*: boolean (see section 1.4.2)
  Parameter (default true)
  Set this to false to ignore any irrigation information in the file.

- *enable_fertilization*: boolean (see section 1.4.2)
  Parameter (default true)
  Set this to false to ignore any fertilization information in the file.

9.39 assert

Assert that condition is true, if not, stop the simulation.

- *condition*: **condition** component (see chapter 21)
  Condition to check.

- *message*: string (see section 1.4.5)
  Parameter (has default value with length 32)

  ```
  (message "Required condition not fulfilled")
  ```

  Parameter description:
  Error message to give iff assertion fails.

9.40 message

Write a message to the user.

- *message*: string (see section 1.4.5)
  Parameter
  Message to give to the user.

9.41 warning

Write a warning to the user.

- *message*: string (see section 1.4.5)
  Parameter
  Warning to give to the user.

9.42 error

Write a error message to the user.

- *message*: string (see section 1.4.5)
  Parameter
  Error message to give.

9.43 panic

Write a error message to the user and stop the simulation.

- *message*: string (see section 1.4.5)
  Parameter
  Error message to give.

9.44 set_heat_source

Set external point heat source at height to value.

- *height*: number [**cm**]
  Parameter
  Height of heat source (a negative number).

- *value*: number [$\mathbf{W/m^2}$]
  Parameter
  Value of heat source.

9.45 ridge

Ridge a ridge on the field.

- *ridge*: **Ridge** fixed component (see section 85.12)
  Submodel (has partially specified default value)
  Ridge parameters

9.46 repeat

Perform all of the specified action. When done, repeat the action. The action may take several timesteps.

- *repeat*: **action** component (see chapter 9)
  Action to perform repeatedly.

- *do*: **action** component (see chapter 9)
  Optional component
  Action currently being performed.

9.47 fertilize_base

Shared parameters for all fertilize actions.

- *am*: **am** component (see chapter 11)
  The fertilizer you want to apply.

- *equivalent_weight*: number [**kg N/ha**]
  Optional parameter
  When fertilizing with organic matter, you may let Daisy calculate the amount of dry matter that corresponds to the specified amount of nitrogen. This requires that the fertilizer has specified the 'first_year_utilization' parameter, but not the 'weight' parameter.

- *minimum_weight*: number [**kg N/ha**]
  Parameter (default 0)
  Minimum amount of nitrogen to fertilize with.

- *precision*: submodel (see section 1.4.7)
  Optional submodel
  Let the amount of fertilizer depend on the inorganic nitrogen in the soil. The amount of fertilizer will be the specified 'target', minus the amount already present in the soil zone between 'from' and 'to'.

  ```
  < target
    (from 0 [cm])
    (to -100 [cm]) >
  ```

  - *target*: number [**kg N/ha**]
    Parameter
    How much N you want.
  - *from*: number [**cm**]
    Parameter (default 0)
    Height where you want to start measuring (a negative number).
  - *to*: number [**cm**]
    Parameter (default -100)
    Height where you want to end measuring (a negative number).

- *second_year_compensation*: boolean (see section 1.4.2)
  Parameter (default false)
  Compensate for the second year effect of previous fertilizations. The second year effect is solely governed by the 'second_year_utilization' organic fertilizer parameter. The second year effect does not fade with time, but is zeroed once you fertilize with this flag set.

9.48 fertilize

A 'fertilize_base' model (see 9.47, page 82) build into Daisy.
Apply fertilizer to the soil surface.

- *am*: **am** component (see chapter 11)
  The fertilizer you want to apply.

- *from*: number [**cm**]
  Parameter (default 0)
  Height where you want to start the incorporation (a negative number) OBSOLETE: Use 'fertilize_incorporate' instead.

- *to*: number [**cm**]
  Parameter (default 0)
  Height where you want to end the incorporation (a negative number) OBSOLETE: Use 'fertilize_incorporate' instead.

9.49 incorporate_fertilizer

A 'fertilize_base' model (see 9.47, page 82) build into Daisy.
Incorporate fertilizer.

- *am*: **am** component (see chapter 11)
  The fertilizer you want to apply.

- *volume*: **volume** component (see chapter 81)
  Soil volume to incorporate fertilizer in.

9.50 markvand

Irrigate the field according to MARKVAND scheduling.

- *soil*: **MV_Soil** component (see chapter 6)
  Soil type to schedule irrigation on.
- *flux*: number [**mm/h**]
  Parameter (default 2)
  Water application speed.
- *map*: submodel (see section 1.4.7) sequence
  Map of Daisy crop names into MARKVAND crop descriptions.

  < *Daisy MARKVAND* >

  - *Daisy*: string (see section 1.4.5)
    Parameter
    Name of Daisy crop.
  - *MARKVAND*: **MV_Crop** component (see chapter 5)
    MARKVAND crop description.
- *T_sum*: number [**dg C d**]
  Optional state variable
  Temperature sum since emergence.
- *dt*: number [**d**]
  Optional state variable
  Days since emergence.
- *V_I*: number [**mm**]
  Optional state variable (default 0)
  Amount of water intercepter by leaves.
- *V_r*: number [**mm**]
  Optional state variable
  Amount of available water in root zone. By default, the reservoir will be full at plant emergence.
- *V_e*: number [**mm**]
  Optional state variable
  Amount of available water in top soil reservoir. This is the water that can be extracted by soil evaporation. Included in 'V_r'. By default, the reservoir will be full at plant emergence.
- *C_u*: number [**mm**]
  Optional state variable (default 0)
  Capacity of available water in upper root zone.
- *V_u*: number [**mm**]
  Optional state variable (default 0)
  Amount of available water in upper root zone. Included in 'V_r'.
- *V_b*: number [**mm**]
  Optional state variable
  Amount of water between current and max root depth. By default, the reservoir will be full at plant emergence.

9.51 extern

Select an external scope, and perform action.

- *scope*: **scopesel** component (see chapter 59)
  Scope to evaluate expessions in.

- *action*: **action** component (see chapter 9)
  Action to perform if the condition is false.

9.52 extern_fertigation

Continiues irrigation with mineral nitrogen mix.

If the nitrogen amount is non-zero, it will be applied in the irrigation water if available, and otherwise be spread on the soil surface.

- *NO3*: **number** component (see chapter 45)
  Amount of NO3 in irrigation.

- *NH4*: **number** component (see chapter 45)
  Amount of NH4 in irrigation.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to add irritaion.

- *scope*: **scopesel** component (see chapter 59)
  Component (default 'null')
  Scope to evaluate expessions in.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *from*: number [**cm**]
  Parameter (default -10)
  Height where you want to start the incorporation (a negative number).

- *to*: number [**cm**]
  Parameter
  Height where you want to end the incorporation (a negative number).

- *surface*: **number** component (see chapter 45)
  Amount of surface irrigation applied.

- *overhead*: **number** component (see chapter 45)
  Amount of overhead irrigation applied.

- *subsoil*: **number** component (see chapter 45)
  Amount of subsoil irrigation applied.

9.53 extern_subsoil

Subsoil irrigation controlled externally.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to add irritaion.

- *flux*: **number** component (see chapter 45)
  Amount of irrigation applied.

- *scope*: **scopesel** component (see chapter 59)
  Scope to evaluate expessions in.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *from*: number [**cm**]
  Optional parameter
  Height where you want to start the incorporation (a negative number). OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Optional parameter
  Height where you want to end the incorporation (a negative number). OBSOLETE: Use (volume box (bottom TO)) instead.

- *constituents*: string (see section 1.4.5) sequence
  Parameter
  List of solutes to add to the irrigation water. The values are taken from the external scope, dimensions must be convertible to ppm.

Chapter 10

adsorption

This component describes the adsorption of a chemical to the soil, which among other things affects how large a fraction can be transported with the water.

10.1 none

No adsorption. Used for solutes that are not adsorped to the soil.

Used by chemical default adsorption (see 17.1, page 115) .

10.2 linear

M = rho K C + Theta C

Used by chemical NH4 adsorption (see 17.5, page 122) , chemical Atrazine adsorption (see 17.10, page 123) , chemical Bentazon adsorption (see 17.11, page 123) , chemical IPU adsorption (see 17.12, page 123) , chemical MCPP adsorption (see 17.13, page 123) , chemical Pendimethalin adsorption (see 17.14, page 123) , chemical Ioxynil adsorption (see 17.15, page 123) , chemical 2,4-D adsorption (see 17.16, page 124) , chemical Heptachlor adsorption (see 17.18, page 124) , and chemical DDT adsorption (see 17.19, page 124) .

- *K_clay*: number [$\mathbf{cm^3/g}$]
  Optional parameter
  Clay dependent distribution parameter. It is multiplied with the soil clay fraction to get the clay part of the 'K' factor. If 'K_OC' is specified, 'K_clay' defaults to 0.

- *K_OC*: number [$\mathbf{cm^3/g}$]
  Optional parameter
  Humus dependent distribution parameter. It is multiplied with the soil organic carbon fraction to get the carbon part of the 'K' factor. By default, 'K_OC' is equal to 'K_clay'.

10.3 NH4

A 'linear' model (see 10.2, page 87) build into Daisy.

Adsorption of ammonium.

10.4 Langmuir

M = rho (my_max C) / (K + C) + Theta C

- *K*: number [$\mathbf{g/cm^3}$]
  Parameter
  Half saturation constant.

- *my_max_clay*: number [$\mathbf{g/cm^3}$]
  Optional parameter
  Max adsorption capacity (clay). It is multiplied with the soil clay fraction to get the clay part of 'my_max'. If 'my_max_OC' is specified, 'my_max_clay' defaults to 0.

- *my_max_OC*: number [$\mathbf{g/cm^3}$]
  Optional parameter
  Max adsorption capacity (humus). It is multiplied with the soil organic carbon fraction to get the carbon part of 'my_max'. By default, 'my_max_OC' is equal to 'my_max_clay'.

10.5 Freundlich

M = rho K C^m + Theta C

- *m*: number (dimensionless)
  Parameter
  Freundlich parameter

- *K_clay*: number [$\mathbf{(g/cm^3)^{-m}}$]
  Optional parameter
  Clay dependent distribution parameter. It is multiplied with the soil clay fraction to get the clay part of the 'K' factor. If 'K_OC' is specified, 'K_clay' defaults to 0. The dimension depends on the 'm' parameter.

- *K_OC*: number [$\mathbf{(g/cm^3)^{-m}}$]
  Optional parameter
  Humus dependent distribution parameter. It is multiplied with the soil organic carbon fraction to get the carbon part of the 'K' factor. By default, 'K_OC' is equal to 'K_clay'. The dimension depends on the 'm' parameter.

10.6 vS_S

Model by van Schouwenberg and Schuffelen, 1963, with parameterization by Hansen et.al., 1990.

10.7 full

Full adsorption. Used for non-solutes, fully adsorped in the soil.
Used by chemical solid adsorption (see 17.20, page 124) .

Chapter 11

am

The 'am' component describes various kinds of fertilizer and other added matter such as crop residues. In particular, it describes how they decompose.

11.1 base

Common attributes for all added organic matter models.

- *name*: string (see section 1.4.5)
  Optional state variable
  A name given to this AOM so you can identify it in for example log files.

- *initialized*: boolean (see section 1.4.2)
  State variable (default false)
  True if this AM has been initialized. It will usually be false in user setup files, but true in checkpoints.

- *creation*: **Time** fixed component (see section 85.21)
  Optional submodel
  Time this AM was created.

- *lock*: submodel (see section 1.4.7)
  Optional submodel
  This AM belongs to a still living plant

  < `(crop` *crop*`)`
  `(part` *part*`)` >

  - *crop*: string (see section 1.4.5)
    State variable
    Crop to which this am is locked

  - *part*: string (see section 1.4.5)
    State variable
    Crop part to which this am is locked

- *om*: **AOM** component (see chapter 3) sequence
  Optional component
  The individual AOM pools.

11.2 organic

A 'base' model (see 11.1, page 89) build into Daisy.
Organic fertilizer, typically slurry or manure from animals.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.
- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.
- *weight*: number [**Mg w.w./ha**]
  Parameter (default 0)
  Amount of fertilizer applied.
- *total_C_fraction*: number [<**fraction**>]
  Parameter
  Carbon fraction of dry matter.
- *total_N_fraction*: number [<**fraction**>]
  Parameter
  Nitrogen fraction of dry matter.
- *second_year_utilization*: number [<**fraction**>]
  Optional parameter
  Estimated useful N fraction for the second year. In Denmark, this is governed by legalisation.
- *first_year_utilization*: number [<**fraction**>]
  Optional parameter
  Estimated useful N fraction for the first year. In Denmark, this is governed by legalisation.
- *dry_matter_fraction*: number [<**fraction**>]
  Parameter
  Dry matter fraction of total (wet) weight.
- *NO3_fraction*: number [<**fraction**>]
  Parameter (default 0)
  Nitrate fraction of total N in fertilizer. The remaining nitrogen is assumed to be ammonium or organic.
- *NH4_fraction*: number [<**fraction**>]
  Parameter (default 0)
  Ammonium fraction of total N in fertilizer. The remaining nitrogen is assumed to be nitrate or organic.
- *volatilization*: number [<**fraction**>]
  Parameter (default 0)
  Fraction of NH4 that evaporates on application.

11.3 slurry

A 'organic' model (see 11.2, page 90) defined in 'fertilizer.dai'.

Average based on numbers provided by Torben Bonde, Danish Environmental Protection Agency, approximately 1991. Added by <sha@kvl.dk>, 2000.

11.4 cattle_slurry

A 'slurry' model (see 11.3, page 90) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

11.5 pig_slurry

A 'slurry' model (see 11.3, page 90) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

11.6 manure

A 'slurry' model (see 11.3, page 90) defined in 'fertilizer.dai'.
We have no specific information about manure.

11.7 cattle_manure

A 'manure' model (see 11.6, page 91) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

11.8 pig_manure

A 'manure' model (see 11.6, page 91) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

11.9 horse_manure

A 'manure' model (see 11.6, page 91) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

11.10 Foulum_slurry

A 'slurry' model (see 11.3, page 90) defined in 'fertilizer.dai'.
Parameters provided by DJF in Foulum, 1998. Added by <sha@kvl.dk>, 2000.

11.11 kvaeg_gylle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.12 kvaeg_staldgoedning

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.13 kvaeg_ajle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.14 kvaeg_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.15 slagtesvin_gylle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.16 slagtesvin_staldgoedning

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.17 slagtesvin_ajle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.18 slagtesvin_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.19 soer_gylle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.20 soer_staldgoedning

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.21 soer_ajle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.22 soer_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.23 slagtekyllinger_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.24 hoens_gylle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.25 hoens_staldgoedning

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.26 hoens_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.27 kalkuner_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.28 aender_gaes_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.29 mink_raev_staldgoedning

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.30 mink_raev_gylle

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.31 hest_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.32 faar_geder_dybstroelse

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normtal 2000.

11.33 spildevandsslam

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Miljstyrelsen (2001). Spildevandsslam fra kommunale og private renseanlg i 1999. Orientering fra Miljstyrelsen Nr. 3. Jens Petersen (2001). Gdningsvrdi af organisk affald. JordbrugsForskning nr. 7, september 2001. Claus Petersen (2001). Statistik for jordbrugsmssig anvendelse. Miljprojekt Nr. 621 2001

11.34 komposteret_husholdningsaffald

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Jens Petersen (2001). Gdningsvrdi af organisk affald. JordbrugsForskning nr. 7, september 2001. Ann Marie Eilersen, Jens Chr. Tjell og Mogens Henze (2001). Muligheder for jordbrugsanvendelse af affald fra husholdninger. NUTRAP rapport, http://www.agsci.kvl.dk/nutrap. Torsten Mller og Jakob Magid (2001). Jordbrugets anvendelse af byaffald i Nord og Central Europa. NUTRAP rapport, http://www.agsci.kvl.dk/nutrap

11.35 kartoffelfrugtsaft

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Dansk Procesteknologi I/S (2003). RAPPORT TIL KARTOFFELAFGIFTSFONDEN: KALIUM I FRUGTSAFT - OVERSIGT. LSNINGSMETODER M.V. (Feb. 2003)

11.36 novogro_slam

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Novozymes, Kalundborg: green accounts. http://www.novonordisk.com/reports/press/environmental/er97/s

11.37 novogro-30_slam

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Novozymes, Kalundborg: green accounts. http://www.novonordisk.com/reports/press/environmental/er97/s

11.38 pressesaft_groentpillefabrik

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Albert Baumann (2003) Pers. medd. se side (Grntpiller) <abdangr@post.tele.dk>

11.39 svinegylle_usepareret

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.40 svinegylle_separet_tyk

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.41 svinegylle_separet_tynd

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.42 kvaeggylle_usepareret

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.43 kvaeggylle_separet_tyk

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.
Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.44 kvaeggylle_separet_tynd

A 'slurry' model (see 11.3, page 90) defined in 'dk-fertilizer.dai'.

Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringsstoffraktioner fra separeringsprodukter. Indlg p Efterrskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separet kvg- og svinegylle. Specialeprojekt, KVL.

11.45 root

A 'base' model (see 11.1, page 89) build into Daisy.

Initialization of old root remains.

Used by organic default am (see 46.2, page 243) .

- *depth*: number [**cm**]
  Optional parameter
  How far down does the old root reach? (a negative number) By default, the soils maximal rooting depth will be used.
- *weight*: number [**Mg DM/ha**]
  Parameter (default 1.2)
  Total weight of old root dry matter.
- *dist*: number [**cm**]
  Parameter (default 7)
  Distance to go down in order to decrease the root density to half the original.
- *total_C_fraction*: number [<**fraction**>]
  Parameter (default 0.4)
  Carbon fraction of total root dry matter
- *total_N_fraction*: number [<**fraction**>]
  Parameter (default 0.01)
  Nitrogen fraction of total root dry matter

11.46 state

A 'base' model (see 11.1, page 89) build into Daisy.

Most AM models are only used for initialization, they will be comnverted to this generic model after creation, so this is what you will see in a checkpoint. This model contains a number (typically 2) of separate pools, each of which have their own turnover rate.

11.47 initial

A 'base' model (see 11.1, page 89) build into Daisy.

Initial added organic matter at the start of the simulation.

- *layers*: submodel (see section 1.4.7) sequence
  Carbon content in different soil layers. The carbon is assumed to be uniformly distributed in each layer.

  < *end weight* >

  - *end*: number [**cm**]
    Parameter
    Height where this layer ends (a negative number).

  - *weight*: number [**kg C/m**^{2}]
    Parameter
    Carbon in this layer.

11.48 mineral

Mineral fertilizer.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *weight*: number [**kg N/ha**]
  Parameter (default 0)
  Amount of fertilizer applied.

- *NH4_fraction*: number [<**fraction**>]
  Parameter
  Ammonium fraction of total N in fertilizer. The remaining nitrogen is assumed to be nitrate.

- *volatilization*: number [<**fraction**>]
  Parameter (default 0)
  Fraction of NH4 that evaporates on application.

11.49 Ammonia

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Pure NH4. Added by <sha@kvl.dk>, 2000.

11.50 AmmoniumNitrate

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
A 50-50 mix of NH4 and NO3. Added by <sha@kvl.dk>, 2000.

11.51 Nitrate

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Pure NO3.

11.52 N25S

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
From _Kemira '97_. Added by <sha@kvl.dk>, 2000.

11.53 NPK01

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Various NPK and NS fertilizers

11.54 NPK02

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Various NPK fertilizers

11.55 NP

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Typical NP or NPS fertilizer

11.56 CalciumNitrate

A 'mineral' model (see 11.48, page 97) defined in 'fertilizer.dai'.
Kalksalpeter

Chapter 12

average

Find the average of two numbers.

12.1 arithmetic

Arithmetic average '(a+b)/2'.
    Used by uz1d richards K_average (see 76.2, page 373) .

12.2 harmonic

Harmonic average '2ab/(a+b)'.

12.3 geometric

Geometric average 'sqrt(a*b)'.

Chapter 13

bioclimate

The 'bioclimate' component is responsible for distributing the water and energy provided by the weather component among the crops and soil for a given column.

13.1 default

The default bioclimate model.
Used by column default Bioclimate (see 19.1, page 130) .

- *svat*: **svat** component (see chapter 69)
  Component (default 'none')
  Soil Vegetation Atmosphere component.

- *pet*: **pet** component (see chapter 48)
  Optional component
  Potential Evapotranspiration component.

  Some pet models provide answers for both dry and wet surface. For those, the wet answer will limit total evapotranspiration, while the dry answer will further limit transpiration.

  The default model depends on available climate date.

  If reference evaporation is available in the climate data, Daisy will use these (the weather pet model).

  If vapor pressure and wind are available, it will use Penman-Monteith (PM).

  If the timestep is larger than 12, and daily minimum and maximum temperature are available, Samani and Hargreaves (Hargreaves).

  As a last resort, Makkink (makkink) will be used.

- *difrad*: **difrad** component (see chapter 24)
  Optional component
  Diffuse radiation component.

  By default, choose depending on available climate date.

  If diffuse radiation is available in the climate data, Daisy will use these (the weather difrad model). Otherwise Daisy wil use the DPF model.

- *raddist*: **raddist** component (see chapter 53)
  Component (default 'default')
  Radiation distribution model.

- *Snow*: **Snow** fixed component (see section 85.5)
  Submodel (has fully specified default value)
  Surface snow pack.

- *net_radiation*: **net_radiation** component (see chapter 43)
  Component (default 'brunt')
  Net radiation.

- *NoOfIntervals*: integer
  Parameter (default 30)
  Number of vertical intervals in which we partition the canopy.

- *irrigation_subsoil_permanent*: number [**mm/h**]
  State variable (default 0)
  Long term irrigation below soil surface.

- *canopy_water_storage*: number [**mm**]
  State variable (default 0)
  Intercepted water on canopy.

- *litter_water_storage*: number [**mm**]
  State variable (default 0)
  Intercepted water on litter.

- *max_svat_iterations*: integer
  Parameter (default 100)
  Max number of svat iterations before giving up on cobvergence.

- *max_svat_absolute_difference*: number [**mm/h**]
  Parameter (default 0.01)
  Maximum absolute difference in svat ea values for convergence.

- *maxTdiff*: number [**K**]
  Parameter (default 1)
  Largest temperature difference for convergence.

- *maxEdiff*: number [**Pa**]
  Parameter (default 5)
  Largest humidity difference for convergence.

- *min_sun_angle*: number [**rad**]
  Parameter (default 0.0628319)
  Minimum sun angle above ground for some 'raddist' and 'svat' models.

  The 'DPF' raddist model will zero radiation if the angle is below this, and the 'SSOC' svat model will revert to a one leaf description.

Log Variables

- *albedo*: number (dimensionless)
  Reflection factor.

- *Height*: number [**cm**] canopy boundaries
  End points of canopy layers. First entry is top of canopy, last is soil surface.

- *total_ep*: number [**mm/h**]
  Potential evapotranspiration.

- *total_ea*: number [**mm/h**]
  Actual evapotranspiration.

- *direct_rain*: number [**mm/h**]
  Rain hitting surface directly. This includes rain hitting ponded water or litter, but excludes rain hitting canopy or snow, as well as snow and all forms for irrigation. The intended use is colloid generation.
- *irrigation_overhead*: number [**mm/h**]
  Irrigation above canopy.
- *irrigation_overhead_temperature*: number [**dg C**]
  Water temperature.
- *irrigation_surface*: number [**mm/h**]
  Irrigation below canopy.
- *irrigation_surface_temperature*: number [**dg C**]
  Water temperature.
- *irrigation_subsoil*: number [**mm/h**]
  Irrigation below soil surface this hour.
- *irrigation_total*: number [**mm/h**]
  Total irrigation above of below the soil surface.
- *tillage_water*: number [**mm/h**]
  Water added to surface due to tillage operations.
- *snow_ep*: number [**mm/h**]
  Potential snow evaporation.
- *snow_ea*: number [**mm/h**]
  Actual snow evaporation.
- *snow_water_in*: number [**mm/h**]
  Water entering snow pack.
- *snow_water_in_temperature*: number [**dg C**]
  Temperature of water entering snow pack.
- *snow_water_out*: number [**mm/h**]
  Water leaving snow pack
- *snow_water_out_temperature*: number [**dg C**]
  Temperature of water leaving snow pack.
- *canopy_ep*: number [**mm/h**]
  Potential canopy evaporation.
- *canopy_ea*: number [**mm/h**]
  Actual canopy evaporation.
- *canopy_water_capacity*: number [**mm**]
  Potential intercepted water on canopy.
- *canopy_water_temperature*: number [**dg C**]
  Temperature of incoming water.
- *canopy_water_in*: number [**mm/h**]
  Water entering canopy.
- *canopy_water_out*: number [**mm/h**]
  Canopy drip throughfall.

- *canopy_water_bypass*: number [**mm/h**]
  Water from above bypassing the canopy.

- *canopy_water_below*: number [**mm/h**]
  Total water input below canopy.

- *litter_ep*: number [**mm/h**]
  Potential evaporation litter.

- *litter_ea*: number [**mm/h**]
  Actual litter evaporation.

- *litter_water_capacity*: number [**mm**]
  Potential intercepted water on litter.

- *litter_water_temperature*: number [**dg C**]
  Temperature of incoming water.

- *litter_water_in*: number [**mm/h**]
  Water entering litter.

- *litter_water_out*: number [**mm/h**]
  Litter drip throughfall.

- *pond_ep*: number [**mm/h**]
  Potential evaporation from pond.

- *pond_ea*: number [**mm/h**]
  Actual evaporation from pond.

- *soil_ep*: number [**mm/h**]
  Potential exfiltration.

- *soil_ea*: number [**mm/h**]
  Actual exfiltration.

- *crop_ep*: number [**mm/h**]
  Potential transpiration. Transpiration under the assumption that the soil have an unlimited water supply. For a fully irrigated crop, this will be equal to the actual transpiration.

- *crop_ea_soil*: number [**mm/h**]
  Soil limited transpiration. The part of the potential transpiration that the soil can supply.

- *crop_ea_svat*: number [**mm/h**]
  Transpiration suggested by the SVAT module. Under stressed conditions, the soil, vegetation and atmosphere behave different than what was assumed when calculating the poterntial transpiration.

- *crop_ea*: number [**mm/h**]
  Actual transpiration. This is the transpiration limited either by what the soil can deliver, or what the SVAT module requires.

- *production_stress*: number (dimensionless)
  SVAT module induced stress, -1 means use water stress.

- *CanopyTemperature*: number [**dg C**]
  Actual canopy temperature.

- *SunLeafTemperature*: number [**dg C**]
  Sunlit leaf temperature.
- *ShadowLeafTemperature*: number [**dg C**]
  Shadow leaf temperature.
- *wind_speed_field*: number [**m/s**]
  Wind speed in the field at reference height.
- *wind_speed_weather*: number [**m/s**]
  Measured wind speed.
- *difrad0*: number [$\mathbf{W/m^2}$]
  Diffuse radiation above canopy.
- *total_PAR*: number [$\mathbf{W/m^2}$] canopy boundaries
  Total PAR between canopy layers.
- *sun_PAR*: number [$\mathbf{W/m^2}$] canopy boundaries
  Sun PAR between canopy layers.
- *total_NIR*: number [$\mathbf{W/m^2}$] canopy boundaries
  Total NIR between canopy layers.
- *sun_NIR*: number [$\mathbf{W/m^2}$] canopy boundaries
  Sun NIR between canopy layers.
- *sun_LAI_fraction*: number [<**fraction**>] canopy intervals
  Sunlit LAI in canopy layers.
- *absorbed_total_PAR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed PAR (sun+shade)
- *absorbed_total_NIR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed NIR (sun+shade)
- *absorbed_total_Long_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed long wave radiation (sun+shade)
- *absorbed_total_PAR_soil*: number [$\mathbf{W/m^2}$]
  Soil absorbed PAR (sun+shade)
- *absorbed_total_NIR_soil*: number [$\mathbf{W/m^2}$]
  Soil absorbed NIR (sun+shade)
- *absorbed_total_Long_soil*: number [$\mathbf{W/m^2}$]
  Soil absorbed long wave radiation (sun+shade)
- *absorbed_sun_PAR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed PAR on sunlit leaves
- *absorbed_sun_NIR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed NIR on sunlit leaves
- *absorbed_sun_Long_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed long wave radiatio on sunlit leaves
- *absorbed_shadow_PAR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed PAR on shadow leaves

- *absorbed_shadow_NIR_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed NIR on shadow leaves
- *absorbed_shadow_Long_canopy*: number [$\mathbf{W/m^2}$]
  Canopy absorbed long wave radiation on shadow leaves
- *incoming_Long_radiation*: number [$\mathbf{W/m^2}$]
  Incoming longwave radiation
- *incoming_PAR_radiation*: number [$\mathbf{W/m^2}$]
  Incoming PAR radiation
- *incoming_NIR_radiation*: number [$\mathbf{W/m^2}$]
  Incoming NIR radiation
- *incoming_Total_radiation*: number [$\mathbf{W/m^2}$]
  Incoming radiation, sum of shortwave and longwave

Chapter 14

biopore

A single class of biopores.

< component (height_start *height_start*)
(height_end *height_end*)
(density *density*)
(diameter *diameter*) >

- *height_start*: number [**cm**]
  Parameter
  Biopores starts at this depth (a negative number).

- *height_end*: number [**cm**]
  Parameter
  Biopores ends at this depth (a negative number).

- *density*: **number** component (see chapter 45)
  Biopore density [cm^{-2}] as a function of 'x' [cm].

- *diameter*: number [**cm**]
  Parameter
  Biopore diameter.

Log Variables

- S: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Sink from matrix domain to biopore.

- *infiltration*: number [**cm/h**]
  Surface infiltration.

- *solute_infiltration*: submodel (see section 1.4.7) sequence
  Rate of solute infiltration through surface.

  < *name value* > **Log Variables**

  - *value*: number [$\mathbf{g/cm^2/h}$]
    Value for chemical.
  - *name*: string (see section 1.4.5)
    Name of chemical.

14.1 matrix

Biopores that ends in the matrix.

- *solute*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Chemical concentration in biopore intervals.

  < *name value ...* >

  - *name*: string (see section 1.4.5)
    Optional state variable
    Name of chemical.
  - *value*: number [**g**] sequence
    Optional state variable
    Value for chemical.

- *debug*: integer
  Parameter (default 0)
  Debug level. Increase value to get more debug message.

- *max_iterations*: integer
  Parameter (default 50)
  Maximum number of iterations when seeking convergence.

- *max_absolute_difference*: number [**cm**]
  Parameter (default 0.02)
  Maximum absolute difference in biopore content for convergence.

- *max_relative_difference*: number (dimensionless)
  Parameter (default 0.001)
  Maximum relative difference in biopore content for convergence.

- *xplus*: number [**cm**] sequence
  Optional parameter
  Right side of each biopore interval. Water and chemical content is tracked individually for each interval. By default, use intervals as specified by the geometry.

- *K_wall_relative*: number (dimensionless)
  Parameter
  Relative conductivity of biopore wall compared to matrix.

- *h_bottom*: number [**cm**] sequence
  Optional state variable
  Pressure at the bottom of the biopores in each interval.

- *allow_upward_flow*: boolean (see section 1.4.2)
  Parameter (default true)
  Allow water to enter from saturated soil at the bottom of the biopore. And leave in unsaturated soil above.

Log Variables

- *water*: number [$\mathbf{cm}^3$]
  Water content.

- *iterations*: integer
  Number of iterations used for finding a solution.

- *h3_soil*: number [**cm**] sequence
  Pressure suggested by the soil for each interval.

- *z3_lowest*: number [**cm**] sequence
  Depth of lowest unsaturated cell in each interval. Water may not enter the macropore below this depth.

14.2 drain

Biopores that ends in the drain pipes.

- *pipe_position*: number [**cm**]
  Parameter
  Height pipes are placed in the soil (a negative number). By default, use the height specified for pipes in the column.

Chapter 15

boolean

Generic representation of booleans.

Used by StringerCondClause @ condition (see 85.30, page 447) , and IntegerCondClause @ condition (see 85.31, page 448) .

15.1 false

Always false.

15.2 true

Always true.

Used by source arithmetic valid (see 65.2, page 319) .

15.3 not

True if and only if the operand is not true.

- *operands*: **boolean** component (see chapter 15) array of length 1
  The operand to check.

15.4 string-equal

True iff the supplied strings are identical.

- *values*: string (see section 1.4.5) sequence
  Parameter
  Strings to compare.

15.5 numbers

Base class for boolean expressions involving numbers.

- *operands*: **number** component (see chapter 45) sequence
  List of operands to compare.

15.6 >

A 'numbers' model (see 15.5, page 111) build into Daisy.
True iff each operand is larger than the next.

15.7 >=

A 'numbers' model (see 15.5, page 111) build into Daisy.
True iff each operand is at least as large as the next.

15.8 <

A 'numbers' model (see 15.5, page 111) build into Daisy.
True iff each operand is smaller than the next.

15.9 <=

A 'numbers' model (see 15.5, page 111) build into Daisy.
True iff each operand is smaller than or equal to the next.

15.10 operands

Base class for boolean expressions involving multiple boolean operands.

- *operands*: **boolean** component (see chapter 15) sequence
  List of operands to compare.

15.11 or

A 'operands' model (see 15.10, page 112) build into Daisy.
True if and only if any operand is true.

15.12 and

A 'operands' model (see 15.10, page 112) build into Daisy.
True if and only if all operands are true.

15.13 xor

True if and only if one operand is true, and one false.

- *operands*: **boolean** component (see chapter 15) array of length 2
  The two operands to compare.

Chapter 16

bound

Specify one end of an interval boundary.

16.1 none

No boundary specified.

16.2 full

Maximum value for the interval boundary.

16.3 state

Bound used for checkpoints.
Used by volume box bottom (see 81.1, page 387) .

- *bound*: number [**cm**]
  Optional state variable
  Interval bound to use. Only valid for the 'finite' type.
- *type*: string (see section 1.4.5)
  State variable
  Bound type

16.4 empty

A 'state' model (see 16.3, page 113) build into Daisy.
A 'state' model set to 'none.

16.5 finite

Finite interval bound.

- *bound*: number [**cm**]
  Parameter
  Interval bound to use.

Chapter 17

chemical

This component should, for a specific chemical (typically a pesticide), provide a description of the properties of interest to Daisy.

17.1 default

Read chemical properties as normal Daisy parameters.

- *initial_C*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'C' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [**g/cm**^{3}]
    Parameter
    Concentration in water.

- *adsorption*: **adsorption** component (see chapter 10)
  Component (default 'none')
  Instant equilibrium between sorbed and solute phases.

  Specify the equilibrium model here for chemicals where the sorbed and solute phases typically reaches equilibrium within a single timestep. Slower adsorption processes should be modelled as two chemicals, one with 'none' adsorption and one with 'full' adsorption, and an 'adsorption' reaction between them.

- *initial*: **number** component (see chapter 45)
  Component (default 'const')

  ```
  (initial const 0 [g/cm^3])
  ```

  Parameter description:
  Initial content if otherwise unspecified. [g/cm^3]

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- C: number [$\mathbf{g/cm^3}$] soil cells
  Optional state variable
  Concentration in water.

- *crop_uptake_reflection_factor*: number [<**fraction**>]
  Parameter (default 1)
  How much of the chemical is reflected at crop uptake.

- *canopy_dissipation_rate*: number [$\mathbf{h^{-1}}$]
  Optional parameter
  How fast does the chemical dissipate on canopy. You must specify it with either 'canopy_dissipation_halftime' or 'canopy_dissipation_rate'.

- *canopy_dissipation_halftime*: number [**h**]
  Optional parameter
  How fast does the chemical dissipate on canopy. You must specify it with either 'canopy_dissipation_halftime' or 'canopy_dissipation_rate'.

- *canopy_dissipation_rate_coefficient*: number [$\mathbf{h^{-1}}$]
  Optional parameter
  Obsolete alias for 'canopy_dissipation_rate'.

- *canopy_washoff_coefficient*: number [<**fraction**>]
  Parameter
  Fraction of the chemical that follows the water off the canopy.

- *surface_decompose_rate*: number [$\mathbf{h^{-1}}$]
  Optional parameter
  How fast does the chemical decomposee on surface. You must specify it with either 'surface_decompose_halftime' or 'surface_decompose_rate'. If neither is specified, 'canopy_dissipation_rate' is used.

- *surface_decompose_halftime*: number [**h**]
  Optional parameter
  How fast does the chemical decompose on surface. You must specify it with either 'surface_decompose_halftime' or 'surface_decompose_rate'. If neither is specified, 'canopy_dissipation_rate' is used.

- *diffusion_coefficient*: number [$\mathbf{cm^2/s}$]
  Parameter
  Diffusion coefficient.

- *decompose_rate*: number [$\mathbf{h^{-1}}$]
  Optional parameter
  How fast the chemical is being decomposed in the soil. You must specify it with either 'decompose_rate' or 'decompose_halftime'.

- *decompose_halftime*: number [**h**]
  Optional parameter
  How fast the chemical is being decomposed in the soil. You must specify it with either 'decompose_rate' or 'decompose_halftime'.

- *decompose_heat_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 0 points)
  Heat factor on decomposition.

- *decompose_water_factor*: plf [**cm** → <**none**>]
  Parameter (has default value with 0 points)
  Water potential factor on decomposition.

- *decompose_CO2_factor*: plf [**g CO2-C/cm**^{3}**/h** → <**none**>]
  Parameter (has default value with 0 points)
  CO2 development factor on decomposition.

- *decompose_conc_factor*: plf [**g/cm**^{3} **H2O** → <**none**>]
  Parameter (has default value with 0 points)
  Concentration development factor on decomposition.

- *decompose_depth_factor*: plf [**cm** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (decompose_depth_factor (0 1) (1 1))
  ```

  Parameter description:
  Depth influence on decomposition.

- *decompose_lag_increment*: plf [**g/cm**^{3} → **h**$^{-1}$]
  Parameter (has default value with 0 points)
  Increment lag with the value of this PLF for the current concentration each hour. When lag in any cell reaches 1.0, decomposition begins. It can never be more than 1.0 or less than 0.0. By default, there is no lag.

- *drain_secondary*: boolean (see section 1.4.2)
  Parameter (default false)
  Concentration in secondary soil water user for drainage. If you set this to true the concentration in the secondary domain is used for concentration in drain water. Otherwise, the average concentration is the matix is used. Using the secondary domain is more physically correct, but also more likely to give unstable results.

- *C_below*: **number** component (see chapter 45)
  Component (default 'const')

  ```
  (C_below const -1 [g/cm^3])
  ```

  Parameter description:
  Concentration below the layer of soil being examined. Use a negative number to indicate same concentration as in lowest cell.

- *decompose_products*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  List of products from decomposition.

  < *fraction chemical* >

  - *fraction*: number [<**fraction**>]
    Parameter
    Fraction of decomposed matter that become this chemcial.
  - *chemical*: string (see section 1.4.5)
    Parameter
    Chemical product of decomposed matter.

- *snow_storage*: number [$\mathbf{g/m^2}$]
  State variable (default 0)
  Stored in the snow pack.

- *canopy_storage*: number [$\mathbf{g/m^2}$]
  State variable (default 0)
  Stored on the canopy.

- *litter_storage*: number [$\mathbf{g/m^2}$]
  State variable (default 0)
  Stored in the litter (mulch, surface residuals).

- *surface_storage*: number [$\mathbf{g/m^2}$]
  State variable (default 0)
  Stored on the soil surface. This includes the mixing layer, and constitute 'surface_solute' and 'surface_immobile'.

- *C_secondary*: number [$\mathbf{g/cm^3}$] soil cells
  Optional state variable
  Concentration in secondary domain.

- *initial_C_secondary*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'C_secondary' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [$\mathbf{g/cm^3}$]
    Parameter
    Concentration in secondary domain.

- *initial_C_primary*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'C_primary' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).

  - *value*: number [**g/cm**^{3}]
    Parameter
    Concentration in primary domain.

- *M*: number [**g/cm**^{3}] soil cells
  Optional state variable
  Total mass per volume water, soil, and air.

- *initial_M*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'M' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [**g/cm**^{3}]
    Parameter
    Total mass per volume water, soil, and air.

- *initial_M_primary*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'M_primary' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [**g/cm**^{3}]
    Parameter
    Primary domain mass per volume water, soil, and air.

- *Ms*: number [<**fraction**>] soil cells
  Optional parameter
  Mass in dry soil. This include all matter in both soil and water, relative to the dry matter weight. Only for initialization of the 'M' parameter.

- *initial_Ms*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'Ms' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).

  - *value*: number [<**fraction**>]
    Parameter
    Mass in dry soil. This include all matter in both soil and water, relative to the dry matter weight. Only for initialization of the 'M' parameter.

- *S_permanent*: number [**g/cm^3/h**] soil cells
  State variable (default: an empty sequence)
  Permanent external source, e.g. subsoil irrigation.

- *lag*: number (dimensionless) soil cells
  Optional state variable
  This state variable grows with lag_increment (C) each hour. When it reached 1.0, decomposition begins.

Log Variables

- *spray*: number [**g/m^2/h**]
  Amount currently being applied.

- *tillage*: number [**g/cm^3/h**] soil cells
  Changes during tillage.

- *dt*: number [**h**]
  Suggested timestep length based on sink terms.

- *deposit*: number [**g/m^2/h**]
  Amount deposited from the atmosphere.

- *surface_tillage*: number [**g/m^2/h**]
  Amount removed from surface due to tillage operations.

- *litter_tillage*: number [**g/m^2/h**]
  Amount removed from litter due to tillage operations.

- *snow_in*: number [**g/m^2/h**]
  Entering snow pack.

- *snow_out*: number [**g/m^2/h**]
  Leaking from snow pack.

- *canopy_in*: number [**g/m^2/h**]
  Entering canopy.

- *canopy_dissipate*: number [**g/m^2/h**]
  Dissipating from canopy.

- *canopy_out*: number [**g/m^2/h**]
  Falling through or off the canopy.

- *canopy_harvest*: number [**g/m^2/h**]
  Amount removed with crop harvest.

- *litter_in*: number [**g/m^2/h**]
  Entering litter .

- *litter_decompose*: number [**g/m^2/h**]
  Decomposed from the litter.

- *litter_out*: number [**g/m^2/h**]
  Leaking from litter.

- *surface_solute*: number [**g/m^2**]
  Stored in the soil water of the mixing layer. This is part of 'surface_storage'.

- *surface_immobile*: number [**g/m^2**]
  Bound to soil particles in the mixing layer. This is part of 'surface_storage'.

- *surface_in*: number [**g/m^2/h**]
  Falling on the bare soil surface.

- *surface_runoff*: number [**g/m^2/h**]
  Removed through lateral movement on the soil.

- *surface_decompose*: number [**g/m^2/h**]
  Decomposed from the surface.

- *surface_transform*: number [**g/m^2/h**]
  Added through transformation to surface.

- *surface_mixture*: number [**g/m^2/h**]
  Entering the soil through mixture with ponded water.

- *surface_out*: number [**g/m^2/h**]
  Entering the soil with water infiltration.

- *surface_release*: number [<**fraction**>]
  Fraction of available soil particles released as colloids this timestep. Only relevant for chemicals representing colloids.

  The idea behind this is that reactions that generate colloids will set the value of this variable, and then reactions that convert immobile chemicals into colloid bound chemicals will use it. For this to work, the reactions that set the variable must be listed before the reactions that us it.

  Note that the value is relative to the current timestep.

- *top_storage*: number [**g/m^2**]
  Sum of above ground (surface, liter, snow, canopy) storage.

- *top_loss*: number [**g/m^2/h**]
  Amount lost from the system from the surface. This includes runoff, canopy dissipation and harvest, but not soil infiltration. It also includes the net loss through transformation, which can be negative.

- *C_primary*: number [**g/cm^3**] soil cells
  Concentration in primary domain.

- *M_primary*: number [**g/cm^3**] soil cells
  Primary domain mass per volume water, soil, and air.

- *M_secondary*: number [**g/cm^3**] soil cells
  Mass in secondary domain.

- *M_error*: number [**g/cm^3**] soil cells
  Mass substracted to avoid negative values.

- *S_secondary*: number [**g/cm^3/h**] soil cells
  Secondary matrix source term.

- *S_primary*: number [**g/cm^3/h**] soil cells
  Primary matrix source term.

- *S_tertiary*: number [**g/cm^3/h**] soil cells
  Source term for tertiary (macropore) domain.
- *S_exchange*: number [**g/cm^3/h**] soil cells
  Exchange from primary to secondary domain.
- *S_drain*: number [**g/cm^3/h**] soil cells
  Source term (soil drainage only).
- *S_external*: number [**g/cm^3/h**] soil cells
  External source, such as incorporated fertilizer.
- *S_root*: number [**g/cm^3/h**] soil cells
  Source term (root uptake only, always negative).
- *S_decompose*: number [**g/cm^3/h**] soil cells
  Source term for decompose, is never positive.
- *S_transform*: number [**g/cm^3/h**] soil cells
  Source term for transformations other than sorption.
- *J_primary*: number [**g/cm^2/h**] soil edges
  Transportation in primary matrix water (positive up).
- *J_secondary*: number [**g/cm^2/h**] soil edges
  Transportation in secondary matrix water (positive up).
- *J_matrix*: number [**g/cm^2/h**] soil edges
  Transportation in matrix (positive up).
- *J_tertiary*: number [**g/cm^2/h**] soil edges
  Transportation in tertiary water (positive up).
- *sink_cell*: integer
  Cell with largest forward sink compared to available matter.

17.2 nutrient

A 'default' model (see 17.1, page 115) build into Daisy.
Plants eat this stuff.

17.3 N

A 'nutrient' model (see 17.2, page 122) build into Daisy.
Non-organic nitrogen.

17.4 NO3

A 'N' model (see 17.3, page 122) build into Daisy.
Nitrate-N.
Used by chemistry N trace (see 18.2, page 125) .

17.5 NH4

A 'N' model (see 17.3, page 122) build into Daisy.
Ammonium-N.
Used by chemistry N trace (see 18.2, page 125) .

17.6 common

A 'default' model (see 17.1, page 115) defined in 'chemistry-base.dai'.
For chemicals where we know no better. See also [FOCUS, 2002]

17.7 pesticide

A 'common' model (see 17.6, page 123) defined in 'chemistry-base.dai'.
This stuff protects plants. See also [FOCUS, 2002]

17.8 fungicide

A 'pesticide' model (see 17.7, page 123) defined in 'chemistry-base.dai'.
This stuff kills swamps. See also [FOCUS, 2002]

17.9 herbicide

A 'pesticide' model (see 17.7, page 123) defined in 'chemistry-base.dai'.
This stuff kills plants. See also [FOCUS, 2002]

17.10 Atrazine

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.11 Bentazon

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.12 IPU

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.13 MCPP

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.14 Pendimethalin

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.15 Ioxynil

A 'herbicide' model (see 17.9, page 123) defined in 'chemistry.dai'.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.16 2,4-D

A ‘herbicide’ model (see 17.9, page 123) defined in ‘chemistry.dai’.
Selective herbicide, kill weeds but not crops. See also [FOCUS, 2002]
Used by chemistry pesticides trace (see 18.3, page 125) .

17.17 insecticide

A ‘pesticide’ model (see 17.7, page 123) defined in ‘chemistry-base.dai’.
This stuff kills insects. See also [FOCUS, 2002]

17.18 Heptachlor

A ‘insecticide’ model (see 17.17, page 124) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.19 DDT

A ‘insecticide’ model (see 17.17, page 124) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 18.3, page 125) .

17.20 solid

A ‘default’ model (see 17.1, page 115) defined in ‘chemistry-base.dai’.
Non-dissolvable chemicals

Chapter 18

chemistry

Pesticides and other chemicals.
  < `component  (description` *`description`*`)`
  `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

18.1 default

Handle chemicals and reactions.

- *reaction*: **reaction** component (see chapter 55) sequence
  Component (default: an empty sequence)
  List of chemical reactions you want to simulate.

- *trace*: **chemical** component (see chapter 17) sequence
  Component (default: an empty sequence)
  List of chemicals you want to trace in the simulation.

18.2 N

A 'default' model (see 18.1, page 125) build into Daisy.
  Inorganic nitrogen.
  Used by chemistry nutrient combine (see 18.6, page 127) .

18.3 pesticides

A 'default' model (see 18.1, page 125) defined in 'chemistry.dai'.
  Trace all known pesticides.

18.4 multi

Handle multile chemistries.

Used by column default Chemistry (see 19.1, page 130) .

- *combine*: **chemistry** component (see chapter 18) sequence
  List of chemistry parameterizations you want to combine.

- *ignore*: string (see section 1.4.5) sequence
  State variable (default: an empty sequence)
  Don't warn when spraying one of these chemicals. The first time an untraced chemical not on the list is sprayed on the field, Daisy will issue a warning and add the chemical to this list.

- *max_sink_total*: number (dimensionless)
  Parameter (default 0.5)
  Maximum allowed sink term as a fraction of total content.

  If variable timesteps are enabled, Daisy will try to scale down the timestep in order to ensure that no more than this fraction of the total content is removed by drains or biopores within the timestep.

- *max_sink_solute*: number (dimensionless)
  Parameter (default 0.9)
  Maximum allowed sink term as a fraction of solute content.

  If variable timesteps are enabled, Daisy will try to scale down the timestep in order to ensure that no more than this fraction of the solute content is removed by drains or biopores within the timestep.

- *max_sink_secondary*: number (dimensionless)
  Parameter (default 1.5)
  Maximum allowed sink term as a fraction of secondary domain content.

  If variable timesteps are enabled, Daisy will try to scale down the timestep in order to ensure that no more than this fraction of the secondary domain content is removed by drains or biopores within the timestep. This should usually be above 1 to allow for the case where the secondary domain is emptied within a timestep.

- *min_sink_total*: number (dimensionless)
  Parameter (default 0.01)

  Always allow this fraction of total content to be removed by sink term.

  This overwrites all the 'max_sink' parameters.

Log Variables

- *trace*: **chemical** component (see chapter 17) sequence
  List of chemicals in nested chemistries.

18.5 none

A 'multi' model (see 18.4, page 126) build into Daisy.

No active chemistries.

18.6 nutrient

A ‘multi’ model (see 18.4, page 126) build into Daisy.

Include ’N’ chemistry so organic matter and plants will work.

Chapter 19

column

A 'column' is an one-dimensional vertical description of the soil/crop/atmosphere system. The column component contains most of the other processes in Daisy as submodels.

```
< component  (description description)
             (cite)
             (area 1 [m^2])
             (location) >
```

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *area*: number [$\mathbf{m}^2$]
  State variable (default 1)
  Area covered by this column. When logging multiple columns, the values are weighted by relative area.

- *location*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Location of this column.

  The meaning depends on the number of point in the sequence. 0 points: The column has no specific location. 1 point: The column has a location, but no specific area. 3 or more points: The column represents the area specified by a polygon with the specified corner points.

  ```
  <  x y  >
  ```

  - x: number (dimension not specified)
    Parameter
    X-Coordinate.

  - y: number (dimension not specified)
    Parameter
    Y-Coordinate.

19.1 default

Hansen et.al. 1990. with generic movement in soil.

- *weather*: **weather** component (see chapter 82)
  Optional component
  Weather model for providing climate information during the simulation. If unspecified, used global weather.

- *scope*: **scopesel** component (see chapter 59)
  Component (default 'null')
  Scope to evaluate expessions in.

- *SoilHeat*: **SoilHeat** fixed component (see section 85.6)
  Submodel (has fully specified default value)
  Soil heat capacity and transportation.

- *SoilWater*: **SoilWater** fixed component (see section 85.7)
  Submodel (has fully specified default value)
  Soil water content and transportation.

- *Surface*: **Surface** fixed component (see section 85.8)
  Submodel (has fully specified default value)
  The upper border of the soil.

- *Soil*: **Soil** fixed component (see section 85.11)
  Submodel (has partially specified default value)
  The numeric and physical soil properties.

- *Irrigation*: **Irrigation** fixed component (see section 85.29)
  Submodel (has fully specified default value)
  Active irrigation events.

- *Movement*: **movement** component (see chapter 42)
  Component (default 'vertical')
  Discretization and movement of water, heat and solutes in the soil.

- *Vegetation*: **vegetation** component (see chapter 79)
  Component (default 'crops')
  The crops on the field.

- *Litter*: **litter** component (see chapter 38)
  Component (default 'none')
  The litter layer below the canopy.

- *Bioclimate*: **bioclimate** component (see chapter 13)
  Component (default 'default')
  The water and energy distribution among the crops.

- *Drain*: **drain** component (see chapter 26)
  Component (default 'none')
  Drainage.

- *Groundwater*: **groundwater** component (see chapter 33)
  The groundwater level.

- *Chemistry*: **chemistry** component (see chapter 18)
  Component (default 'multi')

```
(Chemistry multi (combine N))
```

Parameter description:
Chemical compounds in the system.

- *OrganicMatter*: **organic** component (see chapter 46)
  Component (default 'default')
  The organic matter in the soil and on the surface.

- *second_year_utilization*: number [**kg N/ha**]
  State variable (default 0)
  Estimated accumulated second year fertilizer effect.

- *tillage_age*: number [**d**] soil cells
  Optional state variable
  Time since the latest tillage operation was performed. By default, the top 25 cm will have an initial tillage age of 100 days, while the soil below that will have an initial tillage age of 100 years. If you specify fewer values than there are soil cells, the last specified value will be used for the remaining cells.

Log Variables

- *harvest_DM*: number [$\mathbf{g/m^2/h}$]
  Amount of DM removed by harvest this hour.

- *harvest_N*: number [$\mathbf{g/m^2/h}$]
  Amount of nitrogen removed by harvest this hour.

- *harvest_C*: number [$\mathbf{g/m^2/h}$]
  Amount of carbon removed by harvest this hour.

- *residuals_DM*: number [$\mathbf{g/m^2/h}$]
  Amount of dry matter removed from crops to surface and soil this hour. This includes loss as harvest, as well as loss of old leaves and roots.

- *residuals_N_top*: number [$\mathbf{g/m^2/h}$]
  Amount of nitrogen removed from crops to soil this hour. This includes loss as harvest, as well as loss of old leaves.

- *residuals_C_top*: number [$\mathbf{g/m^2/h}$]
  Amount of carbon removed from crops to surface this hour. This includes loss as harvest, as well as loss of old leaves.

- *residuals_N_soil*: number [$\mathbf{g/cm^3/h}$] soil cells
  Amount of nitrogen removed from crops in soil this hour. This includes loss as harvest, as well as loss of old roots.

- *residuals_C_soil*: number [$\mathbf{g/cm^3/h}$] soil cells
  Amount of carbon removed from crops in soil this hour. This includes loss as harvest, as well as loss of old roots.

- *residuals_N_root*: number [$\mathbf{g/m^2/h}$]
  Amount of nitrogen removed from crops to soil this hour. This includes loss as harvest, as well as loss of old roots.

- *residuals_C_root*: number [$\mathbf{g/m^2/h}$]
  Amount of carbon removed from crops to surface this hour. This includes loss as harvest, as well as loss of old roots.

- *surface_water*: number [**mm**]
  Amount of water in the system above ground. This include ponded water, intercepted water and the snow pack.

- *seed_N*: number [**kg N/ha/h**]
  Amount of nitrogen in seed applied this time step.

- *seed_C*: number [**kg C/ha/h**]
  Amount of carbon in seed applied this time step.

- *applied_DM*: number [**ton DM/ha/h**]
  Amount of dry matter applied this time step. This includes dry matter incorporated directly in the soil.

- *first_year_utilization*: number [**kg N/ha/h**]
  Estimated first year fertilizer effect.

19.2 Askov

A 'default' model (see 19.1, page 130) defined in 'dk-soil.dai'.
This is a JB6 soil with free drainage macropores.

19.3 Jyndevad

A 'default' model (see 19.1, page 130) defined in 'dk-soil.dai'.
This is a JB1 soil with free drainage

19.4 Foulum

A 'default' model (see 19.1, page 130) defined in 'dk-soil.dai'.
Provided by DJF as part if the FertOrgaNic project.

Chapter 20

condedge

Find the hydraulic conductivity between two cells.

20.1 arithmetic

Use the arithmetic average of the conductivity in the two cells.
Used by uzrect Mollerup K_average (see 78.3, page 378) .

20.2 harmonic

Use harmonic average of the conductivity of the two cells. This corresponds to using the average hydraulic resistence.

20.3 geometric

Geometric average 'sqrt(a*b)'.

20.4 pressure

Pressure dependent average of the two cells. Use harmonic average of the conductivity of unsaturated cells. This corresponds to using the average hydraulic resistence. For saturated cells, water may stream into unsaturated neigbor cells with saturated conductivity if 'allow_sideways' is true. For cells where pressure is above 'h_lim', water may stream downward to dryer cell with a conductivity corresponding to 'h_lim'.

- *h_lim*: number [**cm**]
  Parameter
  Lower pressure limit for fast downward flow.
- *allow_sideways*: boolean (see section 1.4.2)
  Parameter (default true)
  Allow water to flow fast from saturated cells to all neigbor cells. Not just the cell below.
- *use_h_old*: boolean (see section 1.4.2)
  Parameter (default true)
  Use pressure at the start of the small timestep for enabling fast flow. If false, use the pressure at end of the small timestep.

Chapter 21

condition

A 'condition' component tests the state of the simulation, like whether the water pressure in a specific depth is above a given threshold. Logic conditions like 'and' and 'or' can be used for testing whether multiple conditions are fulfilled simultaneously.

Used by select component when (see 62, page 305) .

21.1 running

True iff the simulation is still running.

21.2 finished

True iff the simulation has finished.

Used by log checkpoint when (see 39.1, page 209) .

21.3 soil_temperature_above

Test if the soil is warmer than the specified temperature.

Used by condition trafficable operands (see 21.15, page 138) .

- *height*: number [**cm**]
  Parameter
  Soil depth in which to test the temperature.
- *temperature*: number [**dg C**]
  Parameter
  Lowest soil temperature for which the condition is true.

21.4 soil_water_pressure_above

Test if the soil is wetter than the specified pressure potential.

- *height*: number [**cm**]
  Parameter
  Depth at which to example the pressure potential.
- *potential*: number [**cm**]
  Parameter
  The soil should be wetter than this for the condition to be true.

21.5 soil_water_content_above

Test if the soil contains more water than the specified amount.

- *water*: number [**mm**]
  Parameter
  The soil should contain more water than this for the condition to be true.
- *from*: number [**cm**]
  Parameter (default 0)
  Top of interval to measure soil water content in.
- *to*: number [**cm**]
  Parameter
  Bottom of interval to measure soil water content in.

21.6 soil_inorganic_N_above

Test if the soil contains more mineral nitrogen than the specified amount.

- *amount*: number [**kg N/ha**]
  Parameter
  The soil should contain more inorganic nitrogen than this for the condition to be true.
- *from*: number [**cm**]
  Parameter (default 0)
  Top of interval to measure soil content in.
- *to*: number [**cm**]
  Parameter
  Bottom of interval to measure soil content in.

21.7 crop_ds_after

True iff the crop has reached development stage 'ds'.

- *crop*: string (see section 1.4.5)
  Parameter
  Name of crop on the field to test. Specify "all" to use combined weight of all crops on the field in test.
- *ds*: number (dimensionless)
  Parameter
  Development stage [-1.0:2.0].

21.8 crop_dm_over

True iff the crop has reached the specified amount of dry matter.

- *crop*: string (see section 1.4.5)
  Parameter
  Name of crop on the field to test.
- *weight*: number [**kg DM/ha**]
  Parameter
  Amount of non-root dry-matter required for the condition to be true.

- *height*: number [**cm**]
  Parameter (default 0)
  Height above which we measure the DM weight.

21.9 crop_dm_sorg_over

True iff the storage organ has reached the specified amount of dry matter.

- *crop*: string (see section 1.4.5)
  Parameter
  Name of crop on the field to test.
- *weight*: number [**kg DM/ha**]
  Parameter
  Amount of non-root dry-matter required for the condition to be true.

21.10 if

If the first condition is true, return the value of the second condition, else return the value of the third condition.

- *if*: **condition** component (see chapter 21)
  Condition to test for.
- *then*: **condition** component (see chapter 21)
  Condition to use of the 'if' test was true.
- *else*: **condition** component (see chapter 21)
  Condition to use if the 'if' test was false.

21.11 false

Always false.

21.12 true

Always true.
Used by program Daisy activate_output (see 52.12, page 273) .

21.13 or

True iff any of the listed conditions are true. The conditions are tested in the sequence listed, until a true is found, or the end of the list is reached.

- *operands*: **condition** component (see chapter 21) sequence
  Conditions to test.

21.14 and

True iff all the listed conditions are true. The conditions are tested in the sequence listed, until a false is found, or the end of the list is reached.

- *operands*: **condition** component (see chapter 21) sequence
  Conditions to test.

21.15 trafficable

A 'and' model (see 21.14, page 137) defined in 'tillage.dai'.

21.16 not

True iff the operand is not true.
Used by condition trafficable operands (see 21.15, page 138) .

- *operand*: **condition** component (see chapter 21)
  Condition to test.

21.17 mm_dd_base

Conditions based on month and day.

- *month*: integer
  Parameter
  Month to test for.

- *day*: integer
  Parameter
  Day in the month to test for.

- *hour*: integer
  Parameter (default 8)
  Hour to test for.

- *minute*: integer
  Parameter (default 0)
  Minute to test for.

- *second*: integer
  Parameter (default 0)
  Second to test for.

21.18 mm_dd

A 'mm_dd_base' model (see 21.17, page 138) build into Daisy.
True a specific month, day and hour in the year.

21.19 before_mm_dd

A 'mm_dd_base' model (see 21.17, page 138) build into Daisy.
True before specific month, day and hour in the year.

21.20 after_mm_dd

A 'mm_dd_base' model (see 21.17, page 138) build into Daisy.
True after specific month, day and hour in the year.

21.21 time

Conditions based on a specific time.

- *time*: **Time** fixed component (see section 85.21)
  Submodel (has partially specified default value)
  Fixed time to test for.

21.22 at

A 'time' model (see 21.21, page 139) build into Daisy.
  True, iff the simulation time is at the specified time.

21.23 before

A 'time' model (see 21.21, page 139) build into Daisy.
  True, iff the simulation time is before the specified time.

21.24 after

A 'time' model (see 21.21, page 139) build into Daisy.
  True, iff the simulation time is after the specified time.

21.25 hour

True, at the specified hour.

- *at*: integer
  Parameter
  Hour when the condition is true [0-23].

21.26 mday

True, at the specified day in the month.

- *at*: integer
  Parameter
  Day in the month when the condition is true [1-31].

21.27 yday

True, at the specified julian day.

- *at*: integer
  Parameter
  Julian day when the condition is true [1-366].

21.28 month

True, at the specified month.

- *at*: integer
  Parameter
  Month when the condition is true [1-12].

21.29 year

True, at the specified year.

- *at*: integer
  Parameter
  Year when the condition is true.

21.30 timestep

Add a timestep to a condition. It is true whenever 'operand' is true, but will let Daisy know what 'timestep' it represents. The timestep is used for the dimension in log files.

- *operand*: **condition** component (see chapter 21)
  Condition to use.
- *timestep*: string (see section 1.4.5)
  Parameter
  Timestep to use.

21.31 hourly

True at the end of each hour.

Used by log Soil water when (see 39.12, page 216) , log Field water when (see 39.13, page 217) , log Field nitrogen when (see 39.14, page 218) , log Soil nitrogen when (see 39.15, page 220) , log Field chemical when (see 39.8, page 212) , and log Soil chemical when (see 39.9, page 214) .

21.32 secondly

True at the end of each second.

21.33 minutely

True at the end of each minute.

21.34 daily

True at the end of each day.

21.35 weekly

True at the end of each week.

21.36 monthly

True at the end of each month.

21.37 yearly

True at the end of each year.

21.38 every

Matches simulation with fixed time intervals.

- *hours*: integer
  State variable (default 0)
  Number of hours.

- *days*: integer
  State variable (default 0)
  Number of days.

- *minutes*: integer
  State variable (default 0)
  Number of minutes.

- *seconds*: integer
  State variable (default 0)
  Number of seconds.

- *microseconds*: integer
  State variable (default 0)
  Number of microseconds.

- *next*: **Time** fixed component (see section 85.21)
  Optional submodel
  Time for next match.

21.39 TSum_above

Test if the temperature sum is above the specified value The temperature sum is the sum of the daily average air temperature since last reset. It is reset once a year. Days where the average is below 0 does not count in the sum.

- *TSum_limit*: number [**dg C d**]
  Parameter
  Temeperature sum above which the condition becomes true.

- *check_hour*: integer
  Parameter (default 6)
  Hour in day to update TSum.

- *reset_mday*: integer
  Parameter (default 1)
  Day in month to reset TSum.

- *reset_month*: integer
  Parameter (default 3)
  Month in year to reset TSum.

- *TSum_now*: number [**dg C d**]
  Optional state variable
  Current temeprature sum since last reset.

21.40 check

Test if a boolean expression is true.

- *expr*: **boolean** component (see chapter 15)
  Expression to evaluate.

21.41 extern

Test if a boolean expression is true,using extern log.

- *scope*: **scopesel** component (see chapter 59)
  Scope to evaluate expession in.

- *expr*: **boolean** component (see chapter 15)
  Expression to evaluate.

21.42 periodic

True if move than a specified walltime has passed since last time it was true.
Used by program Daisy print_time (see 52.12, page 273) .

- *period*: integer
  Parameter (default 1)
  Number of walltime seconds between success.

Chapter 22

crop

The 'crop' component simulates a specific crop on the field, typically averaged over one square meter, not individual plants. Of particular interest is water and nitrogen uptake at different depths, and the vertical leaf area distribution, which are used for competition with other crops.

< `component` `(description` *`description`*`)` `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

22.1 default

Standard Daisy crop model. Hansen, 1999.

- *Harvest*: **Harvesting** fixed component (see section 85.16)
  Submodel (has fully specified default value)
  Harvest parameters.

- *CrpN*: **CrpN** fixed component (see section 85.19)
  Submodel (has partially specified default value)
  Nitrogen parameters.

- *Seed*: **seed** component (see chapter 61)
  Component (default 'LAI')
  Initial crop growth.

- *Root*: **RootSystem** fixed component (see section 85.13)
  Submodel (has fully specified default value)
  Root system.

- *Canopy*: **CanopyStandard** fixed component (see section 85.14)
  Submodel (has partially specified default value)
  Canopy.

- *Prod*: **Production** fixed component (see section 85.17)
  Submodel (has partially specified default value)
  Production.

- *last_time*: **Time** fixed component (see section 85.21)
  Optional submodel
  The time of the previous timestep.

- *Devel*: **phenology** component (see chapter 49)
  Development and phenology.

- *Partit*: **Partition** fixed component (see section 85.18)
  Submodel (has partially specified default value)
  Assimilate partitioning.

- *Vernal*: **vernalization** component (see chapter 80)
  Component (default 'none')
  Vernalization.

- *LeafPhot*: **photosynthesis** component (see chapter 50)
  Component (default 'GL')
  Leaf photosynthesis. Note that if the selected radiation distribution model distinguishes between sunlit and shadow leaves, only the shadow leaves will be

- *water_stress_effect*: **wse** component (see chapter 83)
  Optional component
  Effect of water stress on production. By default, this will be 'none' iff the selected photosynthesis model does handle water stress implicitly, and 'full' otherwise.

- *enable_N_stress*: boolean (see section 1.4.2)
  Optional parameter
  Set this true to let nitrogen stress limit production. By default, it will be true iff the selected photosynthesis model does handle nitrogen stress implicitly.

- *min_light_fraction*: number [<**fraction**>]
  Parameter (default 0)

  When multiple crops are competing for light, this parameter specifies a minumum amount of the light this crop will receive. The idea is that the field has patches where one crop is dominating, as specified by this parameter, and in these patches the crop will not have to compete for light. The crop still needs LAI in order to catch the light though. Competition for water and nutrients are unaffected.

Log Variables

- *sunlit*: **photosynthesis** component (see chapter 50)
  Leaf photosynthesis for sunlit leaves. This will be zero if the selected radiation distribution model does not distinguish between sunlit and shadow leafs.

- *reserved*: **photosynthesis** component (see chapter 50)
  Leaf photosynthesis for reserved leaves. This is used for simulating "patches" in multi-crop systems, such as a clover-grass mixture. This is controled by the 'min_light_fraction' parameter.

22.2 Maize

A 'default' model (see 22.1, page 143) defined in 'maize.dai'.
Roskilde 74-76 ukendt varietet

22.3 Ikuwala Maize

A 'Maize' model (see 22.2, page 145) defined in 'maize.dai'.
SASA Project

22.4 Silage Maize

A 'Maize' model (see 22.2, page 145) defined in 'maize.dai'.
LG11

22.5 Pioneer Maize

A 'default' model (see 22.1, page 143) defined in 'maize.dai'.

22.6 Pea

A 'default' model (see 22.1, page 143) defined in 'pea.dai'.

22.7 Green Pea

A 'Pea' model (see 22.6, page 145) defined in 'pea.dai'.

22.8 Potato; Koege

A 'default' model (see 22.1, page 143) defined in 'potato.dai'.
Kge Project

22.9 Potato; SCRI

A 'default' model (see 22.1, page 143) defined in 'potato.dai'.
Potate parameterized by sha@kvl.dk – 2003. It is based on experimental data from SCRI from 1983 and 1984.

22.10 Potato; FertOrgaNic

A 'Potato; SCRI' model (see 22.9, page 145) defined in 'potato.dai'.
These parameters represent common traits from various site specific calibrations performed as part of the FertOrgaNic project from 2003-2005. It is not recommended for direct use, but as a base for a veriety specific calibration.
The calibration was perfomed by Charlotte Tofteng <cto@kvl.dk> in 2006.

22.11 Potato; Agria

A 'Potato; FertOrgaNic' model (see 22.10, page 145) defined in 'potato.dai'.

Recalibration of Agria potato performed by Charlotte Tofteng <cto@kvl.dk> in 2006 based on experimental results from Czechia (2003-2005), Slovakia (2003) and Portugal (2003).

22.12 Potato; Folva

A 'Potato; FertOrgaNic' model (see 22.10, page 145) defined in 'potato.dai'.

Recalibration of Agria potato performed by Tove Heidman in 2006 based on experimental results from DJF.

22.13 Potato

A 'Potato; Folva' model (see 22.12, page 146) defined in 'potato.dai'.

22.14 Potato; Triada

A 'Potato; FertOrgaNic' model (see 22.10, page 145) defined in 'potato.dai'.

Recalibration of Triada potato performed by Charlotte Tofteng <cto@kvl.dk> based on experimental results from Jadwisin, Poland 2003-2005.

22.15 Rye

A 'default' model (see 22.1, page 143) defined in 'rye.dai'.

22.16 Spring Barley

A 'default' model (see 22.1, page 143) defined in 'sbarley.dai'.

22.17 Spring Barley; Foulum

A 'Spring Barley' model (see 22.16, page 146) defined in 'sbarley.dai'.

RS-Model Projekt

22.18 Sugar Beet

A 'default' model (see 22.1, page 143) defined in 'sugarbeet.dai'.

Neuenkirchen Project

22.19 Fodder Beet

A 'default' model (see 22.1, page 143) defined in 'fodderbeet.dai'.

22.20 Spring Rape

A 'default' model (see 22.1, page 143) defined in 'srape.dai'.

22.21 Spring Wheat

A 'default' model (see 22.1, page 143) defined in 'swheat.dai'.
Kge Project

22.22 Winter Barley

A 'default' model (see 22.1, page 143) defined in 'wbarley.dai'.

22.23 Winter Barley; Foulum

A 'Winter Barley' model (see 22.22, page 147) defined in 'wbarley.dai'.

22.24 Winter Barley; Koge

A 'default' model (see 22.1, page 143) defined in 'wbarley.dai'.
Kge Project

22.25 Winter Rape

A 'default' model (see 22.1, page 143) defined in 'wrape.dai'.

22.26 Winter Wheat

A 'default' model (see 22.1, page 143) defined in 'wwheat.dai'.

22.27 Winter Wheat; Eest

A 'Winter Wheat' model (see 22.26, page 147) defined in 'wwheat.dai'.

22.28 Winter Wheat; Foulum

A 'Winter Wheat' model (see 22.26, page 147) defined in 'wwheat.dai'.
RS-Model Project

22.29 Grass to grain

A 'default' model (see 22.1, page 143) defined in 'grass.dai'.

22.30 Grass

A 'Grass to grain' model (see 22.29, page 147) defined in 'grass.dai'.

22.31 Ryegrass

A 'default' model (see 22.1, page 143) defined in 'ryegrass.dai'.
ryegrass v.27 Henning Hgh Jensen

22.32 Wclover

A 'default' model (see 22.1, page 143) defined in 'wclover.dai'.
v.30 Developed by Henning Hgh Jensen

22.33 Beetroot

A 'default' model (see 22.1, page 143) defined in 'beetroot.dai'.
For AArhus County - Samsoe Project

22.34 Broccoli

A 'default' model (see 22.1, page 143) defined in 'broccoli.dai'.
For AArhus County - Samsoe Project

22.35 Broccoli - transplanted

A 'Broccoli' model (see 22.34, page 148) defined in 'broccoli.dai'.

22.36 Brussels sprouts

A 'default' model (see 22.1, page 143) defined in 'brusselssprouts.dai'.
For AArhus County - Samsoe Project

22.37 Brussels sprouts - transplanted

A 'Brussels sprouts' model (see 22.36, page 148) defined in 'brusselssprouts.dai'.

22.38 Celeriac

A 'default' model (see 22.1, page 143) defined in 'celeriac.dai'.
For AArhus County - Samsoe Project

22.39 Celeriac - transplanted

A 'Celeriac' model (see 22.38, page 148) defined in 'celeriac.dai'.

22.40 Potato; SCRI - AArhus

A 'default' model (see 22.1, page 143) defined in 'earlypotato.dai'.
Potate paramterized by sha@kvl.dk – 2003. It is based on experimental data from SCRI from 1983 and 1984.

22.41 Early potato

A 'Potato; SCRI - AArhus' model (see 22.40, page 148) defined in 'earlypotato.dai'.
For AArhus County - Samsoe Project

22.42 White cabbage

A 'default' model (see 22.1, page 143) defined in 'whitecabbage.dai'.
For AArhus County - Samsoe Project

22.43 White cabbage - transplanted

A 'White cabbage' model (see 22.42, page 149) defined in 'whitecabbage.dai'.

22.44 Early white cabbage - transplanted

A 'White cabbage - transplanted' model (see 22.43, page 149) defined in 'early-whitecabbage.dai'.

22.45 Onion

A 'default' model (see 22.1, page 143) defined in 'onion.dai'.
For AArhus County - Samsoe Project

22.46 Onion - planting of sets

A 'Onion' model (see 22.45, page 149) defined in 'onion_plantingofsets.dai'.

22.47 Rug

A 'default' model (see 22.1, page 143) defined in 'dk-rye.dai'.
Rug med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.48 Vaarbyg

A 'default' model (see 22.1, page 143) defined in 'dk-sbarley.dai'.
Vrbyg med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.49 Vinterbyg

A 'default' model (see 22.1, page 143) defined in 'dk-wbarley.dai'.
Vinterbyg med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.50 Vinterhvede

A 'default' model (see 22.1, page 143) defined in 'dk-wwheat.dai'.
Vinterhvede med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.51 Froegraes

A 'default' model (see 22.1, page 143) defined in 'dk-grass.dai'.

Frgrs med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.52 Graes

A 'Froegraes' model (see 22.51, page 150) defined in 'dk-grass.dai'.

Grs med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.53 Silomajs

A 'default' model (see 22.1, page 143) defined in 'dk-maize.dai'.

Majs med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet. Baseret p Roskilde 74-76 ukendt varietet.

22.54 Aert

A 'default' model (see 22.1, page 143) defined in 'dk-pea.dai'.

rt med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

22.55 Vinterraps

A 'default' model (see 22.1, page 143) defined in 'dk-wrape.dai'.

Vinterraps med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet

22.56 simple

Forced growth crop model.

- *day*: number [**d**]
  State variable (default 0)
  Number of days since sowing (or spring).

- *T_sum*: number [**dg C d**]
  State variable (default 0)
  Temperature sum since sowing (or spring).

- *Root*: **RootSystem** fixed component (see section 85.13)
  Submodel (has fully specified default value)
  Root system.

- *Canopy*: **CanopySimple** fixed component (see section 85.15)
  Submodel (has fully specified default value)
  Canopy.

- *LAIvsTS*: plf [**dg C d** → $\mathbf{m^2/m^2}$]
  Optional parameter
  LAI as a function of T_sum
- *LAIvsDay*: plf [**d** → $\mathbf{m^2/m^2}$]
  Optional parameter
  LAI as a function of number of days since sowing.
- *forced_LAI*: number [$\mathbf{m^2/m^2}$]
  State variable (default 0)
  Minimum LAI, automatically cleared when exceeded by 'LAIvsTS'.
- *height_max*: number [**cm**]
  Parameter (default 80)
  Maximum height of plant, reached when flowering.
- *spring*: integer array of length 2
  Parameter (has default value with length 2)

  ```
  (spring 3 1)
  ```

  Parameter description:
  Zero 'T_sum' at this month and day.
- *spring_LAI*: number [$\mathbf{m^2/m^2}$]
  Parameter (default 0.1)
  Set 'forced_LAI' to this after spring clearence of 'T_sum'.
- *root_DM*: number [**Mg DM/ha**]
  Parameter (default 2)
  Fully developed root drymatter.
- *root_N*: number [**kg N/ha**]
  Parameter (default 20)
  Fully developed root N content.
- *root_am*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (root_am "AOM-SLOW"
           "AOM-FAST")
  ```

  Parameter description:
  Root AM parameters.
- *potential_N*: number [**kg N/ha**]
  Parameter
  Potential N content at harvest.
- *N_actual*: number [$\mathbf{g\ N/m^2}$]
  State variable (default 0)
  N uptake until now.
- *N_b*: number [**kg N/ha**]
  Parameter (default 10)
  N uptake form parameter.
- *N_flowering*: number [<**fraction**>]
  Parameter (default 0.9)
  Fraction of potential N uptake reached at flowering.

Log Variables

- *N_demand*: number [**g N/m^2**]
  Current potential N content.

Chapter 23

depth

Find the depth of two numbers.

23.1 const

Constant depth.

- *value*: number [**cm**]
  Parameter
  Constant depth.

23.2 file

Linear interpolation of depth read from file.

- *file*: string (see section 1.4.5)
  Parameter
  Name of file to read data from. The format of each line in the file is 'YEAR MONTH DAY HEIGHT', where HEIGHT should in cm above ground (i.e. a negative number). Linear interpolation is used between the datapoints.

23.3 extern

Look up depth in an scope.

- *value*: **number** component (see chapter 45)
  Expression that evaluates to a depth.

- *initial_value*: number [**cm**]
  Optional parameter
  Initial depth.

23.4 PLF

Linear interpolation of depth.

- *table*: submodel (see section 1.4.7) sequence
  Height as a function of time. This is a list where each element has the form (TIME VALUE). The TIME entries must be increasing cronologically. The

corresponding VALUE represents the value at that time. In order to find the depth for other times, linear interpolation between the entries in the list will be used.

< *time value* >

– *time*: **Time** fixed component (see section 85.21)
Submodel (has partially specified default value)
Time.

– *value*: number [**cm**]
Parameter
Depth.

Chapter 24

difrad

The 'difrad' component should calculate the diffuse radiation from meterological data.

< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

24.1 weather

Diffuse radiation using weather data.

24.2 DPF

Diffuse radiation calculated using the model of De Pury and Farquhar, 1997. See also [de Pury and Farquhar, 1997]

- *a*: number (dimensionless)
  Parameter (default 0.84)
  Atmospheric transmission coefficient of PAR. Value around 0.6-0.9 depending on dust particles.

- *fa*: number [<**fraction**>]
  Parameter (default 0.5)
  Diffuse radiation proportion. Proportion of attenuated radiation that reaches the surface as diffuse radiation.

Chapter 25

domsorp

Sorption and desorption of DOM to SOM.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

25.1 default

Transformation between two soil chemicals.

- *transform*: **transform** component (see chapter 72)
  Tranformation process between DOM and SOM.

- *dom_pool*: integer
  Parameter
  Number of the DOM pool affected by the transformation.

- *som_pool*: integer
  Parameter
  Number of the SOM pool affected by the transformation.

Log Variables

- *S_C*: number [**g C/cm^3/h**] soil cells
  Carbon converted from DOM to SOM (may be negative).

- *S_N*: number [**g N/cm^3/h**] soil cells
  Carbon converted from DOM to SOM (may be negative).

Chapter 26

drain

Lateral transport of water.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

26.1 none

No lateral transport.
Used by column default Drain (see 19.1, page 130) .

26.2 lateral

Pipe drainage. See also [Hooghoudt, 1940]

- x: number [**cm**]
  Optional parameter
  Horizontal distance to nearest pipe. By default, this is 1/2 L.

- L: number [**cm**]
  Parameter (default 1800)
  Distance between pipes.

- *rad*: number [**cm**]
  Parameter (default 3.5)
  Inner radius of drain pipes.

- *height*: number [**cm**]
  Optional state variable
  Current groundwater level (a negative number).

- *pipe_position*: number [**cm**]
  Parameter (default -110)
  Height pipes are placed in the soil (a negative number).

- *K_to_pipes*: number [**cm/h**]
  Optional parameter
  Horizontal conductivity in saturated soil. By default this is calculated from the horizontal conductivity and the anisotropy of the horizon.

- *eq_depth*: **draineqd** component (see chapter 27)
  Component (default 'MolenWesseling')
  Model for calculating equivalent depth for drains.

Log Variables

- *S*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Pipe drainage.

- *DrainFlow*: number [**cm/h**]
  Drain flow to pipes.

- *EqDrnFlow*: number [**cm/h**]
  Equilibrium drain flow to pipes.

Chapter 27

draineqd

Find the equilibrium drain depth for the Hooghoudt drainage model.

```
< component   (description description)
              (cite cite ...)            ; Has default value. >
```

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (has default value with length 1)

  ```
  (cite hooghoudt)
  ```

  Parameter description:
  BibTeX keys that would be relevant for this model or paramterization.

27.1 none

No modifications. See also [Hooghoudt, 1940]

27.2 Wesseling

Equivalent depth calculated with model by Wesseling. See also [Wesseling, 1973]

27.3 Moody

Equivalent depth calculated with model by Moody. See also [Moody, 1966]

27.4 MolenWesseling

Equivalent depth calculated with model by van der Molen and Wesseling. See also [der Molen and Wesseling, 1991]

Used by drain lateral eq_depth (see 26.2, page 159) .

Chapter 28

element

An element of a compound.

28.1 atom

An atom.

- *mass*: number [**g/mol**]
  Parameter
  Atomic mass.

Chapter 29

equilibrium

Find equilibrium between two soil chemicals.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

29.1 linear

A = K B

- *K*: **number** component (see chapter 45)
  The ratio A/B at equilibrium [].

29.2 Langmuir

A = (my_max B) / (K + B)

- *K*: **number** component (see chapter 45)
  Half saturation constant [g/cm^3].

- *my_max*: **number** component (see chapter 45)
  Max equilibrium capacity [g/cm^3].

29.3 goal_A

Attempt to maintain A at at fixed level.

- *goal_A*: **number** component (see chapter 45)
  The desired level of A [g/cm^3].

- *A_solute*: boolean (see section 1.4.2)
  Parameter
  True iff 'goal_A' is in solute (mass per volume water). If false, the unit is assumed to be mass per volume space.

- *min_B*: **number** component (see chapter 45)
  Do not convert B to A if B is smaller than this [g/cm^3].

- *B_solute*: boolean (see section 1.4.2)
  Parameter
  True iff 'min_B' is in solute (mass per volume water). If false, the unit is assumed to be mass per volume space.

- *debug_cell*: integer
  Parameter (default -1)
  Print debug information for this cell. Set it to a negative number to disable it.

Chapter 30

exchange

A named value to exchange with external models.

< `component` `(name` *`name`*`)`
`(description` *`description`*`)`
`(cite)` >

- *name*: string (see section 1.4.5)
  Parameter
  Name of value to exchange.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

30.1 number

Exchange a numeric value.

- *value*: number (dimension not specified)
  Optional state variable
  Current value to exchange.

- *dimension*: string (see section 1.4.5)
  Parameter
  Dimension of value to exchange.

30.2 name

Exchange a string value.

- *value*: string (see section 1.4.5)
  Parameter
  Current value to exchange.

Chapter 31

format

Text formatting component.

31.1 LaTeX

Format text as LaTeX.

Used by program document format (see 52.3, page 267) , and program docmodel format (see 52.4, page 268) .

Chapter 32

gnuplot

Plot a graph with gnuplot.

< `component` `(description` *`description`*`)` `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

32.1 common

Common parameters.

- *where*: string (see section 1.4.5)
  Optional parameter (default 'screen')
  File to store results in. By default, show them on a window. The format is determined from the file name extension: *.tex: LaTeX code with PostScript specials. *.eps: Encapsulated PostScript. *.pdf: Adobe PDF files.

  The special name 'screen' indicate that the data should be shown on the screen instead of being stored in a file.

- *title*: string (see section 1.4.5)
  Optional parameter
  Title of the plot, if any. Set it to an empty string to disable.

- *extra*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of extra gnuplot commands. The commands will be inserted right before the plot command. Note that if you have multiple plots in the same command file, The extra commands may affect the subsequence plots.

- *device*: string (see section 1.4.5)
  Optional parameter
  Output device. By default, this is derived from the file extenstion.

- *canvas*: string (see section 1.4.5)
  Optional parameter
  Canvas size. By default, this depend on the device.

- *size*: submodel (see section 1.4.7)
  Optional submodel
  Relative to size of plot. The standard size is 1.0, specify other numbers to scale accordingly.

  < *x y* >

  - *x*: number (dimensionless)
    Parameter
    Relative horizontal size of plot.
  - *y*: number (dimensionless)
    Parameter
    Relative vertical size of plot.

- *legend*: string (see section 1.4.5)
  Optional parameter (default 'auto')
  Placement of legend. This can be one of the four corners, named by compas locations (nw, ne, sw, se) to get the legend inside the graph in that corner, 'below' to get the legend below the graph, 'outside' to get the legend to the right of the graph, or 'none' to avoid getting a legend at all.

  The value 'auto' mean the legend will be places in the corner located farthest away from any data points. Note that datapoints outside the graph are ignored, and so are the lines connecting the datapoints. Thus, a line conncting two datapoints, one of them outside the graph, may cross the legend.

32.2 soil

A 'common' model (see 32.1, page 171) build into Daisy.
Generate a 2D gnuplot graph with soil content.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *max*: number [<**user**>]
  Optional parameter
  Fixed highest value. By default determine this from the data.

- *min*: number [<**user**>]
  Optional parameter
  Fixed lowest value. By default determine this from the data.

- *dimension*: string (see section 1.4.5)
  Optional parameter
  Dimension for data. By default, use dimension from file.

- *bottom*: number [**cm**]
  Optional parameter
  Deepest z value in plot. By default, derive value from data file.

- *type*: string (see section 1.4.5)
  State variable (default 'block')
  Plot type. Valid options are 'block' and 'contour'.

- *top*: number [**cm**]
  Parameter (default 0)
  Higest z value in plot.

- *left*: number [**cm**]
  Parameter (default 0)
  Minimum x value in plot.

- *right*: number [**cm**]
  Optional parameter
  Maximum x value in plot. By default, derive value from data file.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *when*: **Time** fixed component (see section 85.21)
  Submodel (has partially specified default value)
  Use value closest to this time.

- *samples*: integer
  Parameter (default 25)
  Number of sample lines for the 'smooth' and 'contour' types.

32.3 time

A 'common' model (see 32.1, page 171) build into Daisy.
Generate a gnuplot graph with times series.

- *source*: **source** component (see chapter 65) sequence
  Time series to plot.

- *begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  First date at x-axis.

- *end*: **Time** fixed component (see section 85.21)
  Optional submodel
  Last date at x-axis.

- *ymin*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left y-axis. By default determine this from the data.

- *ymax*: number [<**user**>]
  Optional parameter
  Fixed highest value on right y-axis. By default determine this from the data.

- *y2min*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left y-axis. By default determine this from the data.

- *y2max*: number [<**user**>]
  Optional parameter
  Fixed highest value on right y-axis. By default determine this from the data.

32.4 xy

A 'common' model (see 32.1, page 171) build into Daisy.
Generate a gnuplot graph with up to two x-axes.

- *source*: **xysource** component (see chapter 84) sequence
  XY series to plot.

- *ymin*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left y-axis. By default determine this from the data.

- *ymax*: number [<**user**>]
  Optional parameter
  Fixed highest value on right y-axis. By default determine this from the data.

- *y2min*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left y-axis. By default determine this from the data.

- *y2max*: number [<**user**>]
  Optional parameter
  Fixed highest value on right y-axis. By default determine this from the data.

- *xmin*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left x-axis. By default determine this from the data.

- *xmax*: number [<**user**>]
  Optional parameter
  Fixed highest value on right x-axis. By default determine this from the data.

- *x2min*: number [<**user**>]
  Optional parameter
  Fixed lowest value on left x-axis. By default determine this from the data.

- *x2max*: number [<**user**>]
  Optional parameter
  Fixed highest value on right x-axis. By default determine this from the data.

32.5 multi

Generate multiple graphs for the gnuplot command file.

- *before*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of extra gnuplot commands. The commands will be inserted right before the first graph.

- *after*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of extra gnuplot commands. The commands will be inserted right after the last graph.

- *graph*: **gnuplot** component (see chapter 32) sequence
  Graphs to plot.

Chapter 33

groundwater

The 'groundwater' component is responsible for specifying the groundwater table at each timestep.

33.1 common

All groundwater models can log height.

Log Variables

- *height*: number [**cm**]
  Groundwater level. Positive numbers indicate free drainage.

33.2 aquitard

A 'common' model (see 33.1, page 177) build into Daisy.
Aquitard groundwater, free drainage.

- *K_aquitard*: number [**cm/h**]
  Parameter (default 0.001)
  Conductivity of the aquitard.
- *Z_aquitard*: number [**cm**]
  Parameter (default 200)
  Thickness of the aquitard. The aquitard begins below the bottommost soil horizon.
- *h_aquifer*: number [**cm**]
  Optional state variable
  Pressure potential in the aquifer below the aquitard. By default. this is Z_aquitard. You can alternatively specify the pressure as a virtual groundwater level. See 'pressure_table'.
- *pressure_table*: **depth** component (see chapter 23)
  Optional component
  Height of groundwater the corresponds to the pressure in the aquifer.

  If you drilled a well down to the aquifer, this is number what the water level in the well would be as height above ground (a negative number). This is different from the actual groundwater table, because the aquitart block the water, and the pipes lead the water away. You can alternatively specify the pressure directly, with 'h_aquifer'.

33.3 static

A 'common' model (see 33.1, page 177) build into Daisy.

Static groundwater level. Provided for backward compatibility, use 'deep' or 'fixed' instead.

- *table*: number [**cm**]
  Parameter (default 1)
  Groundwater level. Positive numbers indicate free drainage.

33.4 lysimeter

A 'common' model (see 33.1, page 177) build into Daisy.

Lysimeter bottom.

33.5 deep

A 'common' model (see 33.1, page 177) build into Daisy.

Deep groundwater, free drainage.

Used by column Askov Groundwater (see 19.2, page 132) , and column Jyndevad Groundwater (see 19.3, page 132) .

33.6 fixed

A 'common' model (see 33.1, page 177) build into Daisy.

Fixed high groundwater level.

- *table*: number [**cm**]
  Parameter
  Groundwater level (negative number below surface).

33.7 file

A 'common' model (see 33.1, page 177) build into Daisy.

Read groundwater table from a file.

- *file*: string (see section 1.4.5)
  Parameter
  Name of file to read data from. The format of each line in the file is 'YEAR MONTH DAY HEIGHT', where HEIGHT should in cm above ground (i.e. a negative number). Linear interpolation is used between the datapoints.

- *offset*: number [**cm**]
  Parameter (default 0)
  Add this to depth from file.

33.8 source

A 'common' model (see 33.1, page 177) build into Daisy.

Read groundwater table from a source.

- *source*: **source** component (see chapter 65)
  Groundwater table time series.

- *offset*: number [**cm**]
  Parameter (default 0)
  Add this to depth from source.

33.9 extern

A 'common' model (see 33.1, page 177) build into Daisy.
Look up groundwater table in an scope.

- *table*: **number** component (see chapter 45)
  Expression that evaluates to groundwate table in.

- *initial_table*: number [**cm**]
  Optional parameter
  Groundwater level for initialization of soil water.

33.10 flux

A 'common' model (see 33.1, page 177) build into Daisy.
Flux groundwater, free drainage.

- *flux*: number [**cm/h**]
  Parameter
  Constant flux to groundwater.

33.11 pipe

Groundwater for pipe (tile) drained soil. If you specify this groundwater model, and does not specify the 'zplus' Soil discretization parameter, an extra aquitard soil horizon approximately a third of the size of 'Z_aquitart' will be added. This will allow the grounwater level to sink into the aquitart. The model cannot handle groundwater levels below the last cell, or above the soil surface.

- x: number [**cm**]
  Optional parameter
  Horizontal distance to nearest pipe. By default, this is 1/2 L.

- L: number [**cm**]
  Parameter (default 1800)
  Distance between pipes.

- *height*: number [**cm**]
  Optional state variable
  Current groundwater level (a negative number).

- *K_aquitard*: number [**cm/h**]
  Parameter (default 0.001)
  Conductivity of the aquitard.

- *Z_aquitard*: number [**cm**]
  Parameter (default 200)
  Thickness of the aquitard. The aquitard begins below the bottommost soil horizon.

- *h_aquifer*: number [**cm**]
  Optional state variable
  Pressure potential in the aquifer below the aquitard. By default. this is Z_aquitard. You can alternatively specify the pressure as a virtual groundwater level. See 'pressure_table'.

- *pressure_table*: **depth** component (see chapter 23)
  Optional component
  Height of groundwater the corresponds to the pressure in the aquifer.

  If you drilled a well down to the aquifer, this is number what the water level in the well would be as height above ground (a negative number). This is different from the actual groundwater table, because the aquitart block the water, and the pipes lead the water away. You can alternatively specify the pressure directly, with 'h_aquifer'.

- *pipe_position*: number [**cm**]
  Parameter (default -110)
  Height pipes are placed in the soil (a negative number).

- *K_to_pipes*: number [**cm/h**]
  Optional parameter
  Horizontal conductivity in saturated soil. By default this is calculated from the horizontal conductivity and the anisotropy of the horizon.

Log Variables

- *S*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Pipe drainage.

- *DrainFlow*: number [**cm/h**]
  Drain flow to pipes.

- *EqDrnFlow*: number [**cm/h**]
  Equilibrium drain flow to pipes.

- *DeepPercolation*: number [**cm/h**]
  Deep percolation to aquifer.

Chapter 34

heatrect

Heat transport in rectangular grid.

34.1 none

No heat transport.

34.2 linear

Linear temperature interpolation between top and bottom.

34.3 Mollerup

Finite volume solution to heat transfer by Mikkel Mollerup.
Used by movement rectangle heat (see 42.2, page 229) .

- *solver*: **solver** component (see chapter 64)
  Component (default 'cxsparse')
  Model used for solving matrix equation system.
- *debug*: integer
  Parameter (default 0)
  Enable additional debug message. A value of 0 means no message, higher numbers means more messages.

Chapter 35

horizon

A 'horizon' is a soil type with specific physical properties. It is the responsibility of the 'horizon' component to specify these properties.

Used by SoilLayer @ horizon (see 85.9, page 421) , and SoilZone @ horizon (see 85.10, page 421) .

```
< component   (hydraulic hydraulic)                ; Default hypres value.
              (tortuosity tortuosity)              ; Default M_Q value.
              (HorHeat HorHeat)                    ; Has default value.
              (description description)
              (cite)
              (C_per_N C_per_N)
              (anisotropy 1 [])
              (dry_bulk_density dry_bulk_density)
              (SOM_C_per_N 11 11 11)
              (SOM_fractions SOM_fractions ...)
              (turnover_factor 1 [])
              (Nitrification Nitrification)        ; Default soil value.
              (secondary_domain secondary_domain)  ; Default none value.
              (attributes) >
```

- *hydraulic*: **hydraulic** component (see chapter 36)
  Component (default 'hypres')
  The hydraulic propeties of the soil.
- *tortuosity*: **tortuosity** component (see chapter 71)
  Component (default 'M_Q')
  The soil tortuosity.
- *HorHeat*: **HorHeat** fixed component (see section 85.22)
  Submodel (has fully specified default value)
  Heat capacity and conductivity.
- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.
- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.
- *C_per_N*: number [**g C/g N**]
  Optional parameter

Total C/N ratio for this horizon. This is the combined initial C/N ratio for all organic matter pools in the horizon. The C/N ration of the AOM and SMB pools is assumed to be known, given that this number is used to find the common C/N ration for the SOM pools. The C/N ration for the SOM pools will then gradually move towards the values specified by 'SOM_C_per_N'. By default, the values given by 'SOM_C_per_N' will be used for initialization.

- *anisotropy*: number (dimensionless)
  Parameter (default 1)
  Horizontal saturated water conductivity relative to vertical saturated water conductivity. The higher this value, the faster the water will move towards drain pipes.

- *dry_bulk_density*: number [**g/cm**^{3}]
  Optional parameter
  The soils dry bulk density. By default, this is calculated from the soil constituents.

- *SOM_C_per_N*: number [**g C/g N**] sequence
  Parameter (has default value with length 3)

  ```
  (SOM_C_per_N 11 11 11 [g C/g N])
  ```

  Parameter description:
  C/N ratio for each SOM pool in this soil. If 'C_per_N' is specified, this is used as a goal only. If 'C_per_N' is unspecified, the SOM pools will be initialized with this value.

- *SOM_fractions*: number (dimensionless) sequence
  Optional parameter
  Fraction of humus in each SOM pool, from slowest to fastest. Negative numbers mean unspecified, let Daisy find appropriate values.

- *turnover_factor*: number (dimensionless)
  Parameter (default 1)
  Factor multiplied to the turnover rate for all organic matter pools in this horizon.

- *Nitrification*: **nitrification** component (see chapter 44)
  Component (default 'soil')
  The soil nitrification process.

- *secondary_domain*: **secondary** component (see chapter 60)
  Component (default 'none')
  Secondary matrix domain for solute movement.

- *attributes*: submodel (see section 1.4.7) sequence
  Optional submodel (default: an empty sequence)
  List of additional attributes for this horizon. Intended for use with pedotransfer functions.

  < *key value* >

  - *key*: string (see section 1.4.5)
    Parameter
    Name of attribute.
  - *value*: number [<**user**>]
    Parameter
    Value of attribute.

35.1 default

USDA/FAO texture classification.
The soil consitutents are automatically normalized.
OBSOLETE: Use the USDA or FAO model instead.

- *clay*: number (dimensionless)
  Parameter
  Relative fraction of clay in soil.

- *silt*: number (dimensionless)
  Parameter
  Relative fraction of silt in soil.

- *sand*: number (dimensionless)
  Optional parameter
  Relative fraction of sand in soil.

- *fine_sand*: number (dimensionless)
  Optional parameter
  Relative fraction of fine sand in soil. NOTE: Not a real texture class, use 'sand' instead.

- *coarse_sand*: number (dimensionless)
  Optional parameter
  Relative fraction of coarse sand in soil. NOTE: Not a real texture class, use 'sand' instead.

- *humus*: number (dimensionless)
  Parameter
  Relative fraction of humus in soil.

35.2 aquitard

A 'default' model (see 35.1, page 185) build into Daisy.
Tecture for implicit aquitard horizon.

35.3 Ap_JB1

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.4 Ap_JB2

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.5 Ap_JB3

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.6 Ap_JB4

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.7 Ap_JB5

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.8 Ap_JB6

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.9 Ap_JB7

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.10 B_JB1

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.11 B_JB2

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.12 B_JB3

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.13 B_JB4

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.14 B_JB5

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.15 B_JB6

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.16 B_JB7

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.17 C_JB1

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.18 C_JB2

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.19 C_JB3

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.20 C_JB4

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.21 C_JB5

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.22 C_JB6

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.23 C_JB7

A 'default' model (see 35.1, page 185) defined in 'dk-horizon.dai'.

35.24 Askov Ap

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.

Askov 10 & 30 cm soil. Data from O.H. Jacobsen (1989): Umttet hydraulisk ledningsevne i nogle danske jorde. Beretning nr. S 2030. Statens Planteavlsforsg.

35.25 Askov B

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.

Askov 50 & 70 cm soil. Data from O.H. Jacobsen (1989): Umttet hydraulisk ledningsevne i nogle danske jorde. Beretning nr. S 2030. Statens Planteavlsforsg.

35.26 Askov C

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.

Askov 90 cm soil. Data from O.H. Jacobsen (1989): Umttet hydraulisk ledningsevne i nogle danske jorde. Beretning nr. S 2030. Statens Planteavlsforsg.

35.27 Jyndevad Ap

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.

Jyndevad 15 cm soil. Data from O.H. Jacobsen (1989): Umttet hydraulisk ledningsevne i nogle danske jorde. Beretning nr. S 2030. Statens Planteavlsforsg.

35.28 Jyndevad C

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.

Jyndevad 50 cm soil. Data from O.H. Jacobsen (1989): Umttet hydraulisk ledningsevne i nogle danske jorde .Beretning nr. S 2030. Statens Planteavlsforsg.

35.29 Foulum AP

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.
Provided by DJF as part if the FertOrgaNic project.

35.30 Foulum B

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.
Provided by DJF as part if the FertOrgaNic project.

35.31 Foulum C

A 'default' model (see 35.1, page 185) defined in 'dk-soil.dai'.
Provided by DJF as part if the FertOrgaNic project.

35.32 USDA3

A horizon using USDA3 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 50 [um].
- *sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 50 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.33 FAO3

A 'USDA3' model (see 35.32, page 188) build into Daisy.
'FAO3' is anothor name for 'USDA3'

35.34 USDA7

A horizon using USDA7 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].

- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 50 [um].
- *very_fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 50 [um] and 100 [um].
- *fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 100 [um] and 250 [um].
- *medium_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 250 [um] and 500 [um].
- *coarse_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 500 [um] and 1000 [um].
- *very_corase_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 1000 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.35 FAO7

A 'USDA7' model (see 35.34, page 188) build into Daisy.
'FAO7' is anothor name for 'USDA7'

35.36 ISSS3

A horizon using ISSS3 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 20 [um].
- *sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 20 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.

- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.37 ISSS4

A horizon using ISSS4 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 20 [um].
- *fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 20 [um] and 200 [um].
- *coarse_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 200 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.38 USPRA3

A horizon using USPRA3 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 5 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 5 [um] and 50 [um].
- *sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 50 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.39 USPRA4

A horizon using USPRA4 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 5 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 5 [um] and 50 [um].
- *fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 50 [um] and 250 [um].
- *coarse_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 250 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.40 BSI3

A horizon using BSI3 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 60 [um].
- *sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 60 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.41 MIT3

A 'BSI3' model (see 35.40, page 191) build into Daisy.
'MIT3' is anothor name for 'BSI3'

35.42 DIN3

A 'BSI3' model (see 35.40, page 191) build into Daisy.
'DIN3' is anothor name for 'BSI3'

35.43 BSI7

A horizon using BSI7 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].

- *fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 60 [um] and 200 [um].

- *medium_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 200 [um] and 600 [um].

- *coarse_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 600 [um] and 2000 [um].

- *fine_silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 6 [um].

- *medium_silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 6 [um] and 20 [um].

- *coarse_silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 20 [um] and 60 [um].

- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.

- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.44 MIT7

A 'BSI7' model (see 35.43, page 192) build into Daisy.
'MIT7' is anothor name for 'BSI7'

35.45 DIN5

A horizon using DIN5 texture classification.

- *clay*: number [<**fraction**>]
  Parameter
  Mineral particles up to 2 [um].
- *silt*: number [<**fraction**>]
  Parameter
  Mineral particles between 2 [um] and 60 [um].
- *fine_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 60 [um] and 200 [um].
- *medium_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 200 [um] and 600 [um].
- *coarse_sand*: number [<**fraction**>]
  Parameter
  Mineral particles between 600 [um] and 2000 [um].
- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.

35.46 numeric

A horizon using explicit texture classification.

- *humus*: number [<**fraction**>]
  Parameter
  Humus content of soil.
- *fractions*: number [<**fraction**>] sequence
  Parameter
  Fraction of particles between the corresponding numrical limits.
- *normalize*: boolean (see section 1.4.2)
  Parameter (default false)
  If this is true, normalize the mineral fraction to 1.0. Otherwise, give an error if the sum is not 1.0.
- *limits*: number [**um**] sequence
  Parameter
  Numerical limits for particle sizes.

Chapter 36

hydraulic

This component is responsible for specifying the soils hydraulic properties.
Used by horizon component hydraulic (see 35, page 183) .
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

36.1 B_BaC_Bimodal

Brooks and Corey retention curve model with Burdine theory. Bimodal hydraulic conductivity curve.

- *lambda*: number (dimensionless)
  Parameter
  Pore size index.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.

- *Theta_b*: number (dimensionless)
  Parameter
  Water content at 'h_b'.

- *K_b*: number [**cm/h**]
  Parameter
  Water conductivity at 'h_b'.

36.2 M_BaC_Bimodal

Brooks and Corey retention curve model with Mualem theory. Bimodal hydraulic conductivity curve.

- *lambda*: number (dimensionless)
  Parameter
  Pore size index.
- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.
- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.
- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.
- *Theta_b*: number (dimensionless)
  Parameter
  Water content at 'h_b'.
- *K_b*: number [**cm/h**]
  Parameter
  Water conductivity at 'h_b'.

36.3 mod_C

Modified Campbell retention curve model with Burdine theory.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.
- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.
- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  `< h K >`

  - h: number [**cm**]
    Parameter
    Soil water pressure.
  - K: number [**cm/h**]
    Parameter
    Water conductivity.

- b: number (dimensionless)
  Parameter
  Campbell parameter.

36.4 B_BaC

Brooks and Corey retention curve model with Burdine theory.

Used by horizon Jyndevad Ap hydraulic (see 35.27, page 187) , and horizon Jyndevad C hydraulic (see 35.28, page 187) .

- *lambda*: number (dimensionless)
  Parameter
  Pore size index.
- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.
- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.
- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.
- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  `< h K >`

  - h: number [**cm**]
    Parameter
    Soil water pressure.
  - K: number [**cm/h**]
    Parameter
    Water conductivity.

36.5 M_BaC

Brooks and Corey retention curve model with Mualem theory.

- *lambda*: number (dimensionless)
  Parameter
  Pore size index.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.

- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < `h K` >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.

36.6 B_C

Campbell retention curve model with Burdine theory. See also [CAMPBELL, 1974, Burdine, 1953]

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.

- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < `h K` >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.

- *b*: number (dimensionless)
  Parameter
  Campbell parameter.

36.7 M_C

Campbell retention curve model with Mualem theory. See also [CAMPBELL, 1974, Mualem, 1976]

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.
- *h_b*: number [**cm**]
  Parameter
  Bubbling pressure.
- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.
  - < `h K` >
  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.
- *b*: number (dimensionless)
  Parameter
  Campbell parameter.

36.8 B_vG

van Genuchten retention curve model with Burdine theory.

- *alpha*: number [$\mathbf{cm}^{-1}$]
  Parameter
  van Genuchten alpha.
- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.
- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < *h K* >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.

- *n*: number (dimensionless)
  Parameter
  van Genuchten n.

- *l*: number (dimensionless)
  Parameter (default 2)
  tortuosity parameter.

36.9 M_vG

van Genuchten retention curve model with Mualem theory.

Used by horizon Ap_JB1 hydraulic (see 35.3, page 185) , horizon Ap_JB2 hydraulic (see 35.4, page 185) , horizon Ap_JB3 hydraulic (see 35.5, page 185) , horizon Ap_JB4 hydraulic (see 35.6, page 185) , horizon Ap_JB5 hydraulic (see 35.7, page 186) , horizon Ap_JB6 hydraulic (see 35.8, page 186) , horizon Ap_JB7 hydraulic (see 35.9, page 186) , horizon B_JB1 hydraulic (see 35.10, page 186) , horizon B_JB2 hydraulic (see 35.11, page 186) , horizon B_JB3 hydraulic (see 35.12, page 186) , horizon B_JB4 hydraulic (see 35.13, page 186) , horizon B_JB5 hydraulic (see 35.14, page 186) , horizon B_JB6 hydraulic (see 35.15, page 186) , horizon B_JB7 hydraulic (see 35.16, page 186) , horizon C_JB1 hydraulic (see 35.17, page 186) , horizon C_JB2 hydraulic (see 35.18, page 186) , horizon C_JB3 hydraulic (see 35.19, page 187) , horizon C_JB4 hydraulic (see 35.20, page 187) , horizon C_JB5 hydraulic (see 35.21, page 187) , horizon C_JB6 hydraulic (see 35.22, page 187) , horizon C_JB7 hydraulic (see 35.23, page 187) , horizon Askov Ap hydraulic (see 35.24, page 187) , horizon Askov B hydraulic (see 35.25, page 187) , horizon Askov C hydraulic (see 35.26, page 187) , horizon Foulum AP hydraulic (see 35.29, page 188) , horizon Foulum B hydraulic (see 35.30, page 188) , and horizon Foulum C hydraulic (see 35.31, page 188) .

- *alpha*: number [$\mathbf{cm}^{-1}$]
  Parameter
  van Genuchten alpha.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < h K >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.

- *n*: number (dimensionless)
  Parameter
  van Genuchten n.

- *l*: number (dimensionless)
  Parameter (default 0.5)
  tortuosity parameter.

36.10 yolo

Yolo soil. Haverkamp et.al., 1977.

- *M_intervals*: integer
  Parameter (default 500)
  Number of intervals for numeric integration of K.

36.11 M_vG_compact

van Genuchten retention curve model with Mualem theory and compaction.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *ref_alpha*: number [$\mathbf{cm}^{-1}$]
  Parameter
  Reference van Genuchten alpha.

- *ref_n*: number (dimensionless)
  Parameter
  Reference van Genuchten n.

- *ref_K_sat*: number [**cm/h**]
  Parameter
  Reference water conductivity of saturated soil.

- *mod_alpha*: plf [<**fraction**> → <**none**>]
  Parameter
  Porosity modifier for van Genuchten alpha.

- *mod_n*: plf [<**fraction**> → <**none**>]
  Parameter
  Porosity modifier for van Genuchten n.

- *mod_K_sat*: plf [<**fraction**> → <**none**>]
  Parameter
  Porosity modifier for water conductivity of saturated soil.

36.12 Cosby_et_al

Modified Campbell retention curve model with Burdine theory. Parameters estimated from soil texture as specified by Cosby et at.

36.13 hypres

van Genuchten retention curve model with Mualem theory. Parameters specified by the HYPRES transfer function. See also [Wösten et al., 1999]

Used by horizon component hydraulic (see 35, page 183) .

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < `h K` >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.

- *topsoil*: boolean (see section 1.4.2)
  Optional parameter
  If set true this horizon will be initialized as a topsoil (i.e. the plowing layer), if set false it will be initialized as a subsoil. By default, the horizon will be initialized as a topsoil if and only if it is the topmost horison in the soil profile.

36.14 M_vGp

van Genuchten retention curve model with Mualem theory. A p_m (h) function is multiplied to the conductivity to simulate the change near maropores.

p_m = $(1/(-h\ X + 1))^f$; h > h_m p_m = $(1/(-h_m\ X + 1))^f$; h <= h_m X = 1 cm^{-1} See also [Børgesen et al., 2006]

- *alpha*: number [$\mathbf{cm}^{-1}$]
  Parameter
  van Genuchten alpha.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *n*: number (dimensionless)
  Parameter
  van Genuchten n.

- *l*: number (dimensionless)
  Parameter (default 0.5)
  tortuosity parameter.

- *h_m*: number [**cm**]
  Parameter
  Pressure point of chance between matrix and macropores.

- *f*: number (dimensionless)
  Parameter
  Macropores conductivity curve shape parameter.

36.15 MACRO

van Genuchten retention curve model with Mualem theory. The near saturated retention and hydraulic properties have been adjusted to take macropores into account. See also [Larsbo and Jarvis, 2003]

- *alpha*: number [$\mathbf{cm}^{-1}$]
  Parameter
  van Genuchten alpha.

- *Theta_sat*: number [<**fraction**>]
  State variable
  Saturation point.

- *Theta_res*: number [<**fraction**>]
  Parameter (default 0)
  Soil residual water.

- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.

- *h_b*: number [**cm**]
  Parameter
  Pressure at boundary point of change between matrix and macropores domains.

- *Theta_b*: number [<**fraction**>]
  Parameter
  Water content at boundary point.

- *K_b*: number [**cm/h**]
  Optional parameter
  Water conductivity at boundary point.

- *n*: number (dimensionless)
  Parameter
  van Genuchten n.
- *l*: number (dimensionless)
  Parameter (default 0.5)
  tortuosity parameter.
- *n_ma*: number (dimensionless)
  Parameter
  Macropore size distribution factor.
- *enable_K_macro*: boolean (see section 1.4.2)
  Parameter (default true)
  Include contribution from macropores in conductivity curve.
- *enable_Theta_macro*: boolean (see section 1.4.2)
  Parameter (default true)
  Include contribution from macropores in retention curve.

36.16 B_C_inverse

Campbell retention curve model with Burdine theory.

This implementation is based on inverse modelling, you specify water at wilting point (pF 4.2) and field capacity (pF 2.0), from which the retention curve (the b and h_b parameters) is derived. Based on this, the condutivity curve is fully specified by a single point. See also [CAMPBELL, 1974, Burdine, 1953]

- *Theta_sat*: number [<**fraction**>]
  Optional state variable
  Saturation point. By default, this will be estimated from soil composition and dry bulk density.
- *K_sat*: number [**cm/h**]
  Optional parameter
  Water conductivity of saturated soil.
- *K_at_h*: submodel (see section 1.4.7)
  Optional submodel
  Water conductivity at specified pressure.

  < *h K* >

  - *h*: number [**cm**]
    Parameter
    Soil water pressure.
  - *K*: number [**cm/h**]
    Parameter
    Water conductivity.
- *Theta_fc*: number [<**fraction**>]
  Parameter
  Field capacity.
- *Theta_wp*: number [<**fraction**>]
  Optional parameter
  Wilting point. By default, this value will be estimated from texture. See also [Madsen and Platou, 1983]

Chapter 37

integer

Generic representation of integers.

37.1 const

Always give the specified value.

- *value*: integer
  Parameter
  Fixed value for this integer.

37.2 cond

Return the value of the first clause whose condition is true.

- *clauses*: **IntegerCondClause** fixed component (see section 85.31) sequence
  List of clauses to match for.

37.3 sqr

Take the square of its argument.

- *operand*: **integer** component (see chapter 37)
  Operand for this function.

37.4 max

Use the largest value of its operands.

- *operands*: **integer** component (see chapter 37) sequence
  The operands for this function.

37.5 min

Use the smallest value of its operands.

- *operands*: **integer** component (see chapter 37) sequence
  The operands for this function.

37.6 *

Use the product of its operands.

- *operands*: **integer** component (see chapter 37) sequence
  The operands for this function.

37.7 +

Use the sum of its operands.

- *operands*: **integer** component (see chapter 37) sequence
  The operands for this function.

37.8 -

Negate integer or subtract integers. With one operand, negates it. With more than one operand, subtracts all but the first from the first.

- *operands*: **integer** component (see chapter 37) sequence
  The operands for this function.

37.9 mod

Modulo the first operand by the rest.

- *operands*: **integer** component (see chapter 37) array of length 2
  The operands for this function.

37.10 div

Divide the first operand by the rest.

- *operands*: **integer** component (see chapter 37) array of length 2
  The operands for this function.

Chapter 38

litter

Litter, surface residuals, or mulch below canopy.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

38.1 none

The effect of surface residuals is ignored by the model.
Used by column default Litter (see 19.1, page 130) .

38.2 permanent

A permanent litter layer cover the ground, as for example in a forest.

- *albedo*: number (dimensionless)
  Optional parameter
  Reflection factor. By default, the surface albedo will be used.

- *interception_capacity*: number [**mm**]
  Parameter
  Storage capacity of litter.

- *vapor_flux_factor*: number [<**fraction**>]
  Parameter (default 1)
  Reduction factor for potential evaporation below litter.

38.3 residue

A dynamic litter layer based on applied fertilizer and crop residuals.

A 'mulch area index' is calculated from the surface organic matericals, and from that a mulch cover is calculated based on Beer's law similarily to how the crop cover is calculated from the leaf area index. See also [Scopel et al., 2004]

- *albedo*: number (dimensionless)
  Optional parameter
  Reflection factor. By default, the surface albedo will be used.

- *vapor_flux_factor*: number [<**fraction**>]
  Parameter (default 0)
  Reduction factor for potential evaporation below litter. Only area covered by residue is affected.

- *water_capacity*: number [**L/kg**]
  Parameter
  Water holding capacity of surface residulas.

- *specific_AI*: number [$\mathbf{m^2}$**/kg DM**]
  Parameter
  Area covered per litter mass.

- *extinction_coefficent*: number (dimensionless)
  Parameter
  Beer's law extinction coefficient for litter.

38.4 Millet

A 'residue' model (see 38.3, page 207) build into Daisy.
Millet crop residues in Planaltina. See also [Macena et al., 2003]

38.5 Maize

A 'residue' model (see 38.3, page 207) build into Daisy.
Maize crop residues in La Tinaja. See also [Scopel et al., 1998]

Chapter 39

log

Running a simulation is uninteresting, unless you can get access to the results in one way or another. The purpose of the 'log' component is to provide this access. Most 'log' models does this by writing a summary of the state to a log file.

39.1 checkpoint

Create a checkpoint of the entire simulation state, suitable for later hot start.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *where*: string (see section 1.4.5)
  Parameter (default 'checkpoint')
  File name prefix for the generated checkpoint. The time will be appended, together with the '.dai' suffix.

- *when*: **condition** component (see chapter 21)
  Component (default 'finished')
  Make a checkpoint every time this condition is true.

39.2 harvest

Create a log of all harvests.

- *where*: string (see section 1.4.5)
  Parameter (default 'harvest.dlf')
  Name of the log file to create.

- *print_header*: string (see section 1.4.5)
  Parameter (default 'true')
  If this is set to 'false', no header is printed. If this is set to 'true', a full header is printer. If this is set to 'fixed', a small fixed size header is printed.

- *print_tags*: boolean (see section 1.4.2)
  Parameter (default true)
  Print a tag line in the file.

- *print_dimension*: boolean (see section 1.4.2)
  Parameter (default true)
  Print a line with units after the tag line.

- *print_N*: boolean (see section 1.4.2)
  Parameter (default true)
  Print nitrogen content of harvest.

- *print_C*: boolean (see section 1.4.2)
  Parameter (default false)
  Print carbon content of harvest.

39.3 select

Select variables to log.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to log.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *from*: number [**cm**]
  Optional parameter
  Default 'from' value for all entries. By default, use the top of the soil. OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Optional parameter
  Default 'to' value for all entries. By default, use the bottom of the soil. OBSOLETE: Use (volume box (bottom TO)) instead.

- *when*: **condition** component (see chapter 21)
  Add entries to the log file when this condition is true.

- *entries*: **select** component (see chapter 62) sequence
  What to log in each column.

- *time_columns*: boolean (see section 1.4.2)
  Optional parameter
  Iff true, add columns for year, month, mday and hour in the begining of the lines. By default, this will be true of you have not specified any time entries yourself.

39.4 DLF

A 'select' base model (see 39.3, page 210) build into Daisy.

Shared base class for log models generating Daisy Log File formatted results.

- *where*: string (see section 1.4.5)
  Parameter
  Name of the log file to create.

- *print_header*: string (see section 1.4.5)
  Parameter (default 'true')
  If this is set to 'false', no header is printed. If this is set to 'true', a full header is printer. If this is set to 'fixed', a small fixed size header is printed.

- *print_tags*: boolean (see section 1.4.2)
  Parameter (default true)
  Print a tag line in the file.

- *print_dimension*: boolean (see section 1.4.2)
  Parameter (default true)
  Print a line with units after the tag line.

- *parameter_names*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of string parameters to print to the table header.

  For example, if you have defined 'column' and 'crop' parameters for this table log parameterization, you can print them to the log file header by specifying '(names column crop)'.

- *print_initial*: boolean (see section 1.4.2)
  Parameter (default true)
  Print a line with initial values when logging starts.

- *flush*: boolean (see section 1.4.2)
  Parameter (default false)
  Flush to disk after each entry (for debugging).

- *record_separator*: string (see section 1.4.5)
  Parameter (default ' ')
  String to print between records (time steps).

- *field_separator*: string (see section 1.4.5)
  Parameter (default ' ')
  String to print between fields.

- *error_string*: string (see section 1.4.5)
  Parameter (default '!')
  String to print when errors are encountered.

- *missing_value*: string (see section 1.4.5)
  Parameter (default '00.00')
  String to print when the path doesn't match anything. This can be relevant for example if you are logging a crop, and there are no crops on the field.

- *array_separator*: string (see section 1.4.5)
  Parameter (default ' ')
  String to print between array entries.

39.5 table

A 'DLF' model (see 39.4, page 211) build into Daisy.
Each selected variable is represented by a column in the specified log file.

- *summary*: **summary** component (see chapter 68) sequence
  Component (default: an empty sequence)
  Summaries for this log file.

39.6 column

A 'table' model (see 39.5, page 212) defined in 'log-std.dai'.
A log table for a specific column.

- *column*: string (see section 1.4.5)
  Parameter (default '*')
  Name of column to log. Use "*" to log all columns.

- *colfid*: **string** component (see chapter 67)
  Component (default 'cond')

```
(colfid cond (("string-equal" "${column}" "*")
              "")
             ((true)
              "${column}_"))
```

  Parameter description:
  File component name indicating column logged.

39.7 chemical

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.
A log table for a specific chemical.

- *chemical*: string (see section 1.4.5)
  Parameter (default '*')
  Name of chemical to log.

- *chemfid*: **string** component (see chapter 67)
  Component (default 'cond')

```
(chemfid cond (("string-equal" "${chemical}" "*")
               chemicals)
              ((true)
               "${chemical}"))
```

  Parameter description:
  File name component indicating chemical logged.

39.8 Field chemical

A 'chemical' model (see 39.7, page 212) defined in 'log-std.dai'.
Content, transport and transformation of chemicals in the field.

- *unit*: string (see section 1.4.5)
  Parameter (default 'g/ha')
  Base unit.

Table columns:

- *Spray*: [**g/ha**]
  Applied to the surface.

  Chemicals can be added to the surface through the spray, fertilize and irrigate management operations.

- *Deposit*: [**g/ha**]
  Athmospheric deposition.

  This is usually zero when tracking pesticides, but can be relevant for atmospheric polutants, and is also signficant when for nitrogen.

- *Harvest*: [**g/ha**]
  Removed by harvest.

  Some amount of the chemical may be left on the canopy, and removed during harvest. It is usually not sinificant for the overall balance, but might be important to the consumers, depending on the chemical.

- *Dissipate*: [**g/ha**]
  Dissipating from the canopy.

  Some chemicals evaporate or decompose when stored on the canopy.

- *Litter Decompose*: [**g/ha**]
  Decomposed while stored in the litter pack.

- *Surface Decompose*: [**g/ha**]
  Decomposed while stored on the soil surface.

- *Surface Transform*: [**g/ha**]
  Added from chemical transformation on the soil surface.

- *Runoff*: [**g/ha**]
  Lost from soil surface due to lateral water flow.

- *Leak-Matrix*: [**g/ha**]
  Lost due to leaching.

- *Soil-Drain*: [**g/ha**]
  Lost from the soil matrix to drain pipes.

- *Surface-Drain*: [**g/ha**]
  Amount moving directly from surface to drain, through biopores.

  This never come in contact with the soil matrix.

- *External*: [**g/ha**]
  Added to the soil through some external mechanism, such as subsoil irrigation.

- *Uptake*: [**g/ha**]
  Removed from the soil matrix through plant roots.

- *Soil Decompose*: [**g/ha**]
  Decomposed while in the soil matrix.

- *Soil Transform*: [**g/ha**]
  Added through chemical transformation in the soil matrix.
- *Snow*: [**g/ha**]
  Amount stored in the snow pack.
- *Canopy*: [**g/ha**]
  Amount stored in on the canopy.
- *Litter*: [**g/ha**]
  Amount stored in the surface litter (mulch, residuals, fertilizer).
- *Surface*: [**g/ha**]
  Amount stored on the soil surface.
- *Soil*: [**g/ha**]
  Amount stored in the soil matrix.
- *Tertiary*: [**g/ha**]
  Amount stored in the biopores.
- *Error*: [**g/ha**]
  Amount borrowed from the future.

  This can be temporaily non-zero if some process remove the chemical fast compared to the timestep of the simulation.

39.9 Soil chemical

A 'chemical' model (see 39.7, page 212) defined in 'log-std.dai'.
Content, transport and transformation of chemicals in the soil.

- *unit*: string (see section 1.4.5)
  Parameter (default 'g/ha')
  Base unit.

Table columns:

- *In*: [**g/ha**]
  Infiltration.
- *Leak-Matrix*: [**g/ha**]
  Leaching.
- *Tertiary*: [**g/ha**]
  Net-loss to biopores.

  This is positive if the flux from the soil matrix to the biopores is larger than the flux from the biopores to the soil matrix.
- *Tillage*: [**g/ha**]
  Added through tillage operations.
- *Drain*: [**g/ha**]
  Lost to drains.
- *External*: [**g/ha**]
  Added externally, for example through subsoil irrigation.
- *Uptake*: [**g/ha**]
  Lost with water uptake by plant roots.

- *Decompose*: [**g/ha**]
  Decomposed.

- *Transform*: [**g/ha**]
  Added by chemical teransformation.

- *Content*: [**g/ha**]
  Total amount in the specified soil interval.

- *Error*: [**g/ha**]
  Borrowed from the future.

  This can be be non-zero if some process remove the chemical too fast compared to the simulation timestep.

39.10 crop

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.
A log table for a specific crop.

- *crop*: string (see section 1.4.5)
  Parameter (default '*')
  Name of crop to log. Use "*" to log all crops.

- *cropfid*: **string** component (see chapter 67)
  Component (default 'cond')

```
(cropfid cond (("string-equal" "${crop}" "*")
               crops)
              ((true)
               "${crop}"))
```

  Parameter description:
  File component name indicating crop logged.

39.11 biopore

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.
A log table for a specific biopore class.

- *biopore*: string (see section 1.4.5)
  Parameter (default '*')
  Name of biopore to log. Use "*" to log all biopore classes.

- *bioporefid*: **string** component (see chapter 67)
  Component (default 'cond')

```
(bioporefid cond (("string-equal" "${biopore}" "*")
                  biopores)
                 ((true)
                  "${biopore}"))
```

  Parameter description:
  File component name indicating biopore logged.

39.12 Soil water

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.

Information about water input, output and content for the soil.

Table columns:

- *Matrix infiltration*: [**mm**]
  Amount of water entering the soil through the matrix system.

  This is the normal infiltration channel.

- *Matrix percolation*: [**mm**]
  Amount of water leaving the soil through the matrix system.

  This is the normal percolation channel.

- *Subsoil irrigation*: [**mm**]
  mount of water incorporated directly into the soil.

  his include both subsoil irrigation and any water in fertilizer irectly incorporated into the soil.

- *Tertiary*: [**mm**]
  Net amount of water leaving the soil matrix to the biopores.

  This may be zero if the same amount of water enters and leaves the interval.

- *Drain flow*: [**mm**]
  Water in the soil interval that flows to the drain pipes.

  Note that water flow to the drain pipes from all soil layers, both above and below the pipes, and even from the automatically inserted aquitard horizon that lies below the normal soil horizons. So if you have specified less than the full soil profile, that is, if you have set the 'from' or 'to' parameters, the amount you see logged here will be less than the total amount of water in the drain pipes.

  If the soil is not drained, that is, if the specified groundwater model is not 'pipe', this amount will be zero.

- *Root extraction*: [**mm**]
  Water in the soil interval extracted by the roots.

  This number will be equal to the transpiration if the soil interval being logged includes the entire root zone.

- *Freezing*: [**mm**]
  Water in the soil interval turned into ice.

  This number will be zero unless the 'enable_ice' SoilHeat parameter has been set.

- *Tillage*: [**mm**]
  Net amount of water being added to the soil interval by tillage operations.

  This number will be zero if the soil interval includes the all the soil affected by the tillage operation, which is the normal case. The number will be negative if more water have been added than removed by the tillage operation.

- *Soil water*: [**mm**]
  The total amount of water in the soil interval

39.13 Field water

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.

Information about water input, output and content for the system.

The intended use of this log is large scale water balance, for example reservoir management. It provide information about how much water is in the field (down to a specified depth), where it is located in the field (surface, soil matrix or in biopores), as well as the sources, sinks and amounts of water entering or leaving the system. It does not provide information about internal translocation of water between surface, soil matrix and biopores, use see the 'Soil water' log instead for that.

For the balances of this log to work, you must include the entire root zone, as well as the biopore zone.

Table columns:

- *Precipitation*: [**mm**]
  Total amount of water entering the system as rain and snow.
- *Irrigation*: [**mm**]
  Total amount of water added to the system from irrigation.
- *Potential evapotranspiration*: [**mm**]
  There is energy enough to evaporate this amount of water.
- *Actual evapotranspiration*: [**mm**]
  Amount of water removed by evaporation and transpiration.

  The evaporation part covers water evaporated from the soil surface, the snow pack, or intertercepted water on the canopy. The transpiration covers water removed from the soil through the root system and the stomata on the leafs.

  The actual evapotranspiration will be lower than the potential evapotranspiration when the available water is insufficient.
- *Matrix percolation*: [**mm**]
  This is the amount of water leaving the system through the soil bottom. It can be negative if there are cabilary raise.
- *Soil drain flow*: [**mm**]
  Amount of water leaving the system to drain pipes.

  Note that the total amount of water in the drain pipes may be higher than this if you have specified the 'to' parameter, as the whole soil profile will contribute to the drain flow, and only the contributions from the soil interval being logged is counted here.
- *Surface drain flow*: [**mm**]
  Flux of water from surface to drain through biopores, bypassing the matrix.
- *Runoff*: [**mm**]
  Amount of water running of the surface.

  This is intended to simulate water runoff from a surface with a slop. However, since Daisy is a one dimensional model, the water have nowhere go to. For typical setups, this value will be zero.
- *Tertiary water*: [**mm**]
  Total amount of water in the biopores.

  Note the the biopore system does not keep track of where the water is located within the soil, so you will always get the total amount, even if you otherwise only log part of the soil.

- *Soil water*: [**mm**]
  Total water content of the soil.

- *Surface water*: [**mm**]
  Total water stored on the surface. This includes the snow pack, intercepted water, ponding, and water in the litter layer.

39.14 Field nitrogen

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.

Nitrogen input, output, transformation and content for the system.

The intended use of this log is large scale nitrogen balance, for example reservoir management. It provide information about how much nitrogen is in the field (down to a specified depth), where it is located in the field (surface, soil matrix or in biopores), what form is has (crop, soil organic matter, or mineral) as well as the sources, sinks and amounts of nitrogen entering or leaving the system, and transformation between the four forms. It does not provide information about internal translocation of nitrogen between surface, soil matrix and biopores, use see the 'Soil nitrogen' log instead for that.

For the balances of this log to work, you must include the entire root zone, as well as the biopore zone.

- *unit*: string (see section 1.4.5)
  Parameter (default 'kg N/ha')
  Base unit.

Table columns:

- *Min-Surface-Fertilizer*: [**kg N/ha**]
  Mineral fertilizer added above the soil surface.

- *Min-Soil-Fertilizer*: [**kg N/ha**]
  Mineral fertilizer incorporated into the soil.

- *Deposition*: [**kg N/ha**]
  Nitrogen added to the soil surface as atmospheric deposition.

- *Leaching*: [**kg N/ha**]
  Loss from leaching below the specified depth. This can be negative in case of cabilary rise.

- *Soil-Drain*: [**kg N/ha**]
  Loss of nitrogen from the soil matrix to drain pipes.

  This include both loss directly from the soil matrix to the drain, and loss from the soil matrix to biopores that are connected to the drainage system.

- *Surface-Drain*: [**kg N/ha**]
  Loss of nitrogen from above soil surface to drain pipes through biopores. This loss bypasses the soil matrix entirely.

- *Surface-Loss*: [**kg N/ha**]
  Loss of mineral nitrogen stored above surface.

  This includes various sources that are usually rare or small, such as surface runoff, mineral nitrogen on leaves removed at harvest, and chemical transformations. It does not include volatilization or infiltration.

- *Min-Surface*: [**kg N/ha**]
  Mineral nitrogen stored above the soil surface. This includes leaves, ponded water, etc.

- *Min-Soil*: [**kg N/ha**]
  Mineal nitrogen stored in the soil matrix.

- *Biopores*: [**kg N/ha**]
  Mineral nitrogen stored in biopores.

- *Error*: [**kg N/ha**]
  Mineral nitrogen 'borrowed from the future'.

  This can be non-zero if the processes that remove nitrogen from some place in the soil are fast compared to the timestep. If this happens, you should rerun the simulation withh a shorter timestep. Or readjust the system to slow down those processes that remove mineral nitrogen.

- *Mineralization*: [**kg N/ha**]
  NH4 produced in the soil matrix by mineralization of organic matter.

- *Immobilization*: [**kg N/ha**]
  NO3 immobilized as a result of organic matter turnover.

- *Crop-Uptake*: [**kg N/ha**]
  Nitrogen uptake by the roots.

- *Volatilization*: [**kg N/ha**]
  NH4-fertilizer lost during the application.

- *N2O-Nitrification*: [**kg N/ha**]
  N2O production in the soil matrix due to nitrification.

  The N2O produced is assumed to escape to the atmosphere, and will not be traced by the model.

- *Denitrification*: [**kg N/ha**]
  Loss of NO3 due to denitrification.

- *Fixated*: [**kg N/ha**]
  Atmospheric nitrogen fixated by the crop.

- *Org-Fertilizer*: [**kg N/ha**]
  Organically bound nitrogen supplied by fertilizers.

- *Seed*: [**kg N/ha**]
  Nitrogen supplied in seeds when sowing.

- *Harvest*: [**kg N/ha**]
  Nitrogen removed as part of the harvest.

- *Residuals-Surface*: [**kg N/ha**]
  Above ground plant residuals.

  This includes both exfoiliation during crop growth, and any above ground residuals left after harvest.

- *Residuals-Soil*: [**kg N/ha**]
  Below ground plant residuals.

  This includes root death and rhizodeposition during plant growth, as well as the roots left after harvest.

- *Org-Surface*: [**kg N/ha**]
  Nitrogen content of organic material on the soil surface.

  This does not include nitrogen in the crop.

- *Org-Soil*: [**kg N/ha**]
  Nitrogen content of organic material below the soil surface.

  This does not include nitrogen in living roots.

- *Crop*: [**kg N/ha**]
  Nitrogen content of the crop, excluding dead leaves.

- *Dead leaves*: [**kg N/ha**]
  Nitrogen content of dead leaves that are still sitting on the crop.

39.15 Soil nitrogen

A 'column' model (see 39.6, page 212) defined in 'log-std.dai'.
Nitrogen input, output, transformation and content for the soil.

- *unit*: string (see section 1.4.5)
  Parameter (default 'kg N/ha')
  Base unit.

Table columns:

- *NO3-In*: [**kg N/ha**]
  NO3-N infiltration.

- *NO3-Leak-Matrix*: [**kg N/ha**]
  NO3-N leaching.

- *NO3-Tertiary*: [**kg N/ha**]
  Net movement of NO3-N from soil matrix to biopores.

  This will be negative if more NO3 moves out of the biopores than into the biopores in the specified soil interval. A common case where this happens is when NO3 enter the biopores from the soil surface.

- *NO3-Drain*: [**kg N/ha**]
  NO3-N lost to drains.

- *NO3-Incorp*: [**kg N/ha**]
  NO3-N incorporated directly into the soil.

- *NO3-Tillage*: [**kg N/ha**]
  NO3-N added by tillage operations.

- *NO3-Uptake*: [**kg N/ha**]
  NO3-N removed by plant roots.

- *NO3-Content*: [**kg N/ha**]
  NO3-N content of the soil matrix (excluding biopores).

- *NO3-Error*: [**kg N/ha**]
  NO3-N borrowed from the future.

  This may be temporarily non-zero if some process remove NO3 too fast compared to the time step.

- *NH4-In*: [**kg N/ha**]
  NH4-N infiltration.

- *NH4-Leak-Matrix*: [**kg N/ha**]
  NH4-N leaching.

- *NH4-Tertiary*: [**kg N/ha**]
  Net movement of NH4-N from soil matrix to biopores.

  This will be negative if more NH4 moves out of the biopores than into the biopores in the specified soil interval. A common case where this happens is when NH4 enter the biopores from the soil surface.

- *NH4-Drain*: [**kg N/ha**]
  NH4-N lost to drain pipes.

- *NH4-Tillage*: [**kg N/ha**]
  NH4-N added by tillage operations.

- *NH4-Incorp*: [**kg N/ha**]
  NH4-N fertilizer incorporated directly into the soil.

- *NH4-Uptake*: [**kg N/ha**]
  NH4-N removed by plant roots.

- *NH4-Content*: [**kg N/ha**]
  NO3-N content of the soil matrix (excluding biopores).

- *NH4-Error*: [**kg N/ha**]
  NH4-N borrowed from the future.

  This may be temporarily non-zero if some process remove NH4 too fast compared to the time step..

- *Denitrification*: [**kg N/ha**]
  NO3-N removed by denitrification.

- *NH4-Nitrification*: [**kg N/ha**]
  NH4-N removed by nitrification.

- *NO3-Nitrification*: [**kg N/ha**]
  NO3-N added by nitrification.

- *N2O-Nitrification*: [**kg N/ha**]
  NH4-N lost as N2O as a byproduct of the nitrification process.

  NH4-Nitrification = NO3-Nitrification + N2O-Nitrification

- *NH4-Mineralization*: [**kg N/ha**]
  NH4-N added from mineralization of organic matter in the soil.

- *NO3-Immobilization*: [**kg N/ha**]
  NO3-N removed by the organic matter turnover processes.

- *Residuals-N*: [**kg N/ha**]
  Nitrogen in organic matter added to the soil from roots.

- *Tillage-Org_N*: [**kg N/ha**]
  Nitrogen in organic matter added to the soil by tillage operations.

- *Bioincorporation*: [**kg N/ha**]
  Nitrogen in organic matter added to the soil by bioincroporation.

  This includes dead leaves removed from the soil surface by earthworms.

- *AOM*: [**kg N/ha**]
  Total amount of nitrogen in added organic matter in the soil.

  AOM consists of fertilizer and plant residuals that has not yet been decomposed.

- *SOM*: [**kg N/ha**]
  Total amount of nitrogen in soil humus.

  This consist of dead organic matter that can no longer be traced to its origins.

- *SMB*: [**kg N/ha**]
  Nitrogen in soil microbiological organisms.

  The living part of the soil (excluding plant roots).

- *Buffer*: [**kg N/ha**]
  Nitrogen in humus added by fertilizer but not yet available for turnover.

  This is usually zero.

39.16 regress

A 'DLF' model (see 39.4, page 211) build into Daisy.

Maintain a DLF file containing simulation results from multiple runs. Warn if result from this run diverge from previous runs. The intention is that model should be used for regression testing.

39.17 extern

A 'select' model (see 39.3, page 210) build into Daisy.

Log simulation state for extern use.

- *numbers*: submodel (see section 1.4.7) sequence
  Optional submodel
  Inititial numeric values. By default, none.

  < `(value` *value*`)`
  `(name` *name*`)` >

  - *value*: number (dimension not specified)
    State variable
    Numeric value.
  - *name*: string (see section 1.4.5)
    State variable
    Name to refer to number with.

- *where*: string (see section 1.4.5)
  Optional parameter
  Name of the extern log to use. By default, use the model name.

- *parameter_names*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of parameters to export.

For example, if you have defined 'column' and 'crop' parameters for this extern log parameterization, you can export them to through the API interface by specifying '(names column crop)'.

Chapter 40

macro

Preferention flow in soil macro pores.

40.1 none

No macropores.

40.2 default

The area between 'height_start' and 'height_end' contains macropores, which are initiated when the water potential reach 'pressure_initiate', and then immediately emptied down to 'pressure_end'. The water entering the macropore is distributed in soil below as a source term, according to the 'distribution' parameter.

- *distribution*: plf [**cm** → <**fraction**>]
  Parameter
  Distribution of macropore end points as a function of height. The function should start with '1' at 'height_end', and then decrease to '0' at 'height_start'. It can be constant, but may never increase. The value indicates the fraction of macropores which ends at the given where all macropores is assumed to start at the top.

- *height_start*: number [**cm**]
  Optional parameter
  Macropores starts at this depth (a negative number). If not specified, use the last point in 'distribution'.

- *height_end*: number [**cm**]
  Optional parameter
  Macropores ends at this depth (a negative number). If not specified, use the first point in 'distribution'.

- *pressure_initiate*: number [**cm**]
  Parameter (default -3)
  Pressure needed to init pref.flow

- *pressure_end*: number [**cm**]
  Parameter (default -30)
  Pressure after pref.flow has been init

- *pond_max*: number [**mm**]
  Parameter (default 0.5)
  Maximum height of ponding before spilling into macropores. After macropores are activated pond will have this height.

Log Variables

- *S_p*: number [$\mathbf{h}^{-1}$]
  Macropore sink term.

Chapter 41

mactrans

Macropore transportation of solutes.

41.1 default

Solute follows water.

Used by tertiary old mactrans (see 70.2, page 341) .

Chapter 42

movement

This component handles the movement in the soil.
< `component (Tertiary` *`Tertiary`*`)` >

- *Tertiary*: **tertiary** component (see chapter 70)
  Tertiary (that is, non-matrix) transport method.

Log Variables

- *water_failure_level*: integer
  The number of the last water transport model to fail. It is -1 if the first model succeded, and 0 if the first model failed but the second succeded.

- *solute_failure_level*: integer
  The number of the last solute transport model to fail. It is -1 if the first model succeded, and 0 if the first model failed but the second succeded.

42.1 solute

Shared paramaters for handling solutes.

- *matrix_solute*: **transport** component (see chapter 73) sequence
  Matrix solute transport models. Each model will be tried in turn, until one succeeds. If none succeeds, the simulation ends.

- *matrix_solid*: **transport** component (see chapter 73)
  Component (default 'none')
  Matrix solute transport model used for fully sorbed constituents.

42.2 rectangle

A 'solute' model (see 42.1, page 229) build into Daisy.
Two dimensional movement in a rectangular grid.

- *Geometry*: **GeometryRect** fixed component (see section 85.26)
  Submodel (has partially specified default value)
  Discretization of the soil.

- *matrix_water*: **uzrect** component (see chapter 78) sequence
  Component (has default value with length 3)

```
(matrix_water Mollerup
              "v+h"
              const)
```

Parameter description:
Matrix water transport models. Each model will be tried in turn, until one succeeds. If none succeeds, the simulation ends.

- *drainpoints*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Location of cells with drain pipes.

  $<$ ***z x*** $>$

  - z: number [**cm**]
    Parameter
    Vertical position.
  - x: number [**cm**]
    Parameter
    Horizontal position.

- *heat*: **heatrect** component (see chapter 34)
  Component (default 'Mollerup')
  Heat transport model.

42.3 vertical

A 'solute' model (see 42.1, page 229) build into Daisy.
One dimensional movement.
Used by column default Movement (see 19.1, page 130) .

- *Geometry*: **Geometry1D** fixed component (see section 85.24)
  Submodel (has fully specified default value)
  Discretization of the soil.

- *matrix_water*: **uzmodel** component (see chapter 77) sequence
  Component (has default value with length 2)

  ```
  (matrix_water richards
                lr)
  ```

  Parameter description:
  Vertical matrix water transport models. Each model will be tried in turn, until one succeeds. If none succeeds, the simulation ends.

Chapter 43

net_radiation

The purpose of this component is to calculate the net radiation from other meteorological data.

< `component` >

Log Variables

- *net_radiation*: number [$\mathbf{W/m^2}$]
  The calculated net radiation (positive downwards).

- *L_n*: number [$\mathbf{W/m^2}$]
  The calculated net longwave radiation (positive downwards).

- *L_ia*: number [$\mathbf{W/m^2}$]
  The calculated incoming longwave radiation (positive downwards).

- *L_i0*: number [$\mathbf{W/m^2}$]
  The calculated clear sky incoming longwave radiation (positive downwards).

- *epsilon_0*: number (dimensionless)
  Atmospheric effective clearsky emmisivity (range 0-1).

- *black_body_radiation*: number [$\mathbf{W/m^2}$]
  Radiation emitted by black bodies at current air temperature. Stefan-Boltzmann's law.

43.1 brunt

Brunt, 1932. Default parameterization by Jensen et.al., 1990. FAO recommendation.

Used by bioclimate default net_radiation (see 13.1, page 101) .

- *b*: number [**1/sqrt(kPa)**]
  Parameter (default 0.14)
  Brunt 'b' parameter (vapor pressure factor).

- *a*: number (dimensionless)
  Parameter (default 0.34)
  Brunt 'a' parameter (offset).

43.2 idso_jackson

Idso and Jackson, 1969

43.3 brutsaert

Brutsaert, 1975

43.4 swinbank

Swinbank, 1963

43.5 satterlund

Satterlund, 1979

43.6 prata

Prata, 1996

Chapter 44

nitrification

The nitrification process, transforming ammonium into nitrate and nitrous oxide.
Used by horizon component Nitrification (see 35, page 183) .
< `component (N2O_fraction 0.02 [<fraction>])` >

- *N2O_fraction*: number [<**fraction**>]
  Parameter (default 0.02)
  Fraction of ammonium lost as N2O.

44.1 solute

k_10 * C / (k + C). Michaelis-Menten kinetics, with nitrification based on ammonium solute.

- *k*: number [**g/cm**^{3}]
  Parameter
  Half saturation constant.

- *k_10*: number [**h**$^{-1}$]
  Parameter
  Max rate.

- *heat_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 0 points)
  Heat factor.

- *water_factor*: plf [**cm** → <**none**>]
  Parameter (has default value with 0 points)
  Water potential factor.

44.2 soil

k_10 * M / (k + M). Michaelis-Menten kinetics, with nitrification based on total ammonium content.
Used by horizon component Nitrification (see 35, page 183) .

- *k*: number [**g N/cm**^{3}]
  Parameter (default 5e-005)
  Half saturation constant.

- *k_10*: number [**g N/cm^3/h**]
  Parameter (default 2.08333e-007)
  Max rate.

- *heat_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 0 points)
  Heat factor.

- *water_factor*: plf [**cm** → <**none**>]
  Parameter (has default value with 0 points)
  Water potential factor.

Chapter 45

number

Generic representation of numbers.

Used by select component expr (see 62, page 305) , and biopore component density (see 14, page 107) .

45.1 initial_C

Find initial content from concentration.

Used by chemical NO3 initial (see 17.4, page 122) , and chemical NH4 initial (see 17.5, page 122) .

- *C*: number [**g/cm**^{3}]
  Parameter
  Initial concentration in soil water.

45.2 initial_NO3

A 'initial_C' model (see 45.1, page 235) build into Daisy.

Initial NO3 concentration in soil water.

45.3 initial_NH4

A 'initial_C' model (see 45.1, page 235) build into Daisy.

Initial NH4 concentration in soil water.

45.4 const

Always give the specified value.

Used by chemical default initial (see 17.1, page 115) .

- *value*: number [<**user**>]
  Parameter
  Fixed value for this number.

45.5 initial_zero

A 'const' model (see 45.4, page 235) build into Daisy.

Initial zero concentration in soil water.

45.6 zero_gradient

A 'const' model (see 45.4, page 235) build into Daisy.
Assume same concentration in groundwater as in the bottom of the soil profile.

45.7 if

Select between two numbers depending on a boolean expression.

- *if*: **boolean** component (see chapter 15)
  Select which number to use.
- *then*: **number** component (see chapter 45)
  Use this if true.
- *else*: **number** component (see chapter 45)
  Use this if false.

45.8 horizon

Find soil value at specific horizon.

- *horizon*: **horizon** component (see chapter 35)
  The soil horizon whose properties we want to examine.
- *h*: **number** component (see chapter 45)
  The tension we want to compare with.
- *top_soil*: boolean (see section 1.4.2)
  Parameter
  Set this to true for the A horizon.

45.9 soil_Theta

A 'horizon' model (see 45.8, page 236) build into Daisy.
Find water content (Theta) for a given pressure (h).

45.10 soil_K

A 'horizon' model (see 45.8, page 236) build into Daisy.
Find hydraulic conductivity (K) for a given pressure (h).

45.11 soil_heat_capacity

A 'horizon' model (see 45.8, page 236) build into Daisy.
Find heat capacity for a given pressure (h).

45.12 soil_heat_conductivity

A 'horizon' model (see 45.8, page 236) build into Daisy.
Find heat conductivity for a given pressure (h).

45.13 x

The value of the symbol 'x' in the current scope.
Used by xysource loop x (see 84.3, page 401) .

45.14 get

Get the value of symbol in the current scope.

- *name*: string (see section 1.4.5)
  Parameter
  Name of a the symbol.

- *dimension*: string (see section 1.4.5)
  Parameter
  Expected dimension for the symbol.

45.15 fetch

Fetch the value and dimension in the current scope.

- *name*: string (see section 1.4.5)
  Parameter
  Name of a the symbol.

45.16 child

Numbers based on another number.

- *value*: **number** component (see chapter 45)
  Operand for this function.

45.17 identity

A 'child' model (see 45.16, page 237) build into Daisy.
Pass value unchanged.

- *dimension*: string (see section 1.4.5)
  Optional parameter
  Dimension of this value.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

45.18 convert

A 'child' model (see 45.16, page 237) build into Daisy.
Convert to specified dimension.

- *value*: **number** component (see chapter 45)
  Operand for this function.

- *dimension*: string (see section 1.4.5)
  Parameter
  Dimension to convert to.

45.19 dim

A 'child' model (see 45.16, page 237) build into Daisy.
Specify dimension for number.

- *value*: **number** component (see chapter 45)
  Operand for this function.

- *dimension*: string (see section 1.4.5)
  Parameter
  Dimension to use.

- *warn_known*: boolean (see section 1.4.2)
  Parameter (default true)
  Issue a warning if the dimensions is already known.

45.20 source

Extract information from a time series.

- *source*: **source** component (see chapter 65)
  The time series we want to extract a number from.

- *begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  Ignore values before or at this date.

- *end*: **Time** fixed component (see section 85.21)
  Optional submodel
  Ignore values after this date.

45.21 source_unique

A 'source' model (see 45.20, page 238) build into Daisy.
Find unique number in time series.

45.22 source_average

A 'source' model (see 45.20, page 238) build into Daisy.
Find average number in time series.

45.23 source_sum

A 'source' model (see 45.20, page 238) build into Daisy.
Calculate the sum of the values in a time series.

45.24 source_increase

A 'source' model (see 45.20, page 238) build into Daisy.
Find increase in value during time series.

45.25 log10

Take the base 10 logarithm of its argument.

- *operand*: **number** component (see chapter 45)
  Operand for this function.

45.26 ln

Take the natural logarithm of its argument.

- *operand*: **number** component (see chapter 45)
  Operand for this function.

45.27 exp

Take the exponential of its argument.

- *operand*: **number** component (see chapter 45)
  Operand for this function.

45.28 sqrt

Take the square root of its argument.

- *operand*: **number** component (see chapter 45)
  Operand for this function.

45.29 sqr

Take the square of its argument.

- *operand*: **number** component (see chapter 45)
  Operand for this function.

45.30 pow

Raise 'base' to the power of 'exponent'.

- *base*: **number** component (see chapter 45)
  The base operand for this function.
- *exponent*: **number** component (see chapter 45)
  The exponent operand for this function.

45.31 max

Use the largest value of its operands.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.32 min

Use the smallest value of its operands.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.33 *

Use the product of its operands.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.34 +

Use the sum of its operands.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.35 -

Negate number or subtract numbers. With one operand, negates it. With more than one operand, subtracts all but the first from the first.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.36 /

Divide the first operand by the rest.

- *operands*: **number** component (see chapter 45) sequence
  The operands for this function.

45.37 depth

Find soil value at specific depth.

- *column*: **column** component (see chapter 19)
  The soil column whose properties we want to examine.
- *h*: **number** component (see chapter 45)
  The tension we want to compare with.
- *z*: **number** component (see chapter 45)
  The height we want to compare with.

45.38 depth_Theta

A 'depth' model (see 45.37, page 240) build into Daisy.
Find water content (Theta) for a given pressure (h).

45.39 depth_K

A 'depth' model (see 45.37, page 240) build into Daisy.
Find water conductivity (K) for a given pressure (h).

45.40 soil_h

Find pressure (h) for a given water content (Theta).

- *horizon*: **horizon** component (see chapter 35)
  The soil horizon whose properties we want to examine.
- *Theta*: **number** component (see chapter 45)
  The water content we want to compare with.
- *top_soil*: boolean (see section 1.4.2)
  Parameter
  Set this to true for the A horizon.

45.41 let

Bind symbols in 'clauses' in a new scope, and evaluate 'expr' in that scope.

- *clauses*: submodel (see section 1.4.7) sequence
  List of identifiers and values to bind in this scope.

  < *identifier expr* >

  - *identifier*: string (see section 1.4.5)
    Parameter
    Identifier to bind.
  - *expr*: **number** component (see chapter 45)
    Value to give it.
- *expr*: **number** component (see chapter 45)
  Expression to evaluate.

45.42 plf

Look up argumen in a piecewise linear function.

- *operand*: **number** component (see chapter 45)
  Operand for this function.
- *domain*: string (see section 1.4.5)
  Parameter (default '<unknown>')
  Unit for the operand of the function.
- *range*: string (see section 1.4.5)
  Parameter (default '<unknown>')
  Unit for the operand of the function.

- *points*: submodel (see section 1.4.7) sequence
  List of points (x y) defining the piecewise linear function. The x values must be ordered lowest first.

  < ***x y*** >

  - x: number [<**user**>]
    Parameter
    Operand.
  - y: number [<**user**>]
    Parameter
    Value.

Chapter 46

organic

Turnover of organic matter in the soil.

46.1 none

Ignore all soil organic matter dynamics.

46.2 default

Mineralization and immobilization in soil. See also [Hansen et al., 1991, Bruun et al., 2003]

Used by column default OrganicMatter (see 19.1, page 130) , and column Foulum OrganicMatter (see 19.4, page 132) .

- *ClayOM*: **ClayOM** component (see chapter 4)
  Component (default 'old')
  Clay effect model.

- *domsorp*: **domsorp** component (see chapter 25) sequence
  Component (default: an empty sequence)
  Interchange between DOM and SOM pools.

- *Bioincorporation*: **Bioincorporation** fixed component (see section 85.3)
  Submodel (has fully specified default value)
  Biological incorporation of litter.

- *am*: **am** component (see chapter 11) sequence
  Component (has default value with length 1)

  ```
  (am root)
  ```

  Parameter description:
  Added organic matter pools.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

```
(cite "daisy-fertilizer" "daisy-somnew")
```

Parameter description:
BibTeX keys that would be relevant for this model or paramterization.

- *heat_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 0 points)
  Default heat factor, used if not specified by OM pool.

- *water_factor*: plf [**cm** → <**none**>]
  Parameter (has default value with 0 points)
  Default water potential factor, used if not specified by OM pool. If the PLF is empty, a build-in PLF of pF will be used instead. It is 0.6 at pF $<$ 0, 1.0 at 1.5 $<$ pF $<$ 2.5, and 0 at pF $>$ 6.5.

- *active_underground*: boolean (see section 1.4.2)
  Parameter (default false)
  Set this flag to turn on mineralization below the root zone.

- *K_NH4*: number [$\mathbf{h}^{-1}$]
  Parameter (default 0.020833)
  Maximal immobilization rate for ammonium.

- *K_NO3*: number [$\mathbf{h}^{-1}$]
  Parameter (default 0.020833)
  Maximal immobilization rate for nitrate.

- *CO2_threshold*: number [$\mathbf{h}^{-1}$]
  Parameter (default 0.0001)
  Turnover rate above which pools will contribute to 'CO2_fast'.

- *buffer*: submodel (see section 1.4.7)
  Submodel (has fully specified default value)

  ```
  (buffer )
  ```

  Parameter description:
  Buffer between AOM pools and SOM.

  ```
  < (N)
    (C)
    (where 1)
    (turnover_rate 1 [h^-1])
    (turnover_halftime turnover_halftime) >
  ```

  - *N*: number [**g N/cm**^{3}] soil cells
    State variable (default: an empty sequence)
    Buffer nitrogen content.
  - *C*: number [**g C/cm**^{3}] soil cells
    State variable (default: an empty sequence)
    Buffer carbon content.
  - *where*: integer
    Parameter (default 1)
    The SOM pool to move the buffer content into. The first and slow SOM pool is numbered '0', the second and faster is numbered '1'.

  - *turnover_rate*: number [$\mathbf{h}^{-1}$]
    Parameter (default 1)
    Turnover rate from buffer into SOM. Ignored if you specify 'turnover_halftime'.
  - *turnover_halftime*: number [**h**]
    Optional parameter
    Turnover halftime from buffer into SOM. Overrules 'turnover_rate' if specified.

- *smb*: **SMB** component (see chapter 7) sequence
  Component (has default value with length 2)

  ```
  (smb "SMB-SLOW"
       "SMB-FAST")
  ```

  Parameter description:
  Soil MicroBiomass pools. Initial value will be estimated based on equilibrium with AM and SOM pools.

- *som*: **SOM** component (see chapter 8) sequence
  Component (has default value with length 3)

  ```
  (som "SOM-SLOW"
       "SOM-FAST"
       "SOM-INERT")
  ```

  Parameter description:
  Soil Organic Matter pools.

- *initial_SOM*: submodel (see section 1.4.7) sequence
  Optional submodel
  Layered initialization of soil SOM content.

  < *end weight* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *weight*: number [**kg C/m**^{2}]
    Parameter
    Organic carbon content of this layer.

- *dom*: **DOM** fixed component (see section 85.20) sequence
  Submodel (default: an empty sequence)
  Dissolved Organic Matter pools.

- *smb_tillage_factor*: plf [**d** → <**none**>] sequence
  Parameter (default: an empty sequence)
  Tillage influence on turnover rates for each SMB pool. If no value is given, tillage will have no influence.

- *som_tillage_factor*: plf [**d** → <**none**>] sequence
  Parameter (default: an empty sequence)
  Tillage influence on SOM turnover rates for each SOM pool. If no value is given, tillage will have no influence.

- *min_AM_C*: number [**g C/m**^{2}]
  Parameter (default 0.5)
  Minimal amount of carbon in AOM ensuring it is not removed.

- *min_AM_N*: number [**g N/m^2**]
  Parameter (default 0.05)
  Minimal amount of nitrogen in AOM ensuring it is not removed.

- *init*: submodel (see section 1.4.7)
  Submodel (has fully specified default value)

  ```
  (init )
  ```

  Parameter description:
  Parameters for initialization of the SOM and SMB pools.

  If the C content of all the pools have been specified explicitly, use those values. Otherwise, get the total C content from either the 'initial_SOM' parameter if specified, or else from the humus content specified in the soil horizons.

  If 'SOM_fractions' has been specified, the pools will be initialized assuming the SMB pools are in equilibrium. Otherwise, also SOM pools expect the first will be assumed to be in equilibrium as well.

  ```
  <   (h -100 [cm])
      (root 800 [kg C/ha/y])
      (T T)
      (end end)
      (efficiency 0.5 0.5)
      (fractions 0 1 0)
      (dist 7 [cm])
      (input input)
      (bioinc 0 [kg C/ha/y])
      (variable_pool variable_pool)
      (variable_pool_2 variable_pool_2)
      (background_mineralization background_mineralization)
      (SOM_limit_where 0)
      (SOM_limit_lower 0.3 0.7 0)
      (SOM_limit_upper 0.7 0.3 0)
      (debug_equations)
      (debug_rows true)
      (debug_to_screen false)
      (top_summary top_summary) >
  ```

  - *h*: number [**cm**]
    Parameter (default -100)
    Pressure used for equilibrium.
  - *root*: number [**kg C/ha/y**]
    Parameter (default 800)
    Amount of carbon added to the organic matter system from dead roots.
    This is part of the total amount specified by the 'input' parameter.
  - *T*: number [**dg C**]
    Optional parameter
    Temperature used for equilibrium.
    By default, the yearly average from the weather component will be used.
  - *end*: number [**cm**]
    Optional parameter
    Depth of non-root input.
    The input will distributes uniformly down to this size, after subtracting the part of the input allocated to the 'root' parameter.

By default, the end of the first horizon will be used.

– *efficiency*: number [<**fraction**>] sequence
Parameter (has default value with length 2)

```
(efficiency 0.5 0.5 [])
```

Parameter description:
The efficiency this pool can be digested by each of the SMB pools. This is only used if you specify the input parameter.

– *fractions*: number [<**fraction**>] sequence
Parameter (has default value with length 3)

```
(fractions 0 1 0 [])
```

Parameter description:
Desitinations for AOM input. The first numbers corresponds to each SMB pool, while the last number correspond to the SOM buffer. This is only used if you specify the input parameter.

– *dist*: number [**cm**]
Parameter (default 7)
Distance to go down in order to decrease the root density to half the original.

– *input*: number [**kg C/ha/y**]
Optional parameter
Amount of carbon added to the organic matter system.

If this is unspecifed, the input rate from the inital added matter pools will be used instead.

– *bioinc*: number [**kg C/ha/y**]
Parameter (default 0)
Amount of carbon added to the organic matter system from bioincorporation.

This is part of the total amount specified by the 'input' parameter.

– *variable_pool*: integer
Optional parameter
If neither the C content nor 'SOM_fractions' are specified, equilibrium is assumed for all SOM pools except the one specified by this parameter. If you set this to -1 (or any number nor corresponding to a SOM pool), equilibrium will be assumed for all pools, and the humus content specified by the horizon will be ignored. Note, the numbering is zero-based, so '0' specifies SOM1. By default, the slowest active pool will be used.

– *variable_pool_2*: integer
Optional parameter
If 'background_mineralization' is specified, this pool is no longer assumed to be in equilibrium. Note, the numbering is zero-based, so '0' specifies SOM1. By default, the second slowest active pool will be used.

– *background_mineralization*: number [**kg N/ha/y**]
Optional parameter
The background mineralization is the mineralization from all SMB and SOM pools, but not from the AOM pools.

If neither the C content of individual pools nor 'SOM_fractions' are speecified, the SOM and SMB pools will be initialized so all pools in the top soil (above 'end', usually the first horizon) are in equilibrium except those specified by 'variable_pool' and 'variable_pool_2', usually SOM1

and SOM2. These two will be initialized so the background mineralization will be the specified number. The subsoil is not affected by this parameter.

If the background mineralization is unspecified, 'variable_pool_2' will be assumed to be in equilibrium instead.

- *SOM_limit_where*: integer
  Parameter (default 0)
  This is the SOM pool that must be within the limits specified by 'SOM_limit_lower' and 'SOM_limit_upper'. Use negative number to disable. Note, the numbering is zero-based, so '0' specifies SOM1.
- *SOM_limit_lower*: number [<**fraction**>] sequence
  Parameter (has default value with length 3)

  ```
  (SOM_limit_lower 0.3 0.7 0 [])
  ```

  Parameter description:
  Lower limit for for automatic SOM partitioning.

  The SOM pool specified by 'SOM_limit_where' must contain at least the fraction of the total SOM content given in this list, where the first number correspond to the SOM1 fraction, the second number to SOM2, etc. If the fraction is below the one given in this list, the SOM partitioning in this list will be used instead.

  If the SOM partitioning have been specified directly, either by the 'SOM_fractions' horizon parameter or by specifying the C content of each pool, this parameter will be ignored. The limit is also ignore for soil layers below 'end'.
- *SOM_limit_upper*: number [<**fraction**>] sequence
  Parameter (has default value with length 3)

  ```
  (SOM_limit_upper 0.7 0.3 0 [])
  ```

  Parameter description:
  Upper limit for for automatic SOM partitioning. Works like 'SOM_limit_lower'.
- *debug_equations*: integer sequence
  Parameter (default: an empty sequence)
  Print equations used for initialization for the specified intervals.
- *debug_rows*: boolean (see section 1.4.2)
  Parameter (default true)
  Print summari information for each row.
- *debug_to_screen*: boolean (see section 1.4.2)
  Parameter (default false)
  If true, print debug information to screen, else to the 'daisy.log' file.
- *top_summary*: string (see section 1.4.5)
  Optional parameter
  Name of file to print a summary of the organic carbon and nitrogen content in the zone down to the 'end' parameter. If unspecified, no such file will be generated, but the summary will still be found in 'daisy.log'.

Log Variables

- *humus*: number [$\mathbf{g/cm^3}$] soil cells
  Total organic matter in the soil layer.

- *CO2*: number [**g CO2-C/cm^3/h**] soil cells
  CO2 evolution in soil from all pools.
- *NO3_source*: number [**g N/cm^3/h**] soil cells
  Mineralization this time step (negative numbers mean immobilization).
- *NH4_source*: number [**g N/cm^3/h**] soil cells
  Mineralization this time step (negative numbers mean immobilization).
- *fertilized_N*: number [**g N/cm^2/h**]
  Amount of organic bound nitrogen applied. This includes nitrogen incorporated directly in the soil.
- *fertilized_C*: number [**g C/cm^2/h**]
  Amount of organic bound carbon applied. This includes carbon incorporated directly in the soil.
- *tillage_N_top*: number [**g N/m^2/h**]
  Amount of nitrogen added to surface during tillage. This is a negative number.
- *tillage_C_top*: number [**g C/m^2/h**]
  Amount of carbon added to surface during tillage. This is a negative number.
- *tillage_N_soil*: number [**g N/cm^3/h**] soil cells
  Amount of nitrogen added to soil during tillage.
- *tillage_C_soil*: number [**g C/cm^3/h**] soil cells
  Amount of carbon added to surface during tillage.
- *total_C*: number [**g C/cm^3**] soil cells
  Total organic C in the soil layer.
- *total_N*: number [**g N/cm^3**] soil cells
  Total organic N in the soil layer.
- *CO2_fast*: number [**g CO2-C/cm^3/h**] soil cells
  CO2 evolution in soil from pools faster than 'CO2_threshold'.
- *top_CO2*: number [**g CO2-C/cm^2/h**]
  CO2 evolution at surface.
- *top_DM*: number [**kg DM/m^2**]
  Added organic dry matter on top of surface.
- *abiotic_factor*: number (dimensionless) soil cells
  Product of current heat and water factors.

Chapter 47

parser

To start the simulation, many parameters must be specified and state variables must be given an initial value. It is the responsibility of the 'parser' component to read these data from an external source (typically a setup file), and convert them into the internal format.

Used by Toplevel @ input (see 85.27, page 445) .

47.1 file

Read a setup file containing lots of parentheses.

- *where*: string (see section 1.4.5)
  Parameter
  File to read from.

Chapter 48

pet

The 'pet' component should calculate the potential evapotranspiration from meteorological data, as well as the crop and soil state.
< `component` >

Log Variables

- *wet*: number [**mm/h**]
  Potential evapotranspiration for a wet system.
- *dry*: number [**mm/h**]
  Potential evapotranspiration for a dry system.
- *reference_evapotranspiration*: number [**mm/h**]
  Reference evapotranspiration for a dry system.

48.1 PM

Potential evopotranspiration using Penman-Monteith.

- *rb*: number [**s/m**]
  Parameter (default 20)
  Boundary layer resistance for wet surface. used for bare soil only.

48.2 weather

Potential evopotranspiration using weather data.

48.3 makkink

Potential evopotranspiration using Makkink's Equation.

48.4 Hargreaves

Potential evopotranspiration using Samani and Hargreaves.

Hargreaves, G.H., and Samani, Z.A. (1982) Estimating potential evapotranspiration. Tech. Note, J. Irrig. and Drain. Engrg., ASCE, 108(3):225-230.

Hargreaves, G.H., and Samani, Z.A. (1985) Reference crop evapotranspiration from temperature. Appl. Engrg. in Agric., 1(2):96-99.

48.5 FAO_PM

Potential evopotranspiration using Penman-Monteith.

- *rb*: number [**s/m**]
  Parameter (default 20)
  Boundary layer resistance for wet surface.

- *use_wet*: boolean (see section 1.4.2)
  Parameter (default true)
  Use wet PM for wet surface.

  This flag is for compatibility with older version of Daisy, which always used dry PM (and which overestimated the effect of wet PM).

Log Variables

- *reference_evapotranspiration_wet*: number [**mm/h**]
  Reference evapotranspiration for a wet system.

- *potential_evapotranspiration_wet*: number [**mm/h**]
  Potential evapotranspiration for a wet system.

- *potential_evapotranspiration_dry*: number [**mm/h**]
  Potential evapotranspiration for a dry system.

- *Rn*: number [$\mathbf{W/m^2}$]
  Reference net radiation.

- *G*: number [$\mathbf{W/m^2}$]
  Soil heat flux.

Chapter 49

phenology

The development process.

```
< component   (DS -1 [])
              (day_length 0 [h])
              (DAP 0 [d])
              (partial_day_length 0 [h]) >
```

- *DS*: number (dimensionless)
  State variable (default -1)
  Development Stage.

- *day_length*: number [**h**]
  State variable (default 0)
  Number of light hours yesterday.

- *DAP*: number [**d**]
  State variable (default 0)
  Days after planting.

- *partial_day_length*: number [**h**]
  State variable (default 0)
  Number of light hours this day, so far.

49.1 default

Default crop phenology model.

Used by crop Maize Devel (see 22.2, page 145) , crop Pioneer Maize Devel (see 22.5, page 145) , crop Ikuwala Maize Devel (see 22.3, page 145) , crop Silage Maize Devel (see 22.4, page 145) , crop Pea Devel (see 22.6, page 145) , crop Potato; Koege Devel (see 22.8, page 145) , crop Potato; SCRI Devel (see 22.9, page 145) , crop Potato; FertOrgaNic Devel (see 22.10, page 145) , crop Potato; Agria Devel (see 22.11, page 146) , crop Potato; Folva Devel (see 22.12, page 146) , crop Potato; Triada Devel (see 22.14, page 146) , crop Rye Devel (see 22.15, page 146) , crop Spring Barley Devel (see 22.16, page 146) , crop Spring Barley; Foulum Devel (see 22.17, page 146) , crop Sugar Beet Devel (see 22.18, page 146) , crop Fodder Beet Devel (see 22.19, page 146) , crop Spring Rape Devel (see 22.20, page 146) , crop Spring Wheat Devel (see 22.21, page 147) , crop Winter Barley Devel (see 22.22, page 147) , crop Winter Barley; Koge Devel (see 22.24, page 147) , crop Winter Rape Devel (see 22.25, page 147) , crop Winter Wheat Devel (see 22.26, page 147) , crop Winter Wheat; Eest Devel (see 22.27, page 147) , crop Winter Wheat; Foulum Devel (see 22.28, page 147) , crop Grass to grain Devel (see 22.29, page 147) , crop

Grass Devel (see 22.30, page 147) , crop Ryegrass Devel (see 22.31, page 147) , crop Wclover Devel (see 22.32, page 148) , crop Beetroot Devel (see 22.33, page 148) , crop Broccoli Devel (see 22.34, page 148) , crop Broccoli - transplanted Devel (see 22.35, page 148) , crop Brussels sprouts Devel (see 22.36, page 148) , crop Brussels sprouts - transplanted Devel (see 22.37, page 148) , crop Celeriac Devel (see 22.38, page 148) , crop Celeriac - transplanted Devel (see 22.39, page 148) , crop Potato; SCRI - AArhus Devel (see 22.40, page 148) , crop Early potato Devel (see 22.41, page 148) , crop White cabbage Devel (see 22.42, page 149) , crop White cabbage - transplanted Devel (see 22.43, page 149) , crop Early white cabbage - transplanted Devel (see 22.44, page 149) , crop Onion Devel (see 22.45, page 149) , crop Onion - planting of sets Devel (see 22.46, page 149) , crop Rug Devel (see 22.47, page 149) , crop Vaarbyg Devel (see 22.48, page 149) , crop Vinterbyg Devel (see 22.49, page 149) , crop Vinterhvede Devel (see 22.50, page 149) , crop Froegraes Devel (see 22.51, page 150) , crop Graes Devel (see 22.52, page 150) , crop Silomajs Devel (see 22.53, page 150) , crop Aert Devel (see 22.54, page 150) , and crop Vinterraps Devel (see 22.55, page 150) .

- *EmrTSum*: number [**dg C d**]
  Parameter
  Soil temperature sum at emergence.

- *EmrSMF*: plf [**cm → d**]
  Parameter (has default value with 4 points)

  ```
  (EmrSMF (-1000 1) (-150 1) (-50 1) (-30 1))
  ```

  Parameter description:
  Soil moisture (h-function) effect on emergense.

- *DS_Emr*: number [**DS**]
  Parameter (default 0.01)
  Development stage at emergence.

- *DSRate1*: number [**DS/d**]
  Parameter
  Development rate in the vegetative stage.

- *DSRate2*: number [**DS/d**]
  Parameter
  Development rate in the reproductive stage.

- *TempEff1*: plf [**dg C → <none>**]
  Parameter
  Temperature effect, vegetative stage.

- *TempEff2*: plf [**dg C → <none>**]
  Parameter
  Temperature effect, reproductive stage.

- *PhotEff1*: plf [**h → <none>**]
  Parameter
  Photoperiode effect, vegetative stage.

- *DSMature*: number [**DS**]
  Parameter (default 2)
  Development stage at maturation.

- *DSRepeat*: number [**DS**]
  Parameter (default 4)
  Development stage when DS set back is activated.

- *DSSetBack*: number [**DS**]
  Parameter (default 1.7)
  Development stage set back at DSRepeat.

- *defined_until_ds*: number [**DS**]
  Parameter (default 2)
  This parameterization is only valid until the specified development state.

49.2 TSum

Crop phenology model purely based on temperature sums. The length of emergence, and the vegetative and reproductive fase are all based on the specified temperature sums. Temperatures below the specified thresholds do not contribute to the temeprature sum. Cut stress and leaf respiration does not affect this phenology model.

- *EmrTSum*: number [**dg C d**]
  Parameter
  Soil temperature sum at emergence.

- *EmrThrs*: number [**dg C**]
  Parameter (default 0)
  Minimum soil temperature for emergence. Temperature below this will not count in the sum.

- *VegTSum*: number [**dg C d**]
  Parameter
  Air temperature sum for vegetative fase.

- *VegThrs*: number [**dg C**]
  Parameter (default 0)
  Minimum air temperature for development in vegetative fase. Temperature below this will not count in the sum.

- *RepTSum*: number [**dg C d**]
  Parameter
  Air temperature sum for vegetative fase.

- *RepThrs*: number [**dg C**]
  Parameter (default 0)
  Minimum air temperature for development in vegetative fase. Temperature below this will not count in the sum.

Chapter 50

photosynthesis

Leaf photosynthesis.

< `component` `(description` *`description`*`)`
`(cite)`
`(min_PAR 0.1 [W/m`^{2}`]) >`

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *min_PAR*: number [$\mathbf{W/m^2}$]
  Parameter (default 0.1)
  Minimum PAR at top of canopy for photosynthesis. If radiation is below this amount, photosynthesis will be disabled.

50.1 GL

Goudriaan and Laar, 1978.

Used by crop default LeafPhot (see 22.1, page 143) , crop Maize LeafPhot (see 22.2, page 145) , crop Pioneer Maize LeafPhot (see 22.5, page 145) , crop Pea LeafPhot (see 22.6, page 145) , crop Potato; Koege LeafPhot (see 22.8, page 145) , crop Potato; SCRI LeafPhot (see 22.9, page 145) , crop Potato; FertOrgaNic LeafPhot (see 22.10, page 145) , crop Potato; Folva LeafPhot (see 22.12, page 146) , crop Potato; Triada LeafPhot (see 22.14, page 146) , crop Rye LeafPhot (see 22.15, page 146) , crop Spring Barley LeafPhot (see 22.16, page 146) , crop Sugar Beet LeafPhot (see 22.18, page 146) , crop Fodder Beet LeafPhot (see 22.19, page 146) , crop Spring Rape LeafPhot (see 22.20, page 146) , crop Spring Wheat LeafPhot (see 22.21, page 147) , crop Winter Barley LeafPhot (see 22.22, page 147) , crop Winter Barley; Koge LeafPhot (see 22.24, page 147) , crop Winter Rape LeafPhot (see 22.25, page 147) , crop Winter Wheat LeafPhot (see 22.26, page 147) , crop Winter Wheat; Eest LeafPhot (see 22.27, page 147) , crop Grass to grain LeafPhot (see 22.29, page 147) , crop Ryegrass LeafPhot (see 22.31, page 147) , crop Wclover LeafPhot (see 22.32, page 148) , crop Beetroot LeafPhot (see 22.33, page 148) , crop Broccoli LeafPhot (see 22.34, page 148) , crop Brussels sprouts LeafPhot (see 22.36, page 148) , crop Celeriac LeafPhot (see 22.38, page 148) , crop Potato; SCRI

- AArhus LeafPhot (see 22.40, page 148) , crop Early potato LeafPhot (see 22.41, page 148) , crop White cabbage LeafPhot (see 22.42, page 149) , crop Early white cabbage - transplanted LeafPhot (see 22.44, page 149) , crop Onion LeafPhot (see 22.45, page 149) , crop Onion - planting of sets LeafPhot (see 22.46, page 149) , crop Rug LeafPhot (see 22.47, page 149) , crop Vaarbyg LeafPhot (see 22.48, page 149) , crop Vinterbyg LeafPhot (see 22.49, page 149) , crop Vinterhvede LeafPhot (see 22.50, page 149) , crop Froegraes LeafPhot (see 22.51, page 150) , crop Graes LeafPhot (see 22.52, page 150) , crop Silomajs LeafPhot (see 22.53, page 150) , crop Aert LeafPhot (see 22.54, page 150) , and crop Vinterraps LeafPhot (see 22.55, page 150) .

- *Qeff*: number [**(g CO2/m^2/h)/(W/m^2)**]
  Parameter
  Quantum efficiency at low light.

- *Fm*: number [**g CO2/m^2/h**]
  Parameter
  Maximum assimilation rate.

- *TempEff*: plf [**dg C** → <**none**>]
  Parameter
  Temperature factor for assimilate production.

- *DSEff*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (DSEff (0 1) (2 1))
  ```

  Parameter description:
  Development stage factor for assimilate production.

- *DAPEff*: plf [**d** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (DAPEff (0 1) (2 1))
  ```

  Parameter description:
  Age factor for assimilate production. Age is given as day after planting.

50.2 Farquhar

Base parameterization for Farquhar derived photosynthesis models.

Farquhar et al. (1980) photosynthesis and Ball et al. (1987) stomataconductance model coupled as described by Collatz et al., 1991.

- *Xn*: number [**mol/mol/s**]
  Parameter (default 0.00116)
  Slope of relationship between leaf rubisco N and Vmax. Xn = 1.16E-3 mol/mol/s for wheat (de Pury & Farquhar, 1997)

- *Gamma25*: number [**Pa**]
  Parameter (default 3.69)
  CO2 compensation point of photosynthesis. Gamma25 = 3.69 Pa for wheat (Collatz et al., 1991)

- *Ea_Gamma*: number [**J/mol**]
  Parameter (default 29000)
  Actimation energy for Gamma. Ea_Gamma = 29000 (Jordan & Ogren, 1984)

- *N-dist*: **rubiscoNdist** component (see chapter 57)
  Component (default 'exp')
  Rubisco N-distribution in the canopy layer.

- *Stomatacon*: **stomatacon** component (see chapter 66)
  Component (default 'Leuning')
  Stomata conductance of water vapor.

Log Variables

- *LAI*: number []
  Leaf area index for the canopy used in photosynthesis.

- *ABA_effect*: number (dimensionless)
  Water stress effect induced by ABA and crown water potential

- *ci_vector*: number [**Pa**] canopy intervals
  CO2 pressure in Stomatal in each layer.

- *Vm_vector*: number [**mmol/m^2/s**] canopy intervals
  Photosynthetic capacity in each layer.

- *Jm_vector*: number [**mmol/m^2/s**] canopy intervals
  Potential rate of electron transport in each layer.

- *pn_vector*: number [**mol/m^2 leaf/s**] canopy intervals
  Net photosynthesis.

- *cs_vector*: number [**Pa**] canopy intervals
  CO2 pressure at leaf surface.

- *hs_vector*: number [<**fraction**>] canopy intervals
  Relative humidity at leaf surface.

- *gs_vector*: number [**mol/m^2 leaf/s**] canopy intervals
  Stomata cunductance in each layer.

- *Nleaf_vector*: number [**mol N/m^2**] canopy intervals
  Distribution of photosynthetic N-leaf.

- *Ass_vector*: number [**mol CH2O/m^2/h**] canopy intervals
  Brutto assimilate.

- *LAI_vector*: number [**m^2 leaf/m^2 field**] canopy intervals
  LAI.

- *ci_middel*: number [**Pa**]
  Stomata average CO2 pressure.

- *Gamma*: number [**Pa**]
  CO2 compensation point of photosynthesis.

- *gbw*: number [**mol/m^2 leaf/s**]
  Boundary lauer conductance.

- *gs*: number [**mol/m^2 field/s**]
  Stomata conductance.

- *gs_ms*: number [**m/s**]
  Stomata conductance.

- *Ass*: number [**g CH2O/m^2/h**]
  'Net' leaf assimilate of CO2 (brutto photosynthesis).

- *Res*: number [**g CH2O/m^2/h**]
  Farquhar leaf respiration.

- *PAR_*: number [**mol/m^2/h**]
  PAR.

- *Vmax*: number [[**mmol/m^2/s**]]
  Photosynthetic Rubisco capacity.

- *jm*: number [[**mmol/m^2/s**]]
  Potential rate of electron transport.

- *leafPhotN*: number [[**mol N/m^2**]]
  Content of photosynthetic active leaf N.

- *fraction_total*: number []
  Fraction of leaf contributing to the photosynthesis.

50.3 FC_C4

A 'Farquhar' model (see 50.2, page 260) build into Daisy.
C4 photosynthesis and stomatal conductance model by Collatz et al., 1992.

- *alpha*: number [**mol/mol**]
  Parameter (default 0.04)
  Initial slope of photosynthetic light response. alpha = 0.04 (Collatz et al., 1992)

- *beta*: number (dimension not specified)
  Parameter (default 0.93)
  Curvanture parameter

- *Q10k*: number (dimension not specified)
  Parameter (default 1.8)
  Q10k = 1.8 (Collatz et al., 1992)

- *Q10vm*: number (dimension not specified)
  Parameter (default 2.4)
  Q10vm = 2.4 (Collatz et al., 1992)

- *Q10rd*: number (dimension not specified)
  Parameter (default 2)
  Q10rd = 2.0 (Collatz et al., 1992)

- *kj*: number (dimension not specified)
  Parameter (default 0.6)
  Initial slope of photosynthetic CO2 response, kj = 0.6 mol/m/s (Collatz et al., 1992)

- *paab*: number (dimension not specified)
  Parameter (default 0.86)
  Leaf absorbtivity to PAR. paab = 0.86 (Collatz et al., 1992)

- *theta*: number (dimension not specified)
  Parameter (default 0.83)
  Curvature parameter

50.4 FC_C3

A 'Farquhar' model (see 50.2, page 260) build into Daisy.
Photosynthesis for C3 crops described by Faquhar et al. (1980).

- *S*: number [**J/mol/K**]
  Parameter (default 710)
  Electron transport temperature response parameter,(De Pury & Farquhar, 1997)

- *TempEff*: plf [**dg C** → <**none**>]
  Parameter
  Temperature factor for assimilate production.

- *beta*: number (dimension not specified)
  Parameter (default 0.95)
  Curvature, Collatz et al., 1991

- *theta*: number (dimension not specified)
  Parameter (default 0.7)
  Curvature of leaf response of electron transport to irradiance, (De Pury & Farquhar, 1997

- *Kc25*: number [**Pa**]
  Parameter (default 40.4)
  Micahyaelis-Menten constant of Rubisco for CO2. Kc25 = 40.4 Pa for wheat (Collatz et al.,1991)

- *Ko25*: number [**Pa**]
  Parameter (default 24800)
  Micahaelis-Menten constant of Rubisco for O2 at 25 degrees. Ko25 = 24800 Pa for wheat (Collatz et al., 1991)

- *H*: number [**J/mol**]
  Parameter (default 220000)
  Curvature parameter of Jm, (De Pury & Farquhar, 1997)

- *c_Vm*: number (dimension not specified)
  Parameter (default 26.35)
  Temperature scaling constant for Vmax. c_Vm, = 26.35 (Bernacchi et al., 2001)

- *Ea_Vm*: number [**J/mol**]
  Parameter (default 65330)
  Activation energy for Vmax. Ea_Vm = 65330 J/mol (Ball, 1988)

- *Eda_Vm*: number [**J/mol**]
  Parameter (default 202900)
  Deactimation energy for Vmax. Eda_Vm = 202900 J/mol

- *Ea_Jm*: number [**J/mol**]
  Parameter (default 37000)
  Actimation energy for Jm. Ea_Jm = 37000 J/mol (Farquhar et al., 1980).

- *Ea_ko*: number [**J/mol**]
  Parameter (default 36000)
  Actimation energy for ko. Ea_ko 0 36000 J/mol (Badger & Collatz, 1977).

- *Ea_kc*: number [**J/mol**]
  Parameter (default 59400)
  Actimation energy for kc. Ea_kc = 59400 J/mol (Badger & Collatz, 1977)

- *Ea_rd*: number [**J/mol**]
  Parameter (default 66400)
  Actimation energy for rd. Ea_rd = 66400 J/mol (Farquhar et al., 1980)

- *Sv*: number [**J/mol/K**]
  Parameter (default 650)
  Entropy term. Sv = 650 J/mol/K

- *alfa*: number [**mol/mol**]
  Parameter (default 0.08)
  Fraction of PAR effectively absorbed by PSII,

Chapter 51

ponddamp

Dampening affect of ponding on soil erosion from rain.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

51.1 none

KH = 1.0
Used by reaction colgen_Jarvis99 ponddamp (see 55.8, page 285) .

51.2 EUROSEM

KH = exp (-b h) See also [Morgan et al., 1998]

- *b*: number [$\mathbf{mm}^{-1}$]
  Parameter (default 2)
  Exponential degradation coefficient. The range of 'b' is from 0.9 to 3.1, a default value of 2 is proposed by the EUROSEM project.

51.3 Park82

KH = 2.7183 * exp (-h / dds) See also [Park et al., 1982]

51.4 Hairsine91

KH = (h / dds)$^{-0}$.8 See also [Hairsine and Rose, 1991]

Chapter 52

program

Run a program.
Used by Toplevel @ run (see 85.27, page 445) .

52.1 hydraulic

Generate a table of the rentention curve and hydraulic conductivity.

- *hydraulic*: **hydraulic** component (see chapter 36)
  The hydraulic model to show in the table.
- *intervals*: integer
  Parameter (default 50)
  Number of intervals in the table.

52.2 batch

Run a sequence of programs.

- *directory*: string (see section 1.4.5)
  Parameter (default '.')
  Directory in which to initialize, check and run the programs.
- *run*: **program** component (see chapter 52) sequence
  List of programs to run. The programs will be run in the sequence listed.

52.3 document

Generate the components part of the reference manual.

- *format*: **format** component (see chapter 31)
  Component (default 'LaTeX')
  Text format used for the document.
- *where*: string (see section 1.4.5)
  Parameter (default 'components.tex')
  Name of file to store results in.
- *print_parameterizations*: boolean (see section 1.4.2)
  Parameter (default false)
  Include a copy of all loaded parameterizations in document.

52.4 docmodel

Document specific models.

- *component*: string (see section 1.4.5)
  Parameter
  Component to find the models in.
- *format*: **format** component (see chapter 31)
  Component (default 'LaTeX')
  Text format used for the document.
- *where*: string (see section 1.4.5)
  Parameter
  Name of file to store results in.
- *models*: string (see section 1.4.5) sequence
  Parameter
  Models to document.

52.5 gnuplot

Generate a gnuplot command file.

- *cd*: boolean (see section 1.4.2)
  Parameter (default true)
  Set this flag to add a 'cd' command to the current working directory. This is useful under MS Windows when dragging the file to a gnuplot icon.
- *command_file*: string (see section 1.4.5)
  Parameter (default 'daisy.gnuplot')
  File name for gnuplot commands.
- *extra*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  List of extra gnuplot commands. The commands will be inserted right before the list of graphs.
- *graph*: **gnuplot** component (see chapter 32) sequence
  Graphs to plot.

52.6 cd

Change working directory.

- *directory*: string (see section 1.4.5)
  Parameter
  Name of directory to change into.

52.7 write

Write string to file.

- *where*: string (see section 1.4.5)
  Parameter (default 'screen')
  File to write it in. If the value is 'screen', write the string to the screen.

- *what*: string (see section 1.4.5)
  Parameter
  String to write.

52.8 GP2D

Write root density table using 2D extension to Gerwitz and Page

- *DS*: number [**DS**]
  Parameter (default 2)
  Development stage [0-2]. Not currently used.

- *row_width*: number [**cm**]
  Parameter
  Distance between rows.

- *row_position*: number [**cm**]
  Parameter (default 0)
  Position of row on x-axis.

- *WRoot*: number [**g DM/m**^{2}]
  Parameter
  Totoal root dry matter.

- *Geometry*: **GeometryRect** fixed component (see section 85.26)
  Submodel (has partially specified default value)
  Discretization of the soil.

- *soil_depth*: number [**cm**]
  Parameter
  Limit on root depth by soil (no crops have roots below this).

- *crop_depth*: number [**cm**]
  Parameter
  Limit of root depth by crop (no soil have roots below this).

- *crop_width*: number [**cm**]
  Parameter
  Maximum horizontal distance of roots from plant.

52.9 cpedata

Manipulate data from Agrovand.

- *value*: string (see section 1.4.5)
  Parameter
  Tag used for value.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *factor*: number (dimensionless)
  Parameter (default 1)
  Multiply printed value with this number.

- *day*: string (see section 1.4.5)
  Parameter
  Tag used for day.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  If specified, only print entries after this date.

- *weight*: string (see section 1.4.5)
  Optional parameter
  Tag used for weight. If you specify this, 'value' will be given this weight.

- *debug*: integer
  Parameter (default 0)
  Debug level, 0 means no debug information.

- *handle*: string (see section 1.4.5)
  Parameter
  This option determine how the specified variable should be handled.

  average: print mean of values within timestep.

  difference: print difference between last value in timestep, and the last value in the previous timestep.

  sum: print sum of all values within current timestep.

- *origin*: **Time** fixed component (see section 85.21)
  Submodel (has partially specified default value)
  Day 1.

- *every_hour*: boolean (see section 1.4.2)
  Parameter (default false)
  If true, print zeroes for hours with no data.

52.10 Osvaldo

Find the modelling error. This is done between one set of measured data and multiple sets of simulated data.

- *pars*: integer
  Parameter
  Number of simulated dataset to compare.

- *par_file_prefix*: string (see section 1.4.5)
  Parameter
  Beginning of file name before simulation number.

- *par_file_suffix*: string (see section 1.4.5)
  Parameter
  End of file name after simulation number.

- *par_width*: integer
  Parameter
  Number of digits in simulated file name.

- *measured_file*: string (see section 1.4.5)
  Parameter
  Name of file containing measurments.

52.11 hmovie

Manipulate data from Agrovand.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *dim*: string (see section 1.4.5)
  Parameter
  Dimension of all axes.

- *cd*: boolean (see section 1.4.2)
  Parameter (default true)
  Set this flag to add a 'cd' command to the current working directory. This is useful under MS Windows when dragging the file to a gnuplot icon.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *missing_value*: number [<**user**>]
  Parameter
  Replace misisng values with this.

- *command_file*: string (see section 1.4.5)
  Parameter (default 'daisy.gnuplot')
  File name for gnuplot commands.

- *x_min*: number [<**user**>]
  Parameter
  Min value of x-axes.

- *x_max*: number [<**user**>]
  Parameter
  Max value of x-axes.

- *y_min*: number [<**user**>]
  Parameter
  Min value of y-axes.

- *y_max*: number [<**user**>]
  Parameter
  Max value of y-axes.

- *z_min*: number [<**user**>]
  Parameter
  Min value of z-axes.

- *z_max*: number [<**user**>]
  Parameter
  Max value of z-axes.

- *x_tics*: submodel (see section 1.4.7) sequence
  List of tic markers for the x-axes.

  < *value name* >

  - *value*: number (dimension not specified)
    Parameter
    Tic position.
  - *name*: string (see section 1.4.5)
    Parameter
    Tic name.

- *x_tic_max*: number [<**user**>]
  Parameter
  Highest synthetic z value for at x tics.

- *output_base*: string (see section 1.4.5)
  Parameter
  Prefix used for all output files.

- *output_width*: integer
  Parameter
  Number of digits in output file name.

- *tags*: submodel (see section 1.4.7) sequence
  List of column tags for the data file.

  < *x y tag* >

  - *x*: number (dimension not specified)
    Parameter
    X position.
  - *y*: number (dimension not specified)
    Parameter
    Y position.
  - *tag*: string (see section 1.4.5)
    Parameter
    Column name.

52.12 Daisy

The Daisy crop/soil/atmosphere model. See also [Hansen et al., 1990, Hansen, 2002, Hansen et al., 1991]

- *weather*: **weather** component (see chapter 82)
  Optional component
  Weather model for providing climate information during the simulation. Can be overwritten by column specific weather.

- *harvest*: **Harvest** fixed component (see section 85.4) sequence
  Submodel (default: an empty sequence)
  Total list of all crop yields.

- *time*: **Time** fixed component (see section 85.21)
  Submodel (has partially specified default value)
  Current time in the simulation.

- *timestep*: **Timestep** fixed component (see section 85.1)
  Optional submodel
  Length of large timestep in simulation. The default value is 1 hour, anything else is unlikely to work.

- *stop*: **Time** fixed component (see section 85.21)
  Optional submodel
  Latest time where the simulation stops. By default, the simulation will run until the manager request it to stop.

- *column*: **column** component (see chapter 19) sequence
  List of columns to use in this simulation.

- *exchange*: **scope** component (see chapter 58) sequence
  Component (default: an empty sequence)
  List of exchange items for communicating with external models.

- *scope*: **scopesel** component (see chapter 59)
  Component (default 'null')
  Scope to evaluate expessions in.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (has default value with length 3)

  ```
  (cite "daisy-def" "daisy-new" "daisy-fertilizer")
  ```

  Parameter description:
  BibTeX keys that would be relevant for this model or paramterization.

- *output*: **log** component (see chapter 39) sequence
  List of logs for output during the simulation.

- *activate_output*: **condition** component (see chapter 21)
  Component (default 'true')
  Activate output logs when this condition is true. You can use the 'after' condition to avoid logging during an initialization period.

- *log_time_columns*: string (see section 1.4.5) sequence
  Parameter (has default value with length 4)

```
(log_time_columns year month mday hour)
```

Parameter description:
List of default time components to include in log files. Choose between: 'year': Year 'month': Month 'week': Week number (first Thursday is in week 1) 'yday': Julian day 'mday': Day in month 'wday': Weekday (Sunday = 7) 'hour': Hour 'minute': Minute 'second': Second 'microsecond': Microsecond

- *log_prefix*: string (see section 1.4.5)
  Parameter (default '')
  Prefix for log file names. Set it to 'log/' to put all files in a subdir.
- *print_time*: **condition** component (see chapter 21)
  Component (default 'periodic')
  Print simulation time whenever this condition is true. The simulation time will also be printed whenever there are any news to report, like emergence of crop or various management operations. Good values for this parameter would be hourly, daily or monthly.
- *manager*: **action** component (see chapter 9)
  Specify the management operations to perform during the simulation.
- *previous*: **Time** fixed component (see section 85.21)
  Optional submodel
  Previous time in the simulation.
- *next_large*: **Time** fixed component (see section 85.21)
  Optional submodel
  End of next large timestep.
- *minimal_timestep*: **Timestep** fixed component (see section 85.1)
  Optional submodel
  Minimum length of timestep in simulation. By default, this is the same as 'timestep'.

Log Variables

- *dt*: number [**h**]
  Current timestep used by simulation.

52.13 AM_table

Generate a table of fertilizers.

52.14 wse

Generate a table of the water stress effect.

- *wse*: **wse** component (see chapter 83)
  The water stress effect to show in the table.
- *intervals*: integer
  Parameter (default 10)
  Number of intervals in the table.

Chapter 53

raddist

The 'raddist' component calculates the radiation distribution in the canopy.

53.1 default

Default model of radiation distribution in the canopy.
Used by bioclimate default raddist (see 13.1, page 101) .

53.2 sun-shade

Sun-shade model of radiation distribution in the canopy.

- *sigma_PAR*: number (dimensionless)
  Parameter (default 0.15)
  Leaf scattering coefficient of PAR. sigma_PAR=0,15 (Houborg, 2006)
- *sigma_NIR*: number (dimensionless)
  Parameter (default 0.83)
  Leaf scattering coefficient of NIR. sigma_NIR=0,83 (Houborg, 2006)
- *Ps_PAR*: number (dimensionless)
  Parameter (default 0.1)
  Soil reflection coefficient of PAR, Ps_PAR = 0.1 (Houborg, 2006)
- *Ps_NIR*: number (dimensionless)
  Parameter (default 0.18)
  Soil reflection coefficient of NIR, Ps_NIR = 0.18 (Houborg, 2006)

Log Variables

- *IRb0*: number [$\mathbf{W\ m^{-2}}$]
  Beam radiation above the canopy
- *IRd0*: number [$\mathbf{W\ m^{-2}}$]
  Diffuse radiation above the canopy
- *Ph_PAR*: number []
  Canopy reflection coefficeint of beam PAR for horizontal leaves
- *Pcb_PAR*: number []
  Canopy reflection coefficeint of beam PAR for uniform leaf-angel distribution

- *Pscb_PAR*: number []
  Canopy-soil reflection coefficeint of beam PAR for uniform leaf-angel distribution

- *Pscd_PAR*: number []
  Canopy-soil reflection coefficeint of diffuse PAR for uniform leaf-angel distribution

- *Ph_NIR*: number []
  Canopy reflection coefficeint of beam NIR for horizontal leaves

- *Pcb_NIR*: number []
  Canopy reflection coefficeint of beam NIR for uniform leaf-angel distribution

- *Pscb_NIR*: number []
  Canopy-soil reflection coefficeint of beam NIR for uniform leaf-angel distribution

- *Pscd_NIR*: number []
  Canopy-soil reflection coefficeint of diffuse NIR for uniform leaf-angel distribution

Chapter 54

rainergy

Energy in rain.
< `component` `(description` *`description`*`)` `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

54.1 EUROSEM

Kinetic energy model taking vegetation into account. See also [Morgan et al., 1998]
Used by reaction colgen_Morgan98 rainergy (see 55.10, page 287) .

54.2 Brown87

Energy as a semi-empirical function of rain intensity. The energy content in the fraction that hits the canopy is ignored. See also [Brown and Foster, 1987]
Used by reaction colgen_Jarvis99 rainergy (see 55.8, page 285) .

Chapter 55

reaction

Generic transformations between soil chemicals.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

55.1 default

Transformation between two soil chemicals.

- *transform*: **transform** component (see chapter 72)
  Tranformation process between 'A' to 'B'.

- *A*: string (see section 1.4.5)
  Parameter
  Name of first soil component in equilibrium.

- *B*: string (see section 1.4.5)
  Parameter
  Name of second soil component in equilibrium.

Log Variables

- *S_AB*: number [$\mathbf{g/cm^3/h}$] soil cells
  Converted from A to B this timestep (may be negative).

55.2 adsorption

Maintain equilibrium between solute and sorbed from.

- *solute*: string (see section 1.4.5)
  Parameter
  Name of solute form of chemical.

- *equilibrium*: **adsorption** component (see chapter 10)
  Function for calculating equilibrium between solute and sorbed form.

- *sorbed*: string (see section 1.4.5)
  Parameter
  Name of sorbed form of chemical.

- *adsorption_rate*: **number** component (see chapter 45)
  Tranformation rate from solute to sorbed form.

- *desorption_rate*: **number** component (see chapter 45)
  Optional component
  Tranformation rate from sorbed to solute form. By default, this is identical to 'adsorption_rate'.

Log Variables

- *adsorption_source*: number [$\mathbf{g/cm^3/h}$] soil cells
  Converted from solute to sorbed form this timestep (may be negative).

55.3 nitrification

Nitrification. The actual nitrification specification is part of the horizon models, this reaction just applies the models and logs the result.

Used by chemistry N reaction (see 18.2, page 125) .

Log Variables

- *NO3*: number [$\mathbf{g/cm^3/h}$] soil cells
  Amount of nitrate generated this hour.

- *NH4*: number [$\mathbf{g/cm^3/h}$] soil cells
  Amount of ammonium consumed this hour.

- *N2O*: number [$\mathbf{g/cm^3/h}$] soil cells
  Amount of nitrous oxide generated this hour.

55.4 equilibrium

Equilibrium between two soil chemicals.

- *equilibrium*: **equilibrium** component (see chapter 29)
  Function for calculating equilibrium between A and B.

- *A*: string (see section 1.4.5)
  Parameter
  Name of first soil component in equilibrium.

- *secondary*: boolean (see section 1.4.2)
  Parameter (default false)
  Equilibrium should happen in the secondary domain. There will only be a reaction when there is water in the secondary domain (inter-aggregate pores), at both the beginning and end of the timestep. By default, only the content of the primary domain (soil-bound and intra-aggregate pores), will be included in the reaction. There is no way to use this model to specify an equilibrium reaction in the tertiary domain (biopores).

- *primary*: boolean (see section 1.4.2)
  Optional parameter
  Equilibrium should happen in the primary domain. If true, the content of the primary soil domain (soil-bound and intra-aggregate pores), will be included in the reaction. By default, this will be true if 'secondary' is false, and be false if 'secondary' is true.

- *surface*: boolean (see section 1.4.2)
  Parameter (default false)
  Equilibrium should happen in the surface.

- *B*: string (see section 1.4.5)
  Parameter
  Name of second soil component in equilibrium.

- *k_AB*: **number** component (see chapter 45)
  Tranformation rate from soil component 'A' to 'B'.

- *k_BA*: **number** component (see chapter 45)
  Optional component
  Tranformation rate from soil component 'B' to 'A'. By default, this is identical to 'k_AB'.

- *colloid*: string (see section 1.4.5)
  Optional parameter
  Let 'rho_b' denote content of specified chemical. This might affect the evaluation of the 'k_AB' and 'k_BA' parameter expressions, as well as the 'equilibrium' model. By default, 'rho_b' will be the soil dry bulk density.

Log Variables

- *S_AB*: number [$\mathbf{g/cm^3/h}$] soil cells
  Converted from A to B in soil this timestep (may be negative).

- *surface_AB*: number [$\mathbf{g/cm^2/h}$]
  Converted from A to B on surface this timestep (may be negative).

55.5 denitrification

Denitrification in soil, (conversion of nitrate to atmospheric nitrogen).

In this model, it is made proportional to the CO2 development, as specified by the parameter alpha, with a maximum rate specified by the parameter 'K'. The denitrification is also affected by temperature and water pressure. Additional denitrification from CO2 produced from fast OM pools can be triggered by setting alpha_fast or water_factor_fast different. This additional denitrification is limited by K_fast.

Used by chemistry N reaction (see 18.2, page 125) .

- *K*: number [$\mathbf{h^{-1}}$]
  Parameter (default 0.020833)
  Maximum fraction of nitrate converted at each time step from slow pools.

- *alpha*: number [**(g NO3-N/h)/(g CO2-C/h)**]
  Parameter (default 0.1)
  Anaerobic denitrification constant for slow pools.

- *heat_factor*: plf [**dg C** → <**none**>]
  Optional parameter
  Heat factor. By default, use a build in function valid for temperate climates.

- *water_factor*: plf [<**fraction**> → <**none**>]
  Optional parameter (has default value with 2 points)

  ```
  (water_factor (0.7 0) (1 1))
  ```

  Parameter description:
  Water potential factor for slow pools. This is a function of the current water content as a fraction of the maximal water content.

- *K_fast*: number [$\mathbf{h}^{-1}$]
  Optional parameter
  Maximum fraction of nitrate converted at each time step from fast pools. By default this is identical to 'K'.

- *alpha_fast*: number [**(g NO3-N/h)/(g CO2-C/h)**]
  Optional parameter
  Anaerobic denitrification constant for fast pools. This applies to the CO2 produced from turnover of fast OM pools. By default, this is identical to alpha.

- *water_factor_fast*: plf [<**fraction**> → <**none**>]
  Optional parameter
  Water potential factor for fast pools By default, this is identical to the 'water_factor' parameter.

- *redox_height*: number [**cm**]
  Optional parameter
  Height (a negative number) blow which redox processes start. All NO3 below this height will be denitrified immediately. By default no redox denitrification occurs.

Log Variables

- *potential*: number [$\mathbf{g/cm^3/h}$] soil cells
  Potential amount of denitrification at anarobic conditions.

- *converted*: number [$\mathbf{g/cm^3/h}$] sequence
  Amount of denitrification.

- *converted_fast*: number [$\mathbf{g/cm^3/h}$] soil cells
  Additional denitrification due to turnover in fast pools.

- *converted_redox*: number [$\mathbf{g/cm^3/h}$] soil cells
  Additional denitrification due to chemical redox processes.

- *potential_fast*: number [$\mathbf{g/cm^3/h}$] soil cells
  Additional potential due to turnover in fast pools.

55.6 filter_velocity

Filtration of soil colloids.

- *immobile*: string (see section 1.4.5)
  Optional parameter
  Immobile colloids in the soil. By default, filtered colloids are not tracked.

- *mobile*: string (see section 1.4.5)
  Parameter
  Mobile colloids dissolved in soil water.

- *fc_primary*: number [$\mathbf{cm}^{-1}$]
  Parameter
  Filter coefficient in the primary domain

- *fc_secondary*: number [$\mathbf{cm}^{-1}$]
  Parameter
  Filter coefficient in secondary domain

Log Variables

- *F_primary*: number [$\mathbf{g/cm^3/h}$] soil cells
  Filtration in the primary domain (intra-aggregate pores).

- *F_secondary*: number [$\mathbf{g/cm^3/h}$] soil cells
  Filtration in secondary domain (inter-aggregate pores).

55.7 colgen

Shared parameter and log variable for colloid generation models.

- *ponddamp*: **ponddamp** component (see chapter 51)
  Model for calculating 'KH'.

- *colloid*: string (see section 1.4.5)
  Parameter
  Colloid to generate.

Log Variables

- *surface_release*: number [<**fraction**>]
  Fraction of available soil particles released as colloids this timestep.

- *dds*: number [**mm**]
  Median raindrop size.

- *KH*: number [<**fraction**>]
  Ponding factor.

- *D*: number [$\mathbf{g/cm^2/h}$]
  Depletion of detachable particles from top soil.

55.8 colgen_Jarvis99

A 'colgen' model (see 55.7, page 285) build into Daisy.

Colloid generation emulating the MACRO model. See also [Jarvis et al., 1999, Brubaker et al., 1992]

- *rainergy*: **rainergy** component (see chapter 54)
  Component (default 'Brown87')
  Model for calculating energy in rain.

- *Ms*: number [**g/g**]
  Optional state variable
  Current concentration of detachable particles in top soil. By default, 10% of Mmax.

- *tillage_replenish_all*: boolean (see section 1.4.2)
  Parameter (default false)
  Set Ms = Mmax after tillage.

- *Mmax*: number [**g/g**]
  Optional parameter
  Maximum amount of detachable particles. By default, method 1 of Brubaker et al, 1992, will be used.

- *Mmax_tillage_factor*: plf [**d** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (Mmax_tillage_factor (0 1) (1 1))
  ```

  Parameter description:
  Factor to modify Mmax with as a fuction of days after tillage.

- *kd*: number [**g/J**]
  Parameter
  Detachment rate coefficient.

- *kr*: number [**g/cm^2/h**]
  Parameter
  Replenishment rate coefficient.

- *zi*: number [**cm**]
  Optional parameter
  Thickness of surface soil layer. By default, the value of 'z_mixing' from 'Surface' is used.

Log Variables

- *As*: number [**g/cm^2**]
  Current amount of detachable particles in top soil.

- *P*: number [**g/cm^2/h**]
  Replenishment of detachable particles to top soil.

- *KE*: number [**J/cm^2/h**]
  Kinertic energy avalable for colloid generation.

- *E*: number [**J/cm^2/mm**]
  Kinetic energy in rain.

55.9 colgen_Styczen88

A 'colgen' model (see 55.7, page 285) build into Daisy.
Colloid generation using rainfall momentum. See also [Styczen and Høeg-Schmidt, 1988]

- *Ae*: number [$\mathbf{h^2/g/cm^2}$]
  Parameter
  Soil resistance factor.
- *MA*: number [<**fraction**>]
  Parameter
  Protective cover (mulch factor).
- *droplet_diameter*: number [**mm**]
  Parameter
  Size of droplets from vegetation.

Log Variables

- *DH*: number [$\mathbf{kg^2/m/s^2}$]
  Squared vegetation droplet momentum.
- *CM*: number [<**fraction**>]
  Vegetation factor.
- *MR*: number [$\mathbf{(N\ s)^2/m^2/s}$]
  Squared direct rainfall momentum.

55.10 colgen_Morgan98

A 'colgen' model (see 55.7, page 285) build into Daisy.
Colloid generation using kinetic energy, emulating EUROSEM. See also [Morgan et al., 1998]

- *rainergy*: **rainergy** component (see chapter 54)
  Component (default 'EUROSEM')
  Model for calculating energy in rain.
- *kd*: number [**g/J**]
  Parameter
  Detachment rate coefficient. The EUROSEM user manual list values between 0.8 and 6.0 [g/J] for various soils in Table A9.1.

Log Variables

- *KE*: number [$\mathbf{J/cm^2/h}$]
  Kinertic energy avalable for colloid generation.
- *E*: number [$\mathbf{J/cm^2/mm}$]
  Kinetic energy in rain.

55.11 bound_release

Release of chemicals bound to colloids from surface soil.

This follows the generation of colloids on the surface. The colloid generation model should already have calculated the amount of released colloids as either a fraction of the readily available colloids (Jarvis99), or as a fraction of the total

amount of soil in the mixing layer (Styczen88, Morgan98). The same fraction of the immobile chemical on the surface is released in the colloid bound form.

This reaction must be listed after the colloid generation reaction in the setup file.

- *bound*: string (see section 1.4.5)
  Optional parameter
  Chemical bound to colloids. If unspecified, the colloid bound form will not be traced.

- *colloid*: string (see section 1.4.5)
  Optional parameter (default 'colloid')
  Name of colloid whose release we mimic.

- *immobile*: string (see section 1.4.5)
  Parameter
  Immobile (or mixed form) chemical in the soil surface.

Log Variables

- *release*: number [$\mathbf{g/cm^2/h}$]
  Release rate of immobile chemical as colloids.

55.12 sorption

Kinetic linear sorption equilibrium. Faster than the 'equilibrium' reaction model, more flexible than the 'adsorption' reaction model.

- *solute*: string (see section 1.4.5)
  Parameter
  Name of solute form of chemical.

- *K_clay*: number [$\mathbf{cm^3/g}$]
  Optional parameter
  Clay dependent distribution parameter. It is multiplied with the soil clay fraction to get the clay part of the 'K_d' factor. If 'K_OC' is specified, 'K_clay' defaults to 0.

- *K_OC*: number [$\mathbf{cm^3/g}$]
  Optional parameter
  Humus dependent distribution parameter. It is multiplied with the soil organic carbon fraction to get the carbon part of the 'K_d' factor. By default, 'K_OC' is equal to 'K_clay'.

- *sorbed*: string (see section 1.4.5)
  Parameter
  Name of sorbed form of chemical.

- *colloid*: string (see section 1.4.5)
  Optional parameter
  Sorp to this chemical instead of to the soil matrix.

- *K_d*: number [$\mathbf{cm^3/g}$]
  Optional parameter
  Equillibrium parameter: M = C (K_d rho_b + Theta) Here M is the total amount, C is solute concentration, K_d is this parameter, rho_b is the dry bulk density, and Theta is the volumetric water content. By default, K_d is calculated from K_clay and K_OC.

- *k_sorption*: number [$\mathbf{h}^{-1}$]
  Parameter
  Sorption rate.

- *k_desorption*: number [$\mathbf{h}^{-1}$]
  Optional parameter
  Desorption rate. By default, this is identical to 'k_sorption'.

- *soil_enrichment_factor*: number (dimensionless)
  Parameter (default 1)
  Multiply K_d with this number if 'colloid' is set. This represents how much more accesible colloids is compared to the soil matrix.

Log Variables

- *surface_sorption*: number [$\mathbf{g/cm^2/h}$]
  Sorption on surface this timestep (may be negative).

- *S_sorption*: number [$\mathbf{g/cm^3/h}$] soil cells
  Sorption in soil this timestep (may be negative).

- *S_sorption_primary*: number [$\mathbf{g/cm^3/h}$] soil cells
  Sorption in primary domain this timestep (may be negative).

- *S_sorption_secondary*: number [$\mathbf{g/cm^3/h}$] soil cells
  Sorption in secondary domain this timestep (may be negative).

Chapter 56

rootdens

Root density calculations.
Used by RootSystem @ rootdens (see 85.13, page 423) .

```
< component   (description description)
              (cite)
              (SpRtLength 100 [m/g]) >
```

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *SpRtLength*: number [**m/g**]
  Parameter (default 100)
  Specific root length

56.1 Gerwitz+Page74

Use exponential function for root density. See also [Gerwitz and Page, 1974]

- *DensRtTip*: number [$\mathbf{cm/cm^3}$]
  Parameter (default 0.1)
  Root density at (potential) penetration depth.

- *MinDens*: number [$\mathbf{cm/cm^3}$]
  Parameter (default 0)
  Minimal root density Root density will never be below this, as long as there is enough root mass. Extra root mass will be distributed according to Gerwitz and Page. If there are too little root mass, the root will have the same density all the way down.

Log Variables

- *a*: number [$\mathbf{cm^{-1}}$]
  Form parameter. Calculated from 'DensRtTip'.

- *L0*: number [$\mathbf{cm/cm^3}$]
  Root density at soil surface.

56.2 DS_Depth

Specify root density as a function of development stage.

- *entries*: submodel (see section 1.4.7) sequence
  A list of pairs, where the first element of each pair is a development stage (usually a number between 0 (emergence) and 2 (ripe), and the second element is a PLF specifying the relative root density as a function of soil depth in cm (a positive number).

  To find the absolute root density, Daisy will interpolate the relative root density distribution specified for the entries before and after the current development stage, and scale them to match the current total root mass.

  < *index density* >

  - *index*: number (dimension not specified)
    Parameter
    Index for specifying root density.
  - *density*: plf [<**unknown**> → <**none**>]
    Parameter
    Relative root density as a function of root depth .

56.3 DS_Rel

Specify root density as a function of development stage.

- *entries*: submodel (see section 1.4.7) sequence
  A list of pairs, where the first element of each pair is a development stage (usually a number between 0 (emergence) and 2 (ripe), and the second element is a PLF specifying the relative root density as a function of soil depth relative to the total root depth.

  To find the absolute root density, Daisy will interpolate the relative root density distribution specified for the entries before and after the current development stage, and scale them to match the current total root mass.

  < *index density* >

  - *index*: number (dimension not specified)
    Parameter
    Index for specifying root density.
  - *density*: plf [<**unknown**> → <**none**>]
    Parameter
    Relative root density as a function of root depth .

56.4 Depth_Depth

Specify root density as a function of development stage.

- *entries*: submodel (see section 1.4.7) sequence
  A list of pairs, where the first element of each pair is the root depth, (a positive number), and the second element is a PLF specifying the relative root density as a function of soil depth in cm (a positive number).

  To find the absolute root density, Daisy will interpolate the relative root density distribution specified for the entries before and after the current development stage, and scale them to match the current total root mass.

< *index density* >

– *index*: number (dimension not specified)
Parameter
Index for specifying root density.

– *density*: plf [<**unknown**> → <**none**>]
Parameter
Relative root density as a function of root depth .

56.5 Anders Pedersen

Use exponential function for root density. In this variant of Gerwitz and Page, 'a' is specified as a function of development stage. See also [Gerwitz and Page, 1974]

- *q*: number [**cm**]
Parameter
Extra root length below max rooting depth. Root density will decrease linearly from the GP calculated amount at max rooting depth to zero 'q' further down.

- *a_DS*: plf [**DS** → $\mathbf{cm}^{-1}$]
Parameter
Form parameter as a function of development stage.

Log Variables

- *a*: number [$\mathbf{cm}^{-1}$]
Form parameter. Calculated from 'a_DS'.

- *L0*: number [$\mathbf{cm/cm}^3$]
Root density at soil surface.

56.6 GP1D

Use exponential function for root density.

See Gerwitz, S. and E.R. Page (1974): An empirical mathematical model to describe plant root systems. J. Appl. Ecol. 11, 773-781.

- *DensRtTip*: number [$\mathbf{cm/cm}^3$]
Parameter (default 0.1)
Root density at (potential) penetration depth.

- *DensIgnore*: number [$\mathbf{cm/cm}^3$]
Optional parameter
Ignore cells with less than this root density. By default, this is the same as DensRtTip.

Log Variables

- *a*: number [$\mathbf{cm}^{-1}$]
Form parameter. Calculated from 'DensRtTip'.

- *k*: number (dimensionless)
Scale factor due to soil limit.

  Some roots might be below the soil imposed maximum root depth, or in areas with a density lower than the limit specified by DensIgnore. These roots will

be re distributed within the root zone by multiplying the density with this scale factor.

- *L0*: number [**cm/cm**^{3}]
  Root density at soil surface.

56.7 GP2D

Use exponential function for root density in row crops.

This is a two dimension model (z, x), where the z-axis is vertical, and the x-axis is horizontal and ortogonal to the row. The row is assumed to be uniform (dense), allowing us to ignore that dimension.

We assume the root density decrease with horizontal distance to row, as well as depth below row. See also [Gerwitz and Page, 1974]

- *row_position*: number [**cm**]
  State variable (default 0)
  Horizontal position of row crops.

- *debug*: integer
  Parameter (default 0)
  Add debug messages if larger than 0.

- *DensRtTip*: number [**cm/cm**^{3}]
  Parameter (default 0.1)
  Root density at (potential) penetration depth.

- *DensIgnore*: number [**cm/cm**^{3}]
  Optional parameter
  Ignore cells with less than this root density. By default, this is the same as DensRtTip.

- *row_distance*: number [**cm**]
  State variable
  Distance between rows of crops.

Log Variables

- k: number (dimensionless)
  Scale factor due to soil limit.

  Some roots might be below the soil imposed maximum root depth, or in areas with a density lower than the limit specified by 'DensIgnore'. These roots will be re distributed within the root zone by multiplying the density with this scale factor.

- a_z: number [**cm**$^{-1}$]
  Form parameter. Calculated from 'DensRtTip'.

- a_x: number [**cm**$^{-1}$]
  Form parameter. Calculated from 'DensRtTip'.

- *L00*: number [**cm/cm**^{3}]
  Root density at row crop at soil surface.

Chapter 57

rubiscoNdist

The 'rubiscoNdist' component calculates the rubisco N distribution for photosynthesis in the canopy.

57.1 exp

Boegh et al.(2002) rubisco N-distribution model in the canopy for photosynthesis.
Used by photosynthesis Farquhar N-dist (see 50.2, page 260) .

- *kn*: number (dimensionless)
  Parameter (default 0.713)
  Extinction coefficient of nitrogen in the canopy, kn = 0.713 (De Pury &Farquhar, 1997)

- *f_photo*: number (dimensionless)
  Parameter (default 1)
  Fraction of photosynthetically active N in canopy. According to (Boegh et al., 2002) f_photo = 0.75. However, non-functional N is already substracted from leaf-N in the cropN_std module, therefore f_photo = 1.0 as default.

57.2 uniform

Uniform rubisco N-distribution model in the canopy for photosynthesis.

- *f_photo*: number (dimensionless)
  Parameter (default 1)
  Fraction of photosynthetically active N in canopy, f_photo = 0.75 (Boegh et al., 2002). However, non-functional N is already substracted from leaf-N in the cropN_std module, therefore f_photo = 1.0 as default.

57.3 expr

expr rubisco N-distribution model in the canopy.

- *value*: **number** component (see chapter 45)
  Expression that evaluates to the relative rubisco N intesity where 1 is the value in top of the canopy.

- *f_photo*: number (dimensionless)
  Parameter (default 1)

Fraction of photosynthetically active N in canopy. According to (Boegh et al., 2002) f_photo = 0.75. However, non-functional N is already substracted from leaf-N in the cropN_std module, therefore f_photo = 1.0 as default.

57.4 forced

Forced rubisco capacity distribution model in the canopy.

- *value*: **number** component (see chapter 45)
  Expression that evaluates to the relative rubisco capacity where 1 is the value in top of the canopy.

Chapter 58

scope

A scope maps names to values.

58.1 exchange

Exchange values with an external model.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.
- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.
- *entries*: **exchange** component (see chapter 30) sequence
  List of items to exchange.

Chapter 59

scopesel

A method to choose a scope in a Daisy simulation.

59.1 name

Select named scope.

- *frame*: string (see section 1.4.5)
  Parameter
  Name of scope to select.

59.2 null

Select the empty scope.

Used by action extern_fertigation scope (see 9.52, page 85) , column default scope (see 19.1, page 130) , and program Daisy scope (see 52.12, page 273) .

Chapter 60

secondary

Specify secondary domain.

The secondary domain consist typically of soil fractures or other inter-aggregate pores small enough to be dominated by capillarity, yet so large that water moves fast enough that the solute equilibrium with the primary domain (typically intra-aggregate pores) can not be maintained.

This allows a pulse of water to be move through saturated or near saturated soil without solutes in the new water being mixed with solutes in the old water. The effects are twofold: It allows solutes applied to the surface to reach deeper soil layers much faster than it would otherwise, and it protects solutes in the soil matrix from being washed out with fast moving new water.

Used by horizon component secondary_domain (see 35, page 183) .

60.1 none

No secondary domain.

There is always full equilibrium between solute in different size matrix pores.

Used by horizon component secondary_domain (see 35, page 183) .

60.2 alpha

Shared base class for non-empty secondary domains.

- *alpha*: number [$\mathbf{h}^{-1}$]
  Parameter
  Exchange rate between primary and secondary water.

60.3 pressure

A 'alpha' model (see 60.2, page 301) build into Daisy.

Horizon has secondary domain specifyed by pressure thresshold.

The secondary domain consist of water in matrix pores larger than what corresponds to the specified pressure.

- K: number [**cm/h**]
  Parameter (default 0)
  Water conductivity when secondary domain is active. If the secondary domain is already included in the normal conductivity curve, specify 0.0 to use that value instead.

- *h_lim*: number [**cm**]
  Parameter
  Minimal pressure needed for activating secondary domain.

60.4 cracks

A 'alpha' model (see 60.2, page 301) build into Daisy.
Secondary domain specified by aperture and density of soil cracks.

- *density*: number [$\mathbf{m}^{-1}$]
  Parameter
  Density of cracks.

- *aperture*: number [**m**]
  Parameter
  Average distance between walls in cracks.

- *use_secondary*: boolean (see section 1.4.2)
  Parameter (default true)
  Divide soil matrix into two domains for solute transport. Set this to false to make cracks affect only the conductivity curve.

Chapter 61

seed

Initial growth after emergence.

The initial growth process governs the growth of the crop until the point where there is enough leaf area for photosynthesis to take over.

61.1 LAI

Initial crop growth is governed by a forced LAI function.

Used by crop default Seed (see 22.1, page 143) , crop Maize Seed (see 22.2, page 145) , crop Pioneer Maize Seed (see 22.5, page 145) , crop Ikuwala Maize Seed (see 22.3, page 145) , crop Potato; Koege Seed (see 22.8, page 145) , crop Potato; FertOrgaNic Seed (see 22.10, page 145) , crop Potato; Agria Seed (see 22.11, page 146) , crop Potato; Folva Seed (see 22.12, page 146) , crop Potato; Triada Seed (see 22.14, page 146) , crop Sugar Beet Seed (see 22.18, page 146) , crop Fodder Beet Seed (see 22.19, page 146) , crop Winter Rape Seed (see 22.25, page 147) , crop Grass to grain Seed (see 22.29, page 147) , crop Ryegrass Seed (see 22.31, page 147) , crop Wclover Seed (see 22.32, page 148) , crop Beetroot Seed (see 22.33, page 148) , crop Broccoli Seed (see 22.34, page 148) , crop Brussels sprouts Seed (see 22.36, page 148) , crop Celeriac Seed (see 22.38, page 148) , crop White cabbage Seed (see 22.42, page 149) , crop Early white cabbage - transplanted Seed (see 22.44, page 149) , crop Onion Seed (see 22.45, page 149) , crop Vaarbyg Seed (see 22.48, page 149) , crop Froegraes Seed (see 22.51, page 150) , crop Silomajs Seed (see 22.53, page 150) , and crop Vinterraps Seed (see 22.55, page 150) .

- *DSLAI05*: number (dimensionless)
  Parameter (default 0.15)
  DS at CAI=0.5; initial phase.

- *SpLAIfac*: plf [**DS** → <**none**>]
  Parameter (has default value with 4 points)

  ```
  (SpLAIfac (0 3) (0.2 1.5) (0.4 1.25) (0.6 1))
  ```

  Parameter description:
  Factor defining maximum specific leaf weight. Only used during the initial phase.

- *InitCAI*: boolean (see section 1.4.2)
  State variable (default true)
  Initial CAI development phase.

61.2 release

Initial crop growth is governed by carbon released from seeds.

- *C*: number [**g C/m**^{2}]
  Optional state variable
  Unreleased carbon left in seeds.

- *initial_weight*: number [**g w.w./m**^{2}]
  Optional parameter
  Initial seed weight to use when not specified by the sow operation. If not specified here, specifying seed amount when sowing is mandatory.

- *DM_fraction*: number [<**fraction**>]
  Parameter
  Dry matter content in seeds.

- *C_fraction*: number [<**fraction**>]
  Parameter
  Carbon content in seeds.

- *N_fraction*: number [<**fraction**>]
  Parameter
  Nitrogen content in seeds.

- *rate*: number [**h**$^{-1}$]
  Parameter
  Release rate of seed carbon to assimilate pool.

Chapter 62

select

Select part of state.

```
< component   (multi sum)
              (flux flux)
              (expr expr)
              (dimension dimension)
              (factor 1 [?])
              (offset 0 [?])
              (description description)
              (cite)
              (when when)
              (documentation documentation)
              (tag tag)
              (path path ...)
              (spec spec)
              (handle handle)
              (interesting_content true)
              (negate false)
              (accumulate false) >
```

- *multi*: string (see section 1.4.5)
  Optional parameter (default 'sum')
  This option determine how to handle mutiple matches within a timestep. This could be two crops on the same column, or one crop on two columns.

  min: Use smallest value

  max: Use largest value

  sum: Use the sum of all matches, weighted by relative column area if the matches are from different columns.

- *flux*: boolean (see section 1.4.2)
  Optional parameter
  OBSOLETE. This value will be used if 'handle' is not specified.A value of true then means 'sum', and false means 'current'.

- *expr*: **number** component (see chapter 45)
  Optional component
  Expression for findig the value for the log file, given the internal value 'x'. For example '(expr (ln x))' will give you the natural logarithm of the value.

- *dimension*: string (see section 1.4.5)
  Optional parameter

The unit for numbers in this column. These will be printed in the second line of the log file. The character '&' will be replaced with the log timestep. If you do not specify the dimension explicitly, a value will be interfered from 'spec' if available.

- *factor*: number (dimension not specified)
  Parameter (default 1)
  Factor to multiply the calculated value with, before logging. OBSOLETE: Use 'expr' instead.

- *offset*: number (dimension not specified)
  Parameter (default 0)
  Offset to add to the calculated value, before logging. OBSOLETE: Use 'expr' instead.

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *when*: **condition** component (see chapter 21)
  Optional component
  OBSOLETE. If you set this variable, 'flux' will be set to true. This overwrites any direct setting of 'flux'.

- *documentation*: string (see section 1.4.5)
  Optional parameter
  Documentation for this entry.

- *tag*: string (see section 1.4.5)
  Optional parameter
  Tag to identify the column. These will be printed in the first line of the log file. The default tag is the last element in the path.

- *path*: string (see section 1.4.5) sequence
  Parameter
  Sequence of attribute names leading to the variable you want to log in this column. The first name should be one of the attributes of the daisy component itself. What to specify as the next name depends on the type of the attribute you selected before.

  If the value of that attribute itself is a fixed component, you should specify the name of an attribute in that component as the second name.

  If the value is a library component, you should specify the name of the model or parameterization you are interested in, and then the name of the attribute inside the model you want to log.

  The last attribute in the patch should be a number, a number sequence, a string, or an integer. These are the only values which can be logged by this model.

  You can use the special value "*" to match everything at a given level, for example all crops. This way the path can specify multiple values, they will be added before they are printed in the log file. All values that start with a "$"

will work like "*". They are intended to be mapped with the 'set' attribute in the 'table' log model.

- *spec*: submodel (see section 1.4.7)
  Optional submodel
  Specification for the attribute to be logged of the form

  library model submodel* attribute

  Unlike path, the attribute may occur several different places in the simulation, if the model is used at several places. Also, there is no wildcards, so only a single model can be matches. The spec is used for helping Daisy establish a unique dimension and description for the attribute.

  < *library model submodels_and_attribute...* >

  - *library*: string (see section 1.4.5)
    Parameter
    Name of library where the attribute belong. Use 'fixed' to denote a fixed component.
  - *model*: string (see section 1.4.5)
    Parameter
    Name of model or fixed component where the attribute belongs.
  - *submodels_and_attribute*: string (see section 1.4.5) sequence
    Parameter
    Name of submodels and attribute.

- *handle*: string (see section 1.4.5)
  Optional parameter
  This option determine how the specified variable should be logged.

  min: Log the smallest value seen since last time the variable was logged. If 'accumulate' is true, use the smallest value ever.

  max: Log the largest value seen since last time the variable was logged. If 'accumulate' is true, use the largest value ever.

  average: Log the arithmetic average value seen since last time the variable was logged. If 'accumulate' is true, use the average of all values.

  sum: Accumulate value since last time the variable was logged. If 'accumulate' is true, accumulate since the start of the log.

  current: Log the current value for the variable. If 'accumulate' is true, the printed values will be accumulated.

- *interesting_content*: boolean (see section 1.4.2)
  Parameter (default true)
  True if the content of this column is interesting enough to warrent an initial line in the log file. This only affects non-flux variables.

- *negate*: boolean (see section 1.4.2)
  Parameter (default false)
  Switch sign of value. I.e. upward fluxes become downward fluxes.

- *accumulate*: boolean (see section 1.4.2)
  Parameter (default false)
  Log accumulated values.

62.1 array

Log all members of an array.

62.2 value

Log a single numeric value.

62.3 number

A 'value' model (see 62.2, page 308) build into Daisy.

Extract specified number. If used on an array, it will treat them as individual numbers as specified by the 'handle' parameter.

Used by log Field water entries (see 39.13, page 217) , log Field nitrogen entries (see 39.14, page 218) , and log Field chemical entries (see 39.8, page 212) .

62.4 content

A 'value' model (see 62.2, page 308) build into Daisy.

Extract content at specified location. The "location" may be a line, plane or volume if one or more dimension parameters are left out. In that case, the weighted average is used.

- *x*: number [**cm**]
  Optional parameter
  Specify width (distance from left side) to measure content. The value will be a weighted average of all cells containing width. By default, cell in all widths will be included.

- *height*: number [**cm**]
  Optional parameter
  OBSOLETE: Use 'z' instead.

- *z*: number [**cm**]
  Optional parameter
  Specify height (negative below surface) to measure content. The value will be a weighted average of all cells containing height. By default, cell in all heights will be included.

- *y*: number [**cm**]
  Optional parameter
  Specify length (distance from front) to measure content. The value will be a weighted average of all cells containing length. By default, cell in all lengths will be included.

62.5 index

A 'value' model (see 62.2, page 308) build into Daisy.

Extract content at specified array index.

- *index*: integer
  Parameter
  Specify array index to select.

62.6 volume_base

A 'value' model (see 62.2, page 308) build into Daisy.
Shared parameters for volume based logs.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to log.

- *density*: boolean (see section 1.4.2)
  Parameter (default false)
  If true, divide total content with volume. Otherwise, obey 'density_z', 'density_x', and 'density_y'.

- *density_z*: boolean (see section 1.4.2)
  Parameter
  If true, divide total content with volume height. This parameter is ignored if 'density' is true.

- *density_x*: boolean (see section 1.4.2)
  Parameter
  If true, divide total content with volume width. This parameter is ignored if 'density' is true.

- *density_y*: boolean (see section 1.4.2)
  Parameter
  If true, divide total content with volume depth. This parameter is ignored if 'density' is true.

- *min_root_density*: number [$\mathbf{cm/cm^3}$]
  Parameter (default -1)
  Minimum root density in cells.

  Set this paramater to a positive amount in order to log only cells within the (dynamic) root zone. If the root density in the cell is above this amount, the full amount of the data being logged will be included. If the root density is below, the amount included will be scaled down accordingly. That is, if there are no roots, the data for the cell will be scaled to zero, while if there is only half the specified minimum root density, the data for the cell will be scaled to 0.5.

- *min_root_crop*: string (see section 1.4.5)
  Parameter (default '*')
  Name of crop whose roots scould be used for the root density requirements. Set this to "*" to use all roots.

62.7 volume

A 'volume_base' model (see 62.6, page 309) build into Daisy.
Summarize specified volume.

62.8 interval

A 'volume_base' model (see 62.6, page 309) build into Daisy.
Summarize specified interval. This is similar to 'volume', except for the default values of 'density_x' and 'density_y', and the unqiue 'from' and 'to' parameters.

Used by log Soil water entries (see 39.12, page 216) , log Field water entries (see 39.13, page 217) , log Field nitrogen entries (see 39.14, page 218) , log Soil nitrogen entries (see 39.15, page 220) , log Field chemical entries (see 39.8, page 212) , and log Soil chemical entries (see 39.9, page 214) .

- *from*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure from. By default, measure from the top. OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure interval. By default, measure to the bottom. OBSOLETE: Use (volume box (bottom TO)) instead.

62.9 water

A 'volume_base' model (see 62.6, page 309) build into Daisy.
Shared parameters for water limited volumn logging.

- *h*: number [**cm**]
  Parameter
  Pressure to log water content for.

- *h_ice*: number [**cm**]
  Parameter (default 0)
  Pressure at which all air is out of the matrix. When there are no ice, this is 0.0. When there are ice, the ice is presumed to occupy the large pores, so it is h (Theta_sat - X_ice).

62.10 water_volume

A 'water' model (see 62.9, page 310) build into Daisy.
Summarize water content in the specified volume.

62.11 water_interval

A 'water' model (see 62.9, page 310) build into Daisy.
Summarize water content in the specified interval. This is similar to 'water_volume', except for the default values of 'density_x' and 'density_y', and the unqiue 'from' and 'to' parameters.

- *from*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure from. By default, measure from the top. OBSOLETE: Use (volume box (top FROM)) instead.

- *to*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure interval. By default, measure to the bottom. OBSOLETE: Use (volume box (bottom TO)) instead.

62.12 flow

A 'value' base model (see 62.2, page 308) build into Daisy.
Common base for logging flow through a specific plane.

- *volume*: **volume** component (see chapter 81)
  Component (default 'box')
  Soil volume to log flow into.

- *density*: boolean (see section 1.4.2)
  Parameter (default false)
  If true, divide value with volume height.

62.13 flow_top

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from top of specified volume.

- *from*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure from. By default, measure from the top.
  OBSOLETE: Use (volume box (top FROM)) instead.

62.14 flux_top

A 'flow_top' model (see 62.13, page 311) build into Daisy.
Flux leaving top of specified volume. OBSOLETE: Use '(flow_top (negate true) (density true))' instead.
Used by log Soil water entries (see 39.12, page 216) , log Soil nitrogen entries (see 39.15, page 220) , and log Soil chemical entries (see 39.9, page 214) .

62.15 flow_bottom

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from bottom of specified volume.

- *to*: number [**cm**]
  Optional parameter
  Specify height (negative) to measure interval. By default, measure to the bottom. OBSOLETE: Use (volume box (bottom TO)) instead.

62.16 flux_bottom

A 'flow_bottom' model (see 62.15, page 311) build into Daisy.
Flux entering bottom of specified volume. OBSOLETE: Use '(flow_bottom (density true))' instead.
Used by log Soil water entries (see 39.12, page 216) , log Field water entries (see 39.13, page 217) , log Field nitrogen entries (see 39.14, page 218) , log Soil nitrogen entries (see 39.15, page 220) , log Field chemical entries (see 39.8, page 212) , and log Soil chemical entries (see 39.9, page 214) .

62.17 flow_left

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from left of specified volume.

62.18 flow_right

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from right of specified volume.

62.19 flow_front

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from front of specified volume.

62.20 flow_back

A 'flow' model (see 62.12, page 311) build into Daisy.
Extract flow from back of specified volume.

Chapter 63

solute

Water composition.

Chapter 64

solver

A way to solve the matrix equation 'A x = b'.

64.1 none

Don't solve the equation.

64.2 cxsparse

Solve equation using CXSparse library described in:

Direct Methods for Sparse Linear Systems, T. A. Davis, SIAM, Philadelphia, Sept. 2006. Part of the SIAM Book Series on the Fundamentals of Algorithms.

The uBLAS interface was provided by Gunter Winkler <guwi17@gmx.de>.

Used by svat SSOC solver (see 69.3, page 338) , uzrect Mollerup solver (see 78.3, page 378) , transport Mollerup solver (see 73.2, page 349) , and heatrect Mollerup solver (see 34.3, page 181) .

64.3 ublas

Solve equation using UBLAS lu functions.

Chapter 65

source

Time series, with possible error bars and formatting information.
< `component (description` *`description`*`)`
`(cite) >`

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

65.1 column

Read a a single column from a Daisy log, weather or data file.

- *timestep*: string (see section 1.4.5)
  Parameter (default 'h')
  Multiple with this dimension when accumulating.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *dimension*: string (see section 1.4.5)
  Optional parameter
  Dimension of data to plot. By default this is the same as 'original'. If 'factor' is not specified, Daisy will attempt to convert the data.

- *factor*: number (dimension not specified)
  Optional parameter
  Multiply all data by this number. By default Daisy will convert from 'original' to 'dimension'.

- *offset*: number (dimension not specified)
  Parameter (default 0)
  Add this number to all data.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

```
(missing "" "00.00")
```

Parameter description:
List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *tag*: string (see section 1.4.5)
  Parameter
  Name of column in Daisy log file where data is found.

- *handle*: string (see section 1.4.5)
  Parameter (default 'normal')
  Determine how to handle multiple simultaniously. Possible values are:

  sum: use the sum of the values.

  normal: use the arithemetic average of the values, and calculate the standard deviation.

- *accumulate*: boolean (see section 1.4.2)
  Parameter (default false)
  Accumulate values.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the same as 'tag'.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, data from dwf and dlf files will be drawn with lines, and data from ddf files will be drawn with points.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *default_hour*: integer
  Parameter (default 8)
  Hour to assume when nothing else is specified;

- *time_offset*: **Timestep** fixed component (see section 85.1)
  Submodel (has fully specified default value)
  Add this to time from sources. By default, use unmodified times.

- *reset_offset*: boolean (see section 1.4.2)
  Parameter (default false)
  Set offset to first value read. Useful for plotting already accumulated data from a later date.

65.2 arithmetic

Read a daisy log, weather or data file. Calculate a single value for each time step, based on the value in the various columns.

- *timestep*: string (see section 1.4.5)
  Parameter (default 'h')
  Multiple with this dimension when accumulating.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *expr*: **number** component (see chapter 45)
  Expression for calculating the value for this source for each row. The expression can refer to the value in a specific column by the tag for that column.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *handle*: string (see section 1.4.5)
  Parameter (default 'normal')
  Determine how to handle multiple simultaniously. Possible values are:

  sum: use the sum of the values.

  normal: use the arithemetic average of the values, and calculate the standard deviation.

- *accumulate*: boolean (see section 1.4.2)
  Parameter (default false)
  Accumulate values.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the name of the 'expr' object.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, data from dwf and dlf files will be drawn with lines, and data from ddf files will be drawn with points.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *default_hour*: integer
  Parameter (default 8)
  Hour to assume when nothing else is specified;

- *time_offset*: **Timestep** fixed component (see section 85.1)
  Submodel (has fully specified default value)
  Add this to time from sources. By default, use unmodified times.

- *valid*: **boolean** component (see chapter 15)
  Component (default 'true')
  Ignore entries if this boolean expression is false.

65.3 combine

Combine data from multiple sources with a single expression.

- *source*: **source** component (see chapter 65) sequence
  List of sources for data. The style information for the sources is ignored, but the dates, title and value is used as specified by 'expr' to calculate the combines date and value pairs.

- *expr*: **number** component (see chapter 45)
  Expression for calculating the value for this source for each row. A row is any date found in any of the member of 'source'. The expression may refer to the value of each source by its title.

- *accumulate*: boolean (see section 1.4.2)
  Parameter (default false)
  Accumulate values.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the name of the 'expr' object.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, let the first source decide.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

65.4 merge

Merge multiple timeseries into one. Any errorbars on the original timeseries are ignored, but the merged timeseries may have errorbars if there are multiple values for the same time.

- *source*: **source** component (see chapter 65) sequence
  List of timeseries to merge.

- *dimension*: string (see section 1.4.5)
  Optional parameter
  Dimension of data to plot. By default use the first source with a known dimension.

- *accumulate*: boolean (see section 1.4.2)
  Parameter (default false)
  Accumulate values.

- *title*: string (see section 1.4.5)
  Parameter
  Name of data series for the legend on the graph.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, let the first source decide.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

Chapter 66

stomatacon

The 'Stomatacon' component calculates the stomata conductance of water vapour.

< `component` `(description` *`description`*`)` `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

66.1 WSF

Common water stress effect parameters.

- *beta*: number [$\mathbf{cm^3/g}$]
  Parameter (default 0)
  Effect of ABA concentration. The effect is exp (-beta (—ABA— - ABA_min)), where —ABA— is the ABA concentration in the xylem.

- *ABA_min*: number [$\mathbf{g/cm^3}$]
  Parameter (default 0)
  Level of ABA with unstressed production.

- *delta*: number [$\mathbf{MPa^{-1}}$]
  Parameter (default 0)
  Effect of crown water potential. The effect is exp (-delta —psi_c—), where psi_c is the crown potential.

66.2 SHA12

A 'WSF' model (see 66.1, page 323) build into Daisy.

Stomata conductance calculated by the model given by Eq. 12. See also [Ahmadi et al., 2009]

- *min*: number [$\mathbf{mol\ H2O/m^2\ leaf/s}$]
  Optional parameter
  Minimal conductivity.

- *m*: number (dimension not specified)
  Parameter
  Slope parameter, dimension depends on alpha and lambda.

- *alpha*: number (dimensionless)
  Parameter (default 1)
  Humidity effect

- *lambda*: number (dimensionless)
  Parameter (default 1)
  Net photosyhtesis effect

66.3 SHA14

A 'WSF' model (see 66.1, page 323) build into Daisy.

Stomata conductance calculated by the model given by Eq. 14. See also [Ahmadi et al., 2009]

- *max*: number [**mol H2O/m^2 leaf/s**]
  Optional parameter
  Maximal conductivity. By default, there is no maxium.

- *m*: number [**mol H2O/m^2 leaf/s**]
  Parameter
  Conductivity factor.

- *alpha*: number (dimensionless)
  Parameter
  Humidity effect

- *lambda*: number [**m^2 leaf s/umol CO2**]
  Parameter
  Net photosyhtesis effect

66.4 MNA

A 'WSF' model (see 66.1, page 323) build into Daisy.

Stomata conductance calculated by the model given by Eq. 14.

- *max*: number [**mol H2O/m^2 leaf/s**]
  Optional parameter
  Maximal conductivity. By default, there is no maixum.

- *m*: number [**mol H2O/m^2 leaf/s**]
  Parameter
  Conductivity factor.

- *alpha*: number (dimensionless)
  Parameter
  Humidity effect

- *lambda*: number [**umol CO2/m^2 leaf/s**]
  Parameter
  Net photosyhtesis effect

- b: number [**mol/m^2/s**]
  Parameter
  Stomatal intercept. Ball and Berry (1982) & Wang and Leuning(1998): (0.01 mol/m2/s)

66.5 BB_base

A 'WSF' base model (see 66.1, page 323) build into Daisy.
Common parameters for Ball&Berry derived models.

- m: number (dimensionless)
  Parameter
  Stomatal slope factor. Ball and Berry (1982): m = 9 for soyabean. Wang and Leuning(1998): m = 11 for wheat
- b: number [**mol/m^2/s**]
  Parameter
  Stomatal intercept. Ball and Berry (1982) & Wang and Leuning(1998): (0.01 mol/m2/s)

66.6 BB

A 'BB_base' model (see 66.5, page 325) build into Daisy.
Stomata conductance calculated by the Ball & Berry model. See also [Ball et al., 1987]

66.7 Leuning

A 'BB_base' model (see 66.5, page 325) build into Daisy.
Stomata conductance calculated by the Leuning model. See also [Leuning, 1995]
Used by photosynthesis Farquhar Stomatacon (see 50.2, page 260) .

- Do: number [[**Pa**]]
  Parameter (default 1500)
  Empirical coefficient.

Chapter 67

string

Generic representation of strings.

67.1 number

Extract the value of a number.

- *number*: **number** component (see chapter 45)
  Number to manipulate.

67.2 value

A 'number' model (see 67.1, page 327) build into Daisy.
Extract the value of a number as a string.

- *precision*: integer
  Optional parameter
  Number of decimals after point. By default, use a floating format.

67.3 dimension

A 'number' model (see 67.1, page 327) build into Daisy.
Extract the dimension of a number as a string.

67.4 cond

Return the value of the first clause whose condition is true.
Used by log chemical chemfid (see 39.7, page 212) , log crop cropfid (see 39.10, page 215) , log column colfid (see 39.6, page 212) , and log biopore bioporefid (see 39.11, page 215) .

- *clauses*: **StringerCondClause** fixed component (see section 85.30) sequence
  List of clauses to match for.

67.5 identity

Return the specified value.

- *value*: string (see section 1.4.5)
  Parameter
  Constant value.

Chapter 68

summary

Summary reports for log parameterizations.
< `component (description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

68.1 simple

A simple log file summary model.

- *fetch*: **FetchPretty** fixed component (see section 85.28) sequence
  List of columns to fetch for the summary.

- *precision*: integer
  Parameter (default 2)
  Number of digits to print after decimal point.

- *period*: string (see section 1.4.5)
  Optional parameter
  Set this to 'y', 'm', 'w', 'd' or 'h' to get fluxes per time period instead of total amount.

- *where*: string (see section 1.4.5)
  Optional parameter
  File name to store the summary. By default, the summary will be stored in daisy.log and the screen.

- *title*: string (see section 1.4.5)
  Optional parameter
  Title of this summary. By default, use the name of the parameterization.

- *print_sum*: boolean (see section 1.4.2)
  Parameter (default true)
  Print sum of all the summary lines.

- *sum_name*: string (see section 1.4.5)
  Parameter (default 'Sum')
  Name of the sum of all the entries.

68.2 balance

A summary model providing a balance for a log parameterization.

Used by log Field chemical summary (see 39.8, page 212) , and log Soil chemical summary (see 39.9, page 214) .

- *content*: string (see section 1.4.5) sequence
  Parameter
  Tags of columns in log file representing content.
- *precision*: integer
  Parameter (default 2)
  Number of digits to print after decimal point.
- *where*: string (see section 1.4.5)
  Optional parameter
  File name to store the summary. By default, the summary will be stored in daisy.log and the screen.
- *input*: string (see section 1.4.5) sequence
  Parameter
  Tags of columns in log file representing inputs.
- *title*: string (see section 1.4.5)
  Optional parameter
  Title of this summary. By default, use the name of the parameterization.
- *require_top*: boolean (see section 1.4.2)
  Parameter (default false)
  If the balance only hold true when logging the top of the soil, i.e. the 'from' parameter of the log model is 0, this flag should be set.
- *output*: string (see section 1.4.5) sequence
  Parameter
  Tags of columns in log file representing outputs.

68.3 Field water balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.

Water balance for the complete system.

Used by log Field water summary (see 39.13, page 217) .

68.4 Soil water balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.

Water balance for the soil matrix.

The intended use of this log is small scale water management, for example irrigation optimization. Above ground water and water in biopores is considered external to the system. The balances provided by the log will work on any subset of the soil that matches the discretization.

Used by log Soil water summary (see 39.12, page 216) .

68.5 Field nitrogen balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 39.14, page 218) .

68.6 Field mineral nitrogen balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Mineral balance for the complete system.
Used by log Field nitrogen summary (see 39.14, page 218) .

68.7 Field crop nitrogen balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Crop nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 39.14, page 218) .

68.8 Field organic nitrogen balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Organic nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 39.14, page 218) .

68.9 N_balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Unit is kg, precion is g.

68.10 Soil NO3- balance

A 'N_balance' model (see 68.9, page 331) defined in 'log-std.dai'.
NO3- balance for a soil interval.
Used by log Soil nitrogen summary (see 39.15, page 220) .

68.11 Soil NH4+ balance

A 'N_balance' model (see 68.9, page 331) defined in 'log-std.dai'.
NH4+ balance for a soil interval.
Used by log Soil nitrogen summary (see 39.15, page 220) .

68.12 Soil mineral-N balance

A 'N_balance' model (see 68.9, page 331) defined in 'log-std.dai'.
Mineral N balance for a soil interval.
Used by log Soil nitrogen summary (see 39.15, page 220) .

68.13 Soil organic-N balance

A 'N_balance' model (see 68.9, page 331) defined in 'log-std.dai'.
Organic nitrogen balance for a soil interval.
Used by log Soil nitrogen summary (see 39.15, page 220) .

68.14 Soil N balance

A 'balance' model (see 68.2, page 330) defined in 'log-std.dai'.
Nitrogen balance for a soil interval.
Used by log Soil nitrogen summary (see 39.15, page 220) .

Chapter 69

svat

The task of the 'svat' component is to calculate the production stress, given the potential evapotranspiration, the actual evaporation from the surface, meteorological data, and the vegetation and soil state.

69.1 PMSW

Peter van der Keur's SVAT model.

- *albedo*: number (dimensionless)
  Parameter (default 0.2)
  Bulk albedo

- *b1*: number (dimensionless)
  Parameter (default 0.53)
  Brunt coefficient 1

- *b2*: number (dimensionless)
  Parameter (default 0.0065)
  Brunt coefficient 2

- *b3*: number (dimensionless)
  Parameter (default 0.1)
  Brunt coefficient 3

- *b4*: number (dimensionless)
  Parameter (default 0.9)
  Brunt coefficient 4

- *ndif*: number (dimensionless)
  Parameter (default 2.5)
  Eddy diffusivity decay constant in crop

- *c_d*: number (dimensionless)
  Parameter (default 0.05)
  Mean drag coefficient for a leaf

- *z_0s*: number [**m**]
  Parameter (default 0.01)
  Roughness length for soil surface, SG (1990)

- *z0_def*: number [**m**]
  Parameter (default 0.005)
  Roughness length for soil surface, Oke

- *w*: number [**m**]
  Parameter (default 0.0025)
  average leaf width

- *alpha_u*: number (dimensionless)
  Parameter (default 3)
  attenuation coefficient for wind speed

- *arac*: number (dimensionless)
  Parameter (default 0.00662)
  leaf boundary layer resistance coefficient

- *alpha_k*: number (dimensionless)
  Parameter (default 2)
  Att. coefficient of eddy diffusivity through sparse canopy

- *alpha_r*: number (dimensionless)
  Parameter (default 0.5)
  Att. coefficient for vegetation in ACOEFF()

- *theta_w*: number [$\mathbf{cm^3/cm^3}$]
  Parameter (default 0.05)
  Soil water content at 'wilting point'

- *theta_c*: number [$\mathbf{cm^3/cm^3}$]
  Parameter (default 0.25)
  Soil water content at 'field capacity'

- *rcmin_const*: number [**s/m**]
  Parameter (default 30)
  Constant minimum canopy resistance

- *rcmax*: number (dimensionless)
  Parameter (default 1000)
  Maximum canopy resistance

- *tref*: number (dimensionless)
  Parameter (default 298)
  Reference/optimum temperature in temperature dependent constraint function

- *zeta*: number (dimensionless)
  Parameter (default 0.0002)
  Coefficient in vapor pressure dependent constraint function

- *f3const*: number (dimensionless)
  Parameter (default 0.0016)
  Coefficient in temperature dependent constraint function

- *spar*: number (dimensionless)
  Parameter (default 100)
  Reference value of photosynthetically active part of Si

- *tmin*: number [**dg C**]
  Parameter (default 0)
  Minimum temperature for canopy conductance
- *tmax*: number [**dg C**]
  Parameter (default 55)
  Maximum temperature for canopy conductance
- *nu_1*: number (dimensionless)
  Parameter (default 26.5)
  coefficient in Jarvis (1976) constraint function f_temp
- *nu_2*: number (dimensionless)
  Parameter (default 0.57)
  coefficient in Lohammar (1980) constraint function f_def
- *nu_3*: number (dimensionless)
  Parameter (default 0.008)
  coefficient in Steward (1988) constraint function f_theta
- *dt1*: number [**dg C**]
  Parameter (default -5)
  lower solution limit in Newton-Raphson method
- *dt2*: number [**dg C**]
  Parameter (default 5)
  upper solution limit in Newton-Raphson method
- *acc*: number (dimensionless)
  Parameter (default 0.01)
  iteration accuracy in Newton-Raphson method

Log Variables

- *ha*: number [$\mathbf{W/m^2}$]
  Sensible heat flux from source- to screen height
- *netrad_brunt*: number [$\mathbf{W/m^2}$]
  Net radiation by Brunt
- *netlong_brunt*: number [$\mathbf{W/m^2}$]
  Net long radiation by Brunt
- *r_a*: number [**s/m**]
  bulk aerodynamic resistance, neutral conditions
- *r_astab*: number [**s/m**]
  bulk aerodynamic resistance, stability corrected
- *r_aa*: number [**s/m**]
  aerodynamic resistance mean source-ref, uncorrected
- *r_aastab1*: number [**s/m**]
  aerodynamic resistance mean source-ref, corrected 'method 1'
- *r_aastab2*: number [**s/m**]
  aerodynamic resistance mean source-ref, corrected,'method 2'
- *r_as*: number [**s/m**]
  aerodynamic resistance from soil to mean source

- *r_ac*: number [**s/m**]
  aerodynamic resistance from leaf to mean source
- *r_sc_1*: number [**s/m**]
  Bulk canopy resistance (Noilhan et al., 1991)
- *r_sc_2*: number [**s/m**]
  Bulk canopy resistance (Verma et al., 1993)
- *tskin*: number [**dg C**]
  soil/skin temperature
- *tcan*: number [**dg C**]
  canopy temperature at mean source
- *tleaf*: number [**dg C**]
  Leaf temperature
- *e_c_abs*: number [**Pa**]
  vapor pressure at mean source height
- *e_sl_abs*: number [**Pa**]
  saturated vapor pressure at leaf surface
- *hl*: number [$\mathbf{W/m^2}$]
  Sensible heat flux from leaf to mean source
- *hs*: number [$\mathbf{W/m^2}$]
  Sensible heat flux from soil to mean source
- *lea*: number [$\mathbf{W/m^2}$]
  Latent heat flux from source- to screen height
- *lel*: number [$\mathbf{W/m^2}$]
  Latent heat flux from leaf to mean source
- *gflux*: number [$\mathbf{W/m^2}$]
  Ground heat flux
- *dtcta*: number [**dg C**]
  Temperature gradient between mean source og screen height
- *dtltc*: number [**dg C**]
  Temperature gradient between leaf and mean source
- *dtstc*: number [**dg C**]
  Temperature gradient between soil and mean source
- *dtcta_star*: number [**dg C**]
  corrected temp gradient between mean source og screen height
- *dtltc_star*: number [**dg C**]
  corrected temperature gradient between leaf and mean source
- *dtstc_star*: number [**dg C**]
  corrected temperature gradient between soil and mean source
- *theta_0_20*: number [$\mathbf{cm^3/cm^3}$]
  Averaged soil water content in upper 20 cm

- *f_1*: number (dimensionless)
  Constraint function (Noilhan) related to solar radiation
- *f1_dolman*: number (dimensionless)
  ???
- *f_2*: number (dimensionless)
  Constraint function (Noilhan) related to vapor pressure
- *f_3*: number (dimensionless)
  Constraint function (Noilhan) related to air temperature
- *f_4*: number (dimensionless)
  Constraint function (Noilhan) related to soil water content
- *f_temp*: number (dimensionless)
  Constraint function (Verma) related to air temperature
- *f_def*: number (dimensionless)
  Constraint function (Verma) related to vapor pressure
- *f_theta*: number (dimensionless)
  Constraint function (Steward) related to soil water content
- *f_etep*: number (dimensionless)
  Constraint function defined by crop_ea/crop_ep
- *r_sc_js*: number [**s/m**]
  Bulk canopy resistance: f1_dolman*f_def*f3*f4
- *r_sc*: number [**s/m**]
  Bulk canopy resistance: f1_dolman*f_def*f3*f_etep
- *rcmin_star*: number [**s/m**]
  minimum canopy resistance
- *pstress*: number (dimensionless)
  crop production stress
- *ustar_raa*: number (dimensionless)
  friction velocity from RAA()
- *ustar_raastab1*: number (dimensionless)
  friction velocity from RAASTAB1()
- *ustar_raastab2*: number (dimensionless)
  friction velocity from RAASTAB2()
- *env_lai_factor*: number (dimensionless)
  LAI*F_i
- *e_pa*: number [**Pa**]
  vapor pressure at 2 m
- *e_abs*: number [**kg/m^3**]
  absolute vapor pressure
- *tair*: number [**degr.C**]
  air temperature

- *srad*: number [$\mathbf{W/m^2}$]
  global radiation
- *u_ref*: number [**m/s**]
  friction velocity from ??
- *prec*: number [**mm**]
  precipitation

69.2 none

No SVAT in effect.
Used by bioclimate default svat (see 13.1, page 101) .

69.3 SSOC

Sun-Shade Open Canopy.

- *solver*: **solver** component (see chapter 64)
  Component (default 'cxsparse')
  Model used for solving the energy balance equation system.
- *hypostomatous*: boolean (see section 1.4.2)
  Parameter (default true)
  True for hypostomatous leaves. False for amphistomatous leaves (possesing stomata on both surfaces).
- *max_iteration*: integer
  Parameter (default 1500)
  Largest number of iterations before giving up on convergence.
- *z_0b*: number [**m**]
  Parameter (default 0.0006)
  Bare soil roughness height for momentum.

Log Variables

- *LAI*: number [$\mathbf{m^2\ m^{-2}}$]
  Leaf area index.
- *s*: number [$\mathbf{Pa\ K^{-1}}$]
  Slope of water vapour pressure curve.
- *lambda*: number [$\mathbf{J\ kg^{-1}}$]
  Latent heat of vaporization in atmosphere.
- *rho_a*: number [$\mathbf{kg\ m^{-3}}$]
  Air density.
- *gamma*: number [$\mathbf{Pa\ K^{-1}}$]
  Psychrometric constant.
- *T_s*: number [**K**]
  Soil surface temperature.
- *T_0*: number [**K**]
  Surface temperature (large scale).

- *T_c*: number [**K**]
  Canopy-point temperature.
- *T_sun*: number [**K**]
  Temperature of sunlit leaves.
- *T_shadow*: number [**K**]
  Temperature of shadow leaves.
- *g_a*: number [$\mathbf{m\ s^{-1}}$]
  Heat conductance in the atmosphere - from canopy point to reference height (screen height).
- *g_H_s_c*: number [$\mathbf{m\ s^{-1}}$]
  Heat conductance from soil surface to canopy point.
- *g_H_sun_c*: number [$\mathbf{m\ s^{-1}}$]
  Heat conductance from sunlit leaves to canopy point.
- *gb_W_sun*: number [$\mathbf{m\ s^{-1}}$]
  Water conductance for sunlit leaves boundary layer.
- *g_W_sun_c*: number [$\mathbf{m\ s^{-1}}$]
  Water conductance from sunlit leaves to canopy point.
- *G_W_sun_c*: number [$\mathbf{W\ m^{-2}\ K^{-1}}$]
  Scaled water conductance from sunlit leaves to canopy point.
- *g_H_shadow_c*: number [$\mathbf{m\ s^{-1}}$]
  Heat conductance from shadow leaves to canopy point.
- *gb_W_shadow*: number [$\mathbf{m\ s^{-1}}$]
  Water conductance for shadow leaves boundary layer.
- *g_W_shadow_c*: number [$\mathbf{m\ s^{-1}}$]
  Water conductance from shadow leaves to canopy point.
- *e_a*: number [**Pa**]
  Vapour pressure of water in the atmosphere.
- *e_sat_air*: number [**Pa**]
  Saturated vapour pressure of water in the air.
- *e_c*: number [**Pa**]
  Vapour pressure of water in the canopy.
- *R_abs_soil*: number [$\mathbf{W\ m^{-2}}$]
  Absorbed radiation in soil.
- *R_eq_abs_soil*: number [$\mathbf{W\ m^{-2}}$]
  Absorbed radiation in soil at equilibrium.
- *R_abs_sun*: number [$\mathbf{W\ m^{-2}}$]
  Absorbed radiation in sunlit leaves.
- *R_eq_abs_sun*: number [$\mathbf{W\ m^{-2}}$]
  Absorbed radiation in sunlit leaves at equilibrium.
- *R_abs_shadow*: number [$\mathbf{W\ m^{-2}}$]
  Absorbed radiation in shadow leaves.

- *R_eq_abs_shadow*: number [**W m**$^{-2}$]
  Absorbed radiation in shadow leaves at equilibrium.
- *sun_LAI_fraction_total*: number []
  Sunlit fraction of leaf area in the canopy.
- *cover*: number []
  Vegetation cover.
- *H_soil*: number [**W m**$^{-2}$]
  Sensible heat flux from the soil.
- *H_sun*: number [**W m**$^{-2}$]
  Sensible heat flux from the sunlit leaves to the canopy point.
- *H_shadow*: number [**W m**$^{-2}$]
  Sensible heat flux from the shadow leaves to canopy point.
- *H_c_a*: number [**W m**$^{-2}$]
  Sensible heat flux from the canopy point to free atmosphere.
- *LE_sun*: number [**W m**$^{-2}$]
  Latent heat flux from the sunlit leaves to the canopy point.
- *LE_shadow*: number [**W m**$^{-2}$]
  Latent heat flux from the shadow leaves to the canopy point.
- *LE_atm*: number [**W m**$^{-2}$]
  Latent heat flux from the canopy point to the free atmosphere.
- *E_trans*: number [**mm/h**]
  Leaf transpiration.

Chapter 70

tertiary

Transport of water and solute outside the matrix.
Used by movement component Tertiary (see 42, page 229) .

70.1 none

No tertiary transport.
Used by movement rectangle Tertiary (see 42.2, page 229) .

70.2 old

Tertiary water and solute movement based on the obsolete 'macro' and 'mactrans' components. Provided for backward compatibility.
Used by movement vertical Tertiary (see 42.3, page 230) .

- *macro*: **macro** component (see chapter 40)
  Optional component
  Preferential flow model. By default, preferential flow is enabled if and only if the combined amount of humus and clay in the top horizon is above 5%.

- *mactrans*: **mactrans** component (see chapter 41)
  Component (default 'default')
  Solute transport model in macropores.

70.3 biopores

Tertiary domain divided into biopore classes.

- *pressure_initiate*: number [**cm**]
  Parameter (default -3)
  Pressure needed to activate biopore flow.

- *pressure_end*: number [**cm**]
  Parameter (default -30)
  Pressure below which biopore flow is deactivated.

- *pond_max*: number [**cm**]
  Parameter (default 0.05)
  Maximum height of ponding before spilling into biopores. After macropores are activated pond will have this height.

- *classes*: **biopore** component (see chapter 14) sequence
  List of biopore classes.

- *pressure_limit*: number [**cm**]
  Optional parameter
  Limit to pressure difference for moving matrix water gradient to biopores.

  The idea is that the water is extracted from the matrix by a hanging water column in the biopore, and that the suction is equal to the height of this water column. The pressure limit is then the maximal length of the column, or the point where the column breaks.

  By default, this is equal to 'pressure_end'.

- *pressure_barrier*: number [**cm**]
  Parameter (default 5)
  Pressure barrier between matrix and biopore domain. If the pressure difference between the matrix and biopores is below this value, no water will tranfer between the domains. If you specify a too small value for this parameter, the solution may be unstable.

- *active_msg*: string (see section 1.4.5)
  Parameter (default 'range')
  Control biopore activation and deactivation reports.

  Possible values: cell: Report for each cell. range: Report for vertical range. none: No reports.

- *active*: boolean (see section 1.4.2) soil cells
  Optional state variable
  Active biopores in cells.

- *deactivate_steps*: integer
  State variable (default 3)
  No matrix exchange for this number of timesteps. Automatically set when matrix pressure is in a disarray, such as after tillage operations, or calls to reserve models.

Log Variables

- *water_volume*: number [$\mathbf{cm}^3$]
  Water volume.

- *water_height*: number [**cm**]
  Water volume multiplied with surface area.

- *solute_mass*: submodel (see section 1.4.7) sequence
  Total amount of solutes in biopores.

  < *name value* > **Log Variables**

  - *value*: number [**g**]
    Value for chemical.
  - *name*: string (see section 1.4.5)
    Name of chemical.

- *solute_storage*: submodel (see section 1.4.7) sequence
  Total amount of solutes in biopores divided by surface area.

  < *name value* > **Log Variables**

  - *value*: number [**g/cm**^{2}]
    Value for chemical.
  - *name*: string (see section 1.4.5)
    Name of chemical.

- *ddt*: number [**h**]
  Emulated timestep. Timestep scaled for available water.

Chapter 71

tortuosity

Solutes in the soil can't move the shortest way between two points. The tortuosity factor indicates how far the average solute have moved in absolute coordinates, when it has moved a given distance along the curved line. This component is responsible for calculating the soils tortuosity factor.

Used by horizon component tortuosity (see 35, page 183) .

71.1 linear

Linear Impedance factor. a + b Theta.

- *b*: number (dimensionless)
  Parameter (default 2)
  Theta factor.
- *a*: number [$\mathbf{cm}^3/\mathbf{cm}^3$]
  Optional parameter
  Theta offset. By default, this corresponds to the wilting point.

71.2 M_Q

Millington-Quirk. Theta^(7/3) / Theta_sat^2.

Used by horizon component tortuosity (see 35, page 183) .

Chapter 72

transform

Generic transformations between soil components.
< `component` `(description` *`description`*`)`
`(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

72.1 equilibrium

Two soil components reching for equilibrium.

- *equilibrium*: **equilibrium** component (see chapter 29)
  Function for calculating equilibrium between A and B.

- *k_AB*: **number** component (see chapter 45)
  Tranformation rate from soil component 'A' to 'B' [h^{-1}].

- *k_BA*: **number** component (see chapter 45)
  Optional component
  Tranformation rate from soil component 'B' to 'A' [h^{-1}]. By default, this is identical to 'k_AB'.

Chapter 73

transport

Solute transport in primary domain.

73.1 none

Disable all transport except through boundaries.

Used by movement solute matrix_solid (see 42.1, page 229) , movement rectangle matrix_solute (see 42.2, page 229) , and movement vertical matrix_solute (see 42.3, page 230) .

73.2 Mollerup

Coupled vertical and horizontal transport. See Mollerup 2007 for details.

Used by movement rectangle matrix_solute (see 42.2, page 229) .

- *solver*: **solver** component (see chapter 64)
  Component (default 'cxsparse')
  Model used for solving matrix equation system.

- *debug*: integer
  Parameter (default 0)
  Enable additional debug message. A value of 0 means no message, higher numbers means more messages.

- *enable_boundary_diffusion*: boolean (see section 1.4.2)
  Parameter (default true)
  If this is set, diffusion over boundaries is enabled.

- *upstream_weight*: number [<**fraction**>]
  Parameter (default 1)
  Upstream weighting factor: 1 = full upstream formulation, 0.5 = equal weight.

73.3 convection

Pure forward calculation of flow except through upper boundary. J[edge] = q[edge] * C_old[upstream]

Used by movement rectangle matrix_solute (see 42.2, page 229) , and movement vertical matrix_solute (see 42.3, page 230) .

73.4 Hansen

Solute transport using convection-dispersion.

Used by movement vertical matrix_solute (see 42.3, page 230) .

Chapter 74

ui

Top level user interface.
Used by Toplevel @ ui (see 85.27, page 445) .

74.1 none

No user unterface.
This is useful when running from a batch program, or as a component in a larger system.

74.2 progress

Write progress on standard output (or standard error).
This is useful when starting the program from a text terminal, or from inside another program such as an editor that can capture the output.

Chapter 75

unit

Specify units of physical quatities.

Daisy will recognise both the build-in and user defined units, and convert between different units representing the same physical dimension. This is done by converting to and from the base (usually defined by SI) unit for that dimension.

< `component` `(description` *`description`*`)` `(cite)` >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

75.1 base

A base unit.

75.2 dgEast

A 'base' model (see 75.1, page 353) build into Daisy.

Degrees East of Greenwich.

75.3 dgNorth

A 'base' model (see 75.1, page 353) build into Daisy.

Degrees North of Equator.

75.4 <unknown>

A 'base' model (see 75.1, page 353) build into Daisy.

Nothing is known about the dimension of this unit.

75.5 rad

A 'base' model (see 75.1, page 353) build into Daisy.
Radians

75.6 <error>

A 'base' model (see 75.1, page 353) build into Daisy.
Bogus unit.

75.7 pF

log10 (- cmH2O).

75.8 SI

Base parameterization for all SI based units.

- *time*: integer
  Parameter (default 0)
  Dimension, base unit [s].
- *length*: integer
  Parameter (default 0)
  Dimension, base unit [m].
- *mass*: integer
  Parameter (default 0)
  Dimension, base unit [kg].
- *electric_current*: integer
  Parameter (default 0)
  Dimension, base unit [A].
- *thermodynamic_temperature*: integer
  Parameter (default 0)
  Dimension, base unit [K].
- *amount_of_substance*: integer
  Parameter (default 0)
  Dimension, base unit [mol].
- *luminous_intensity*: integer
  Parameter (default 0)
  Dimension, base unit [cd].

75.9 SIfactor

A 'SI' model (see 75.8, page 354) build into Daisy.
Connvert to SI base units by multiplying with a factor.

- *factor*: number (dimensionless)
  Parameter
  Factor to multiply with to get base unit.

75.10 none

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Unitless.

75.11 g/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Solute concentration.

75.12 mm/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Percolation intensity.

75.13 $g/cm^2/mm$

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Irrigation and percolation concentration.

75.14 kg/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Harvest and fertilizing.

75.15 cd

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Candela.

75.16 kg N/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale application and removal of nitrogen.

75.17 mol/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mole per square meter.

75.18 h^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Hour.

75.19 h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Hour.

75.20 g/cm/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per length flux.

75.21 cm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Centimeter.

75.22 m

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Meter.

75.23 kPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kilopascal.

75.24 hPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Hectopascal.

75.25 Pa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pascal.

75.26 s^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Second.

75.27 d^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Day.

75.28 d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Day.

75.29 mm/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Percolation intensity.

75.30 cm/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil water movement.

75.31 m/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base speed.

75.32 cm/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil water movement.

75.33 Mg w.w./ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Wet weight per area.

75.34 T w.w./ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Wet weight per area.

75.35 g w.w./m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Wet weight per area.

75.36 kg w.w./ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Wet weight per area.

75.37 mg N/l

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen concentration.

75.38 ppm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Part per million.

75.39 cm^3/g

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Volume per mass.

75.40 L/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Volume per mass.

75.41 l/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Volume per mass.

75.42 kg/m^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per volume.

75.43 mg/l

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.44 mmol/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Millimole per square meter.

75.45 mg N/kg dry soil

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen concentration in dry soil.

75.46 <fraction>

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Unitless.

75.47 %

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Percent.

75.48

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Unitless.

75.49 <none>

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Unitless.

75.50 cm^3/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil water fraction.

75.51 cm^3 H2O/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil water fraction.

75.52 $g/cm^2/h$

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mass per area flux.

75.53 kg/ha/y

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Deposition.

75.54 g/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.55 g/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Crop scale mass per area.

75.56 g/cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil scale mass per area.

75.57 mg/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Crop scale pesticide per area.

75.58 kg/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale mass per area.

75.59 kg/ha/mm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Irrigation and percolation concentration.

75.60 kg C/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale application and removal of carbon.

75.61 g/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale pesticide mass per area.

75.62 cm^3/ng

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Volume per mass.

75.63 MPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Megapascal.

75.64 cm^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Per centimeter.

75.65 MPa^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Inverse pressure.

75.66 ng/mm/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mass per length flux.

75.67 ng/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Low solute concentration.

75.68 cm^{-2}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Per aquare centimeter.

75.69 m^{-2}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Per aquare meter.

75.70 W/m²

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Watt per square meter.

75.71 MJ/d/m²

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Megajoule per day per square meter.

75.72 MJ/m²/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Megajoule per square meter per day.

75.73 mol/m²/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mole per square meter per second.

75.74 mmol/m²/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Millimole per square meter per second.

75.75 kgN/ha/year

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Deposition.

75.76 mm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Millimeter.

75.77 g/m²/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Gram per square meter per hour.

75.78 ug/m³

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mass per volume.

75.79 ng/l

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Lower solute concentration.

75.80 kg N/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale nitrogen per area.

75.81 g N/cm^2/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen per aquare centimeter per hour.

75.82 g N/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Crop scale nitrogen per area.

75.83 kg C/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale carbon per area.

75.84 g N/cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil scale nitrogen per area.

75.85 g C/cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Soil scale carbon per area.

75.86 g C/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Carbon concentration.

75.87 g N/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen concentration.

75.88 Mg DM/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Ton dry matter per hectar.

75.89 g DM/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Crop scale dry matter per area.

75.90 kg DM/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kilogram dry matter per hectar.

75.91 Mg DM/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Harvest and fertilizing.

75.92 t/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Ton per hectar.

75.93 m^{-2} kg s^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per area flux.

75.94 m s^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base speed.

75.95 kg N/ha/y

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Deposition.

75.96 kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kilogram.

75.97 s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Second.

75.98 A

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Ampere.

75.99 K

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kelvin.

75.100 mol

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mole.

75.101 mg/g

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Milligram per gram.

75.102 g/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Gram per kilogram.

75.103 m^3/cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kilo.

75.104 um

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Micrometer.

75.105 l/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Liter per hectar.

75.106 m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Square meter.

75.107 cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Square centimeter.

75.108 ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Hectare.

75.109 m^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Cube meter.

75.110 cm^3

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Cube centimeter.

75.111 m^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Per meter.

75.112 mm^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Per millimeter.

75.113 g

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Gram.

75.114 mg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Miligram.

75.115 ug

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Microgram.

75.116 hours

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Hour.

75.117 mm/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Percolation intensity.

75.118 l/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Liter per hectar per hour.

75.119 kg/m

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per length.

75.120 kg/m/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per length flux.

75.121 kg/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per area.

75.122 kg DM/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per area.

75.123 g C/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Crop scale carbon per area.

75.124 mg/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Miligram per hectare

75.125 ug/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Microgram per hectare

75.126 ug/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Microgram per square meter.

75.127 Mg/ha

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Ton per hectar.

75.128 g DM/cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Gram dry matter per square centimeter.

75.129 kg/m^2/s

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per area flux.

75.130 kg m^{-2} s^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per area flux.

75.131 g $N/m^2/h$

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen per aquare meter per hour.

75.132 mg/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.133 mg/ha/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.134 ug/ha/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.135 ug/ha/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.136 g/ha/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Pesticide application.

75.137 kg N/ha/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale nitrogen.

75.138 kg C/ha/d

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Field scale carbon.

75.139 ug/m^2/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Microgram per square meter per hour.

75.140 ug/l

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Low solute concentration.

75.141 mg/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.142 ug/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.143 ng/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.144 pg/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.145 fg/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.146 ag/L

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Concentration.

75.147 g N/l

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nitrogen concentration.

75.148 kg N/ha/mm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Irrigation and percolation concentration.

75.149 g/ha/mm

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
gram per hectare per millimeter.

75.150 m^3/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base volume per mass.

75.151 ml/g

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Volume per mass.

75.152 Pa^{-1}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base inverse pressure.

75.153 kg/Pa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Base mass per pressure.

75.154 ng/kPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nanogram per kilopascal.

75.155 ng/MPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Nanogram per megapascal.

75.156 ug/L/MPa

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Microgram per liter per megapascal.

75.157 mol/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Mole per kilogram.

75.158 mmol/kg

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Millimole per kilogram.

75.159 kg/mol

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Kilogram per mole.

75.160 g/mol

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Gram per mole.

75.161 J

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Joule.

75.162 W

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Watt.

75.163 W m^{-2}

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Watt per square meter.

75.164 W/cm^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Watt per square centimeter.

75.165 J/cm^2/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Joule per square centimeter per hour.

75.166 MJ/h/m^2

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Megajoule per hour per square meter.

75.167 MJ/m^2/h

A 'SIfactor' model (see 75.9, page 354) build into Daisy.
Megajoule per square meter per hour.

75.168 factor

Connvert to base units by multiplying with a factor.

- *base*: string (see section 1.4.5)
  Parameter
  Base unit to convert to and from.
- *factor*: number (dimensionless)
  Parameter
  Factor to multiply with to get base unit.

75.169 dgWest

A 'factor' model (see 75.168, page 370) build into Daisy.
Degrees West of Greenwich.

75.170 dgSouth

A 'factor' model (see 75.168, page 370) build into Daisy.
Degrees North of Equator.

75.171 dg

A 'factor' model (see 75.168, page 370) build into Daisy.
Degrees

75.172 new dg

A 'factor' model (see 75.168, page 370) build into Daisy.
New degrees

75.173 offset

Connvert to base units by multiplying factor, then substracting offset.

- *base*: string (see section 1.4.5)
  Parameter
  Base unit to convert to and from.
- *factor*: number (dimensionless)
  Parameter (default 1)
  Factor to multiply with to get base unit.
- *offset*: number (dimensionless)
  Parameter (default 0)
  Offset to add after multiplying with factor to get base unit.

75.174 dg C

A 'offset' model (see 75.173, page 371) build into Daisy.
degree Celcius.

75.175 dg F

A 'offset' model (see 75.173, page 371) build into Daisy.
degree Fahrenheit.

Chapter 76

uz1d

The 'uz1d' component handles the horizontal water movement in the unsaturated zone soil matrix.

76.1 none

Disable transport
Used by uzrect v+h horizontal (see 78.2, page 377) .

76.2 richards

A numerical solution to Richard's Equation.

- *max_time_step_reductions*: integer
  Parameter (default 4)
  Number of times we may reduce the time step before giving up
- *time_step_reduction*: integer
  Parameter (default 4)
  Divide the time step with this at each reduction.
- *max_iterations*: integer
  Parameter (default 25)
  Maximum number of iterations when seeking convergence before reducing the time step.
- *max_absolute_difference*: number [**cm**]
  Parameter (default 0.02)
  Maximum absolute difference in 'h' values for convergence.
- *max_relative_difference*: number (dimensionless)
  Parameter (default 0.001)
  Maximum relative difference in 'h' values for convergence.
- *K_average*: **average** component (see chapter 12)
  Optional component (default 'arithmetic')
  Model for calculating average K between cells.

Chapter 77

uzmodel

The 'uzmodel' component handles the vertical water movement in the unsaturated zone soil matrix.

77.1 none

No water movement, and no sink.

77.2 lr

Use gravitational water movement for wet soil, where h > h_fc. There are no water movement when h < h_fc, except at the layers down to z_top, where there can be Darcy movement.

Used by uzrect v+h vertical (see 78.2, page 377) , and movement vertical matrix_water (see 42.3, page 230) .

- *overflow_warn*: boolean (see section 1.4.2)
  Parameter (default true)
  If true, warn the first time the soil profile is oversaturated.
- *h_fc*: number [**cm**]
  Parameter (default -100)
  Field capacity.
- *z_top*: number [**cm**]
  Parameter (default -10)
  Depth of layer where upward water movement is possible.

77.3 richards

A numerical solution to Richard's Equation.

Used by uzrect v+h vertical (see 78.2, page 377) , and movement vertical matrix_water (see 42.3, page 230) .

- *debug*: integer
  Parameter (default 0)
  Print additional debug messages, higher numbers means more messages.
- *max_time_step_reductions*: integer
  Parameter (default 4)
  Number of times we may reduce the time step before giving up

- *time_step_reduction*: integer
  Parameter (default 4)
  Divide the time step with this at each reduction.

- *max_iterations*: integer
  Parameter (default 25)
  Maximum number of iterations when seeking convergence before reducing the time step.

- *max_absolute_difference*: number [**cm**]
  Parameter (default 0.02)
  Maximum absolute difference in 'h' values for convergence.

- *max_relative_difference*: number (dimensionless)
  Parameter (default 0.001)
  Maximum relative difference in 'h' values for convergence.

- *K_average*: **average** component (see chapter 12)
  Optional component
  Model for calculating average K between cells. The default model is 'geometric' if there are macropores, and 'arithmetic' otherwise.

Chapter 78

uzrect

2D water movement in a rectangular grid.

78.1 const

Steady-state water flow.
Used by movement rectangle matrix_water (see 42.2, page 229) .

- *q_x*: number [**cm/h**]
  Parameter (default 0)
  Horizontal flow.

- *q_z*: number [**cm/h**]
  Parameter (default 0)
  Vertical flow upwards.

78.2 v+h

Transport water in the matrix in two phases, first vertical, then horizontal.
Used by movement rectangle matrix_water (see 42.2, page 229) .

- *vertical*: **uzmodel** component (see chapter 77) sequence
  Component (has default value with length 2)

  ```
  (vertical richards
            lr)
  ```

  Parameter description:
  Vertical matrix water transport models. Each model will be tried in turn, until one succeeds. If none succeeds, the simulation ends.

- *horizontal*: **uz1d** component (see chapter 76) sequence
  Component (has default value with length 1)

  ```
  (horizontal none)
  ```

  Parameter description:
  Horizontal matrix water transport models. Each model will be tried in turn, until one succeeds. If none succeeds, the simulation ends.

78.3 Mollerup

A finite volume solution to matrix water transport. See Mollerup 2007 for details.
Used by movement rectangle matrix_water (see 42.2, page 229) .

- *solver*: **solver** component (see chapter 64)
  Component (default 'cxsparse')
  Model used for solving matrix equation system.

- *debug*: integer
  Parameter (default 0)
  Level of debug messages:

  = 0: no debug messages. > 0: Initial h and Theta per time step. > 1: Same, per iteration. = 3: Upper boundary extra info. = 4: Drain extra info. = 5: Remaining water.

- *max_time_step_reductions*: integer
  Parameter (default 4)
  Number of times we may reduce the time step before giving up

- *time_step_reduction*: integer
  Parameter (default 4)
  Divide the time step with this at each reduction.

- *max_iterations*: integer
  Parameter (default 12)
  Maximum number of iterations when seeking convergence before reducing the time step.

- *max_absolute_difference*: number [**cm**]
  Parameter (default 0.02)
  Maximum absolute difference in 'h' values for convergence.

- *max_relative_difference*: number (dimensionless)
  Parameter (default 0.001)
  Maximum relative difference in 'h' values for convergence.

- *K_average*: **condedge** component (see chapter 20)
  Component (default 'arithmetic')
  Model for calculating average vertical K between cells.

- *max_iterations_timestep_reduction_factor*: integer
  Parameter (default 0)
  Multiply 'max_iterations' with this factor for each timestep reduction.

- *max_number_of_small_time_steps*: integer
  Parameter (default 1000)
  Maximum number of small time steps in a large time step.

- *msg_number_of_small_time_steps*: integer
  Parameter (default 100)
  Number of small time steps in a large time step between message.

- *max_pressure_potential*: number (dimensionless)
  Parameter (default 1e+009)
  Maximum pressure potential for convergence.

- *min_pressure_potential*: number (dimensionless)
  Parameter (default -1e+009)
  minimum pressure potential for convergence.

- *forced_T*: number [**dg C**]
  Optional parameter
  Force transport equations to use this water temperature.

Log Variables

- *Theta_error*: number (dimensionless) soil cells
  Water mass balance error per cell.

- *Kedge*: number [**cm/h**] soil edges
  Conductivity between cells. The value logged is the value used for the last small timestep in the previous large timestep.

Chapter 79

vegetation

That green stuff.

```
< component   (description description)
              (cite)
              (EpInterchange 0.6 [<fraction>]) >
```

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *EpInterchange*: number [<**fraction**>]
  Parameter (default 0.6)
  Canopy adsorbtion fraction of unreached potential soil evaporation.

Log Variables

- *LAI*: number [$\mathbf{m^2/m^2}$]
  Total LAI of all crops on this column

- *height*: number [**cm**]
  Max crop height in canopy

- *albedo*: number (dimensionless)
  Another reflection factor

- *cover*: number [$\mathbf{m^2/m^2}$]
  Fraction of soil covered by crops

- *LAIvsH*: plf [$\mathbf{m^2/m^2} \rightarrow \mathbf{cm}$]
  Total canopy LAI below given height

- *HvsLAI*: plf [$\mathbf{cm} \rightarrow \mathbf{m^2/m^2}$]
  Height in which there is a given LAI below in total canopy

- *ACExt_PAR*: number (dimensionless)
  Canopy extinction coefficient of PAR (how fast the light dim as a function of LAI passed)

- *ACRef_PAR*: number (dimensionless)
  Canopy reflection coefficient of PAR

- *ACExt_NIR*: number (dimensionless)
  Canopy extinction coefficient of NIR (how fast the light dim as a function of LAI passed)

- *ACRef_NIR*: number (dimensionless)
  Canopy reflection coefficient of NIR

- *ARExt*: number (dimensionless)
  Radiation Extinction coefficient (like ACExt, but for all radiation, not just light)

- *EpFactorDry*: number (dimensionless)
  Reference to potential evapotranspiration

- *EpFactorWet*: number (dimensionless)
  Reference to potential evapotranspiration

- *interception_capacity*: number [**mm**]
  Canopy water storage capacity

- *shadow_stomata_conductance*: number [**m/s**]
  Field based stomataconductance of shadow leaves.

- *sunlit_stomata_conductance*: number [**m/s**]
  Field based stomataconductance of sunlit leaves.

79.1 crops

Keep track of all crops on the field.
Used by column default Vegetation (see 19.1, page 130) .

- *crops*: **crop** component (see chapter 22) sequence
  Component (default: an empty sequence)
  List of crops growing in the field

- *ForcedLAI*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  By default, the total LAI for the vegetation will be the sum of the simulated LAI for the individual crops. However, you can force the model to use a different values for LAI by setting this attribute. The specified LAI will be distributed among the crops on the field corresponding to their simulated LAI.

  'ForcedLAI' can be useful if you have measured the total LAI on the field, and want to force the model to confirm to the measurements.

  'ForcedDAY' will not affect the LAI for crops that have not yet emerged. If no crops have emerged on the field, it will be ignored.

  `< year LAIvsDAY >`

  – *year*: integer
    Parameter
    Year for which to use forced LAI.

  - *LAIvsDAY*: plf [$\mathbf{m^2/m^2} \rightarrow$ **yday**]
    Optional parameter
    LAI as a function of Julian day.

    The simulated LAI will be used before the first day you specify and after the last specified day. Simulated LAI will also be used whenever 'LAIvsDAY' becomes negative.

79.2 permanent

Permanent (non-crop) vegetation.

- *Height*: number [**cm**]
  Parameter (default 80)
  Permanent height of vegetation.

- *Root*: **RootSystem** fixed component (see section 85.13)
  Submodel (has fully specified default value)
  Root system.

- *Canopy*: **CanopySimple** fixed component (see section 85.15)
  Submodel (has fully specified default value)
  Canopy.

- *root_DM*: number [**Mg DM/ha**]
  Parameter (default 2)
  Permanent root drymatter.

- *N_actual*: number [**g N/m**^{2}]
  Optional state variable
  N uptake until now (default: 'N_demand').

- *LAIvsDAY*: plf [$\mathbf{m^2/m^2} \rightarrow$ **yday**]
  Parameter
  LAI as a function of Julian day. These numbers are used when there are no yearly numbers (YearlyLAI).

- *YearlyLAI*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Yearly LAI measurements.

  < *year LAIvsDAY* >

  - *year*: integer
    Parameter
    Year for which to use yearly LAI measurements.
  - *LAIvsDAY*: plf [$\mathbf{m^2/m^2} \rightarrow$ **yday**]
    Optional parameter
    LAI as a function of Julian day.

    The default LAI will be used before the first day you specify and after the last specified day. Default LAI will also be used whenever 'LAIvsDAY' becomes negative.

- *LAIfactor*: number (dimensionless)
  Parameter (default 1)
  Multiply calculated LAI with this number for quick scaling.

- *N_per_LAI*: number [**kg N/ha/LAI**]
  Parameter (default 10)
  N content as function of LAI.

- *DM_per_LAI*: number [**Mg DM/ha/LAI**]
  Parameter (default 0.5)
  DM as function of LAI.

- *litter_am*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (litter_am "AOM-SLOW"
             "AOM-FAST")
  ```

  Parameter description:
  Litter AOM parameters.

- *Albedo*: number (dimensionless)
  Parameter (default 0.2)
  Reflection factor.

Log Variables

- *N_demand*: number [**g N/m^2**]
  Current potential N content.

- *N_uptake*: number [**g N/m^2/h**]
  Nitrogen uptake this hour.

- *N_litter*: number [**g N/m^2/h**]
  Nitrogen in litter this hour.

79.3 Grass

A 'permanent' model (see 79.2, page 383) defined in 'vegetation.dai'.

79.4 Coniferous

A 'permanent' model (see 79.2, page 383) defined in 'vegetation.dai'.

79.5 Hardwood

A 'permanent' model (see 79.2, page 383) defined in 'vegetation.dai'.

79.6 Bush

A 'permanent' model (see 79.2, page 383) defined in 'vegetation.dai'.

Chapter 80

vernalization

Requirement for a cold period before flowering.

80.1 none

No vernalization.

Used by crop default Vernal (see 22.1, page 143) .

80.2 default

Temperature sum dependent vernalization.

Used by crop Rye Vernal (see 22.15, page 146) , crop Winter Barley Vernal (see 22.22, page 147) , crop Winter Barley; Koge Vernal (see 22.24, page 147) , crop Winter Rape Vernal (see 22.25, page 147) , crop Winter Wheat Vernal (see 22.26, page 147) , crop Grass to grain Vernal (see 22.29, page 147) , crop Ryegrass Vernal (see 22.31, page 147) , crop Wclover Vernal (see 22.32, page 148) , crop Rug Vernal (see 22.47, page 149) , crop Vinterbyg Vernal (see 22.49, page 149) , crop Vinterhvede Vernal (see 22.50, page 149) , crop Froegraes Vernal (see 22.51, page 150) , and crop Vinterraps Vernal (see 22.55, page 150) .

- *DSLim*: number (dimensionless)
  Parameter
  Development stage at vernalization.

- *TaLim*: number [**dg C**]
  Parameter
  Vernalization temperature threshold.

- *TaSum*: number [**dg C d**]
  State variable
  Vernalization temperature-sum requirement.

Chapter 81

volume

A subset of 3D space.

Used by SoilZone @ volume (see 85.10, page 421) , and Irrigation @ event volume (see 85.29, page 447) .

81.1 box

A volume defined by intervals on each axis. By default, the intervals fill the entire axis. You can modify this by setting the parameters.

Used by action irrigate_subsoil volume (see 9.25, page 70) , action table volume (see 9.38, page 79) , action extern_fertigation volume (see 9.52, page 85) , action extern_subsoil volume (see 9.53, page 86) , log select volume (see 39.3, page 210) , select volume_base volume (see 62.6, page 309) , and select flow volume (see 62.12, page 311) .

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

- *bottom*: **bound** component (see chapter 16)
  Component (default 'state')

  ```
  (bottom state (type none))
  ```

  Parameter description:
  Lower boundary on the z-axis.

- *top*: **bound** component (see chapter 16)
  Component (default 'state')

  ```
  (top state (type none))
  ```

  Parameter description:
  Upper boundary on the z-axis.

- *left*: **bound** component (see chapter 16)
  Component (default 'state')

```
(left state (type none))
```

Parameter description:
Lower boundary on the x-axis.

- *right*: **bound** component (see chapter 16)
  Component (default 'state')

  ```
  (right state (type none))
  ```

  Parameter description:
  Upper boundary on the x-axis.

- *front*: **bound** component (see chapter 16)
  Component (default 'state')

  ```
  (front state (type none))
  ```

  Parameter description:
  Lower boundary on the y-axis.

- *back*: **bound** component (see chapter 16)
  Component (default 'state')

  ```
  (back state (type none))
  ```

  Parameter description:
  Upper boundary on the y-axis.

Chapter 82

weather

A 'wsource' is a source of raw weatherdata.

82.1 weather

Weather interface implementation.

- *snow_fraction*: plf [**dg C** → <**fraction**>]
  Parameter (has default value with 2 points)

  ```
  (snow_fraction (-2 1) (2 0))
  ```

  Parameter description:
  Fraction of precipitation that falls as snow as function of air temperature.

Log Variables

- *precipitation*: number [**mm/h**]
  Precipitation this hour.
- *reference_evapotranspiration*: number [**mm/h**]
  Reference evapotranspiration this hour
- *air_temperature*: number [**dg C**]
  Temperature this hour.
- *global_radiation*: number [$\mathbf{W/m^2}$]
  Global radiation this hour.
- *daily_air_temperature*: number [**dg C**]
  Average temperature this day.
- *daily_min_air_temperature*: number [**dg C**]
  Minumum temperature this day.
- *daily_max_air_temperature*: number [**dg C**]
  Maximum temperature this day.
- *daily_global_radiation*: number [$\mathbf{W/m^2}$]
  Average radiation this day.
- *diffuse_radiation*: number [$\mathbf{W/m^2}$]
  Diffuse radiation this hour.

- *daily_extraterrastial_radiation*: number [**W/m^2**]
  Extraterrestrial radiation this day.

- *rain*: number [**mm/h**]
  Rain this hour.

- *snow*: number [**mm/h**]
  Snow this hour.

- *cloudiness*: number [<**fraction**>]
  Fraction of sky covered by clouds [0-1].

- *daily_cloudiness*: number [<**fraction**>]
  Fraction of sky covered by clouds [0-1].

- *vapor_pressure*: number [**Pa**]
  Humidity.

- *air_pressure*: number [**Pa**]
  Air pressure.

- *wind*: number [**m/s**]
  Wind speed.

- *day_length*: number [**h**]
  Number of light hours this day.

- *deposit*: submodel (see section 1.4.7) sequence
  Total atmospheric deposition of nitrogen.

  < *name value* > **Log Variables**

  - *value*: number [**g/cm^2/h**]
    Value for chemical.
  - *name*: string (see section 1.4.5)
    Name of chemical.

82.2 combine

A 'weather' model (see 82.1, page 389) build into Daisy.
Combine multiple weather sources.

- *entry*: submodel (see section 1.4.7) sequence
  List of weather sources.

  ```
  < (source source)
    (begin begin)
    (end end)
    (use use ...)     ; Has default value. >
  ```

  - *source*: **weather** component (see chapter 82)
    Source of weather data.
  - *begin*: **Time** fixed component (see section 85.21)
    Optional submodel
    Use weather data from source after this time. By default, use data from start of source.

  - *end*: **Time** fixed component (see section 85.21)
    Optional submodel
    Use weather data from source until this time. By default, use data until end of source.
  - *use*: string (see section 1.4.5) sequence
    Parameter (has default value with length 1)

    ```
    (use Any)
    ```

    Parameter description:
    List of weather data to use from source. Specify 'Any' to use all present weather data.

- *reserve*: **weather** component (see chapter 82)
  Component (default 'null')
  Reserve weather model to use when no source match.

82.3 base

A 'weather' base model (see 82.1, page 389) build into Daisy.
Weather that does not change during the simulation.

- *GlobRad*: number [**W/m**^{2}]
  Optional parameter
  Global radiation.

- *AirTemp*: number [**dgC**]
  Optional parameter
  Air temperature.

- *T_min*: number [**dgC**]
  Optional parameter
  Minimum air temperature.

- *T_max*: number [**dgC**]
  Optional parameter
  Maximum air temperature.

- *Precip*: number [**mm/h**]
  Optional parameter
  Precipitation.

- *RefEvap*: number [**mm/h**]
  Optional parameter
  Reference evapotranspiration.

- *VapPres*: number [**Pa**]
  Optional parameter
  Vapor pressure.

- *DiffRad*: number [**W/m**^{2}]
  Optional parameter
  Diffuse radiation.

- *RelHum*: number [**fraction**]
  Optional parameter
  Relative humidity.

- *Wind*: number [**m/s**]
  Optional parameter
  Wind speed.

- *Latitude*: number [**dgNorth**]
  Optional parameter
  Location of station (north-south).

- *Longitude*: number [**dgEast**]
  Optional parameter
  Location of station (east-west).

- *Elevation*: number [**m**]
  Optional parameter
  Station altitude over sea level.

- *TimeZone*: number [**dgEast**]
  Optional parameter
  Time zone.

- *ScreenHeight*: number [**m**]
  Optional parameter
  Measurement altitude over ground level.

- *TAverage*: number [**dgC**]
  Optional parameter
  Yearly average temperature.

- *TAmplitude*: number [**dgC**]
  Optional parameter
  Typical temperature variation over the seasons. If you fit the daily average temperature over a year to a sinus curve,this would be the amplitude.

- *MaxTDay*: number [**yday**]
  Optional parameter
  Typical day where the temperature is highest. If you fit the daily average temperature over a year to a sinus curve,this would be maximum point.

- *NH4WetDep*: number [**ppm**]
  Optional parameter
  NH4 concentration in precipitation.

- *NH4DryDep*: number [**kg N/ha/y**]
  Optional parameter
  Dry deposition of NH4.

- *NO3WetDep*: number [**ppm**]
  Optional parameter
  NO3 concentration in precipitation.

- *NO3DryDep*: number [**kg N/ha/y**]
  Optional parameter
  Dry deposition of NO3.

- *Deposition*: number [**kg N/ha/y**]
  Optional parameter
  Total N deposition.

- *DepDry*: number [<**fraction**>]
  Optional parameter
  Fraction of total N deposition that is dry.

- *DepDryNH4*: number [<**fraction**>]
  Optional parameter
  NH4 fraction of dry deposition.

- *DepWetNH4*: number [<**fraction**>]
  Optional parameter
  NH4 fraction of wet deposition.

- *PAverage*: number [**mm**]
  Optional parameter
  Average precipitation. Used for deviding precipitation into dry and wet.

- *Timestep*: number [**h**]
  Optional parameter
  Timestep for weather data.

- *Surface*: string (see section 1.4.5)
  Optional parameter
  Surface type. Either 'reference' for a weather station standard of short grass, or 'field' for measurements directly at the field.

- *Station*: string (see section 1.4.5)
  Optional parameter
  Name of weather station.

- *Note*: string (see section 1.4.5)
  Optional parameter
  Note regarding this set of data.

- *PrecipCorrect*: number (dimensionless) sequence
  Optional parameter
  Correction factors for precipitation. Can contain one or twelve numbers, in the later case the numbers corresponds to months.

- *Begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  Beginning of weather data.

- *End*: **Time** fixed component (see section 85.21)
  Optional submodel
  End of weather data.

- *begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  Only use data after this date.

- *end*: **Time** fixed component (see section 85.21)
  Optional submodel
  Only use data before this date.

82.4 const

A 'base' model (see 82.3, page 391) build into Daisy.

82.5 table

A 'base' model (see 82.3, page 391) build into Daisy.
Read weather data from a file.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

82.6 default

A 'table' model (see 82.5, page 394) build into Daisy.
Read weather data from specific file.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *missing_years*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  How to get data for dates outside the range of the weather file.

The value is a list of maps. Each map consist of two intervals, and indicates that missing data from the first interval should be read from the second interval instead. Each interval consists of two years, the first and last year of that interval.

When the simulation requests weather data from a date outside the range covered by the weather file, the model will look up each member of the list, to see if the year is covered by the first interval. If so, it will use weather data from the same day in the corresponding year in the second interval.

If a given year is covered by multiple intervals in the list, the first one will be used.

< *from to* >

– *from*: submodel (see section 1.4.7)
Submodel (has partially specified default value)

`(from <missing from> <missing to>)`

Parameter description:
Interval of years to map from.

< *from to* >

∗ *from*: integer
Parameter
First year of interval.

∗ *to*: integer
Parameter
First year of interval.

– *to*: submodel (see section 1.4.7)
Submodel (has partially specified default value)

`(to <missing from> <missing to>)`

Parameter description:
Interval of years to map to.

< *from to* >

∗ *from*: integer
Parameter
First year of interval.

∗ *to*: integer
Parameter
First year of interval.

82.7 indirect

A 'weather' model (see 82.1, page 389) build into Daisy.
Delegate to another weather source.

- *source*: **weather** component (see chapter 82)
Use this weather source.

82.8 time

A 'indirect' model (see 82.7, page 395) build into Daisy.
Change time period for weather data.

You can either specify 'offset' directly, or indirectly through 'from' and 'to'. If you chose the later, offset will be calculated January the first in the year specified with 'from' will be mapped to January the first in the year specified with 'to'. Leap years mean that other dates may not be mapped exactly to corresponding dates in other years.

- *offset*: **Timestep** fixed component (see section 85.1)
  Optional submodel
  Time offset. Positive mean move time ahead.

- *from*: integer
  Optional state variable
  Year in weather data.

- *to*: integer
  Optional state variable
  Year in simulation.

82.9 null

A 'weather' model (see 82.1, page 389) build into Daisy.
Weather that does not change during the simulation.
Used by weather combine reserve (see 82.2, page 390) .

Chapter 83

wse

The water stress effect on crop growth.

83.1 none

Water stress has no effect on plant growth.

83.2 full

Water stress has full effect on crop growth. This means that if there is 50% water stress, assimilate production will be cut into half.

83.3 partial

Water stress has partial effect on crop growth.

With this model, there will be full production when there is enough available soil water to cover the potential evapotranspiration, and no production when there is no soil water available. In between production is controled by the 'y_half' parameter.

See SH:REFERENCE for more explanation.

Used by crop Potato; Triada water_stress_effect (see 22.14, page 146) .

- *y_half*: number [<**fraction**>]
  Parameter
  Effect on assimilate production of water stress. This parameter specifies the effect on assimilate production (compared to potential) when the amount of available soil water is enough to cover exactly half the potential evapotranspiration.

Chapter 84

xysource

XY data series.
< component (description *description*)
(cite) >

- *description*: string (see section 1.4.5)
  Optional parameter
  Description of this model or parameterization. The value will appear in the reference manual, and may also appear in some GUI front ends.

- *cite*: string (see section 1.4.5) sequence
  Parameter (default: an empty sequence)
  BibTeX keys that would be relevant for this model or paramterization.

84.1 arithmetic

Read a daisy log, weather or data file. Calculate an x and an y value for each time step, based on the value in the various columns.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *x*: **number** component (see chapter 45)
  Expression for calculating the x value for this source for each row. The expression can refer to the value in a specific column by the tag for that column.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.

  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *y*: **number** component (see chapter 45)
  Expression for calculating the y value for this source for each row. The expression can refer to the value in a specific column by the tag for that column.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the name of the 'x' and 'y' objects.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, data from dwf and dlf files will be drawn with lines, and data from ddf files will be drawn with points.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

84.2 combine

Combine data from multiple time series with a single expression. Data from times series are matched by date.

- *x*: **number** component (see chapter 45)
  Expression for calculating the x value for this source for each row. A row is any date found in any of the member of 'source'. The expression may refer to the value of each source by its title.

- *source*: **source** component (see chapter 65) sequence
  List of sources for data. The style information for the sources is ignored, but the dates, title and value is used as specified by 'expr' to calculate the combined date and value pairs.

- *y*: **number** component (see chapter 45)
  Expression for calculating the y value for this source for each row. A row is any date found in any of the member of 'source'. The expression may refer to the value of each source by its title.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default a combination of the x and y objects.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, let the first source decide.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

84.3 loop

Calculate x and y pairs based on a single variable.

The variable cover an interval from 'begin' to 'end' in fixed steps 'step'. The name of the variable is specified by 'tag'. The x and y expressions may refer to the variable.

- *x*: **number** component (see chapter 45)
  Component (default 'x')
  Expression for calculating the x value.

- *begin*: number [<**user**>]
  Parameter
  Start of interval.

- *end*: number [<**user**>]
  Parameter
  End of interval.

- *tag*: string (see section 1.4.5)
  Parameter (default 'x')
  Name of free variable to calculate the 'x' and 'y' expressions from.

- *y*: **number** component (see chapter 45)
  Expression for calculating the y value.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the name of the 'x' and 'y' objects.

- *with*: string (see section 1.4.5)
  Parameter (default 'lines')
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *step*: number [<**user**>]
  Parameter
  Disretization within interval.

84.4 inline

A list of x, y pairs.

- *points*: plf [<**unknown**> → <**unknown**>]
  Parameter
  List of (x y) pairs.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the name of the 'x' and 'y' objects.

- *with*: string (see section 1.4.5)
  Parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *x_dimension*: string (see section 1.4.5)
  Parameter
  Dimension for x points.

- *y_dimension*: string (see section 1.4.5)
  Parameter
  Dimension for y points.

84.5 merge

Merge multiple xy data series into one.

- *source*: **xysource** component (see chapter 84) sequence
  XY data series to merge.

- *title*: string (see section 1.4.5)
  Parameter
  Name of data series for the legend on the graph.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, let the first source decide.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *x_dimension*: string (see section 1.4.5)
  Parameter
  Dimension for x points.

- *y_dimension*: string (see section 1.4.5)
  Parameter
  Dimension for y points.

84.6 flux

Read a daisy 2d log file, extract flux through line. Values between 'begin' and 'end' will be accumulated, unless 'when' is set.

- *file*: string (see section 1.4.5)
  Parameter
  Name of Daisy log file where data is found.

- *x*: number [**cm**]
  Optional parameter
  Plot flux through this position.

- *dimension*: string (see section 1.4.5)
  Optional parameter
  Dimension for data. By default, use dimension from file.

- *missing*: string (see section 1.4.5) sequence
  Parameter (has default value with length 2)

  ```
  (missing "" "00.00")
  ```

  Parameter description:
  List of strings indicating missing values.

- *filter*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Only include data from rows that passes all these filters.

  < *tag allowed...* >

  - *tag*: string (see section 1.4.5)
    Parameter
    Name of column in Daisy log file to filter for.
  - *allowed*: string (see section 1.4.5) sequence
    Parameter
    List of allowable values in filter.

- *original*: string (see section 1.4.5) sequence
  Optional parameter
  List of dimensions of the data in the data file.

  If the list has only one element, that element is used as the dimension for all columns in the file. Otherwise, the list must have one element for each column.

  By default Daisy will use the names specified in data file.

- *dim_line*: boolean (see section 1.4.2)
  Optional parameter
  If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

- *z*: number [**cm**]
  Optional parameter
  Plot flux through this depth.

- *begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  Ignore values before this time.

- *end*: **Time** fixed component (see section 85.21)
  Optional submodel
  Ignore values after this time.

- *when*: **Time** fixed component (see section 85.21)
  Optional submodel
  Use value closest to this time.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default the specified 'z' or 'x' value, time, and tag.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, data will be drawn with linespoints.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

- *pos_dim*: string (see section 1.4.5)
  Parameter (default 'cm')
  Dimension for soil position.

84.7 xycombine

Combine data from multiple time series with a single expression. Data from times series are matched by date.

- *source*: **xysource** component (see chapter 84) sequence
  List of sources for data. The style information for the sources is ignored, but the title and value is used as specified by 'expr' to calculate the combined x and y pairs. Either the x or y for all sourcses must be identical.

- *expr*: **number** component (see chapter 45)
  Expression for calculating the value for this source. The expression may refer to the value of each source by its title.

- *title*: string (see section 1.4.5)
  Optional parameter
  Name of data series for the legend on the graph.

  By default use the expression.

- *with*: string (see section 1.4.5)
  Optional parameter
  Specify style used for the data series on the graph. use 'points' to plot each point individually, or 'lines' to draw lines between them.

  By default, let the first source decide.

- *style*: integer
  Optional parameter
  Style to use for this dataset.

  By default, gnuplot will use style 1 for the first source to plot with lines, style 2 for the second, and so forth until it runs out of styles and has to start over. Points work similar, but with its own style counter. For color plots, points and lines with the same style number also have the same color.

  Set style to 0 to reuse the style of the previous series, or to a negative number to explicitly request the default behaviour.

  The 'style' parameter is only used if 'with' is either 'points' or 'lines'.

Chapter 85

Fixed Components

Fixed components are similar to ordinary component, with the exceptions that there can only be one model, that is, only a single implementation of the component, and that it is not possible to define libraries of standard parameterizations for the model.

85.1 Timestep

Relative time.

```
< Timestep  (hours 0)
            (days 0)
            (minutes 0)
            (seconds 0)
            (microseconds 0) >
```

- *hours*: integer
  State variable (default 0)
  Number of hours.

- *days*: integer
  State variable (default 0)
  Number of days.

- *minutes*: integer
  State variable (default 0)
  Number of minutes.

- *seconds*: integer
  State variable (default 0)
  Number of seconds.

- *microseconds*: integer
  State variable (default 0)
  Number of microseconds.

85.2 Weatherdata

Weather data.

```
< Weatherdata  (GlobRad GlobRad)
               (AirTemp AirTemp)
               (T_min T_min)
               (T_max T_max)
               (Precip Precip)
               (RefEvap RefEvap)
               (VapPres VapPres)
               (DiffRad DiffRad)
               (RelHum RelHum)
               (Wind Wind)
               (Latitude Latitude)
               (Longitude Longitude)
               (Elevation Elevation)
               (TimeZone TimeZone)
               (ScreenHeight ScreenHeight)
               (TAverage TAverage)
               (TAmplitude TAmplitude)
               (MaxTDay MaxTDay)
               (NH4WetDep NH4WetDep)
               (NH4DryDep NH4DryDep)
               (NO3WetDep NO3WetDep)
               (NO3DryDep NO3DryDep)
               (Deposition Deposition)
               (DepDry DepDry)
               (DepDryNH4 DepDryNH4)
               (DepWetNH4 DepWetNH4)
               (PAverage PAverage)
               (Timestep Timestep)
               (Surface Surface)
               (Station Station)
               (Note Note)
               (PrecipCorrect PrecipCorrect ...)
               (Begin Begin)
               (End End) >
```

- *GlobRad*: number [**W/m**^{2}]
  Optional parameter
  Global radiation.

- *AirTemp*: number [**dgC**]
  Optional parameter
  Air temperature.

- *T_min*: number [**dgC**]
  Optional parameter
  Minimum air temperature.

- *T_max*: number [**dgC**]
  Optional parameter
  Maximum air temperature.

- *Precip*: number [**mm/h**]
  Optional parameter
  Precipitation.

- *RefEvap*: number [**mm/h**]
  Optional parameter
  Reference evapotranspiration.

- *VapPres*: number [**Pa**]
  Optional parameter
  Vapor pressure.

- *DiffRad*: number [$\mathbf{W/m^2}$]
  Optional parameter
  Diffuse radiation.

- *RelHum*: number [**fraction**]
  Optional parameter
  Relative humidity.

- *Wind*: number [**m/s**]
  Optional parameter
  Wind speed.

- *Latitude*: number [**dgNorth**]
  Optional parameter
  Location of station (north-south).

- *Longitude*: number [**dgEast**]
  Optional parameter
  Location of station (east-west).

- *Elevation*: number [**m**]
  Optional parameter
  Station altitude over sea level.

- *TimeZone*: number [**dgEast**]
  Optional parameter
  Time zone.

- *ScreenHeight*: number [**m**]
  Optional parameter
  Measurement altitude over ground level.

- *TAverage*: number [**dgC**]
  Optional parameter
  Yearly average temperature.

- *TAmplitude*: number [**dgC**]
  Optional parameter
  Typical temperature variation over the seasons. If you fit the daily average temperature over a year to a sinus curve,this would be the amplitude.

- *MaxTDay*: number [**yday**]
  Optional parameter
  Typical day where the temperature is highest. If you fit the daily average temperature over a year to a sinus curve,this would be maximum point.

- *NH4WetDep*: number [**ppm**]
  Optional parameter
  NH4 concentration in precipitation.

- *NH4DryDep*: number [**kg N/ha/y**]
  Optional parameter
  Dry deposition of NH4.
- *NO3WetDep*: number [**ppm**]
  Optional parameter
  NO3 concentration in precipitation.
- *NO3DryDep*: number [**kg N/ha/y**]
  Optional parameter
  Dry deposition of NO3.
- *Deposition*: number [**kg N/ha/y**]
  Optional parameter
  Total N deposition.
- *DepDry*: number [<**fraction**>]
  Optional parameter
  Fraction of total N deposition that is dry.
- *DepDryNH4*: number [<**fraction**>]
  Optional parameter
  NH4 fraction of dry deposition.
- *DepWetNH4*: number [<**fraction**>]
  Optional parameter
  NH4 fraction of wet deposition.
- *PAverage*: number [**mm**]
  Optional parameter
  Average precipitation. Used for deviding precipitation into dry and wet.
- *Timestep*: number [**h**]
  Optional parameter
  Timestep for weather data.
- *Surface*: string (see section 1.4.5)
  Optional parameter
  Surface type. Either 'reference' for a weather station standard of short grass, or 'field' for measurements directly at the field.
- *Station*: string (see section 1.4.5)
  Optional parameter
  Name of weather station.
- *Note*: string (see section 1.4.5)
  Optional parameter
  Note regarding this set of data.
- *PrecipCorrect*: number (dimensionless) sequence
  Optional parameter
  Correction factors for precipitation. Can contain one or twelve numbers, in the later case the numbers corresponds to months.
- *Begin*: **Time** fixed component (see section 85.21)
  Optional submodel
  Beginning of weather data.
- *End*: **Time** fixed component (see section 85.21)
  Optional submodel
  End of weather data.

85.3 Bioincorporation

Biological incorporation of organic matter in soil.

```
< Bioincorporation  (AOM AOM ...)                    ; Has default value.
                    (R_max 0.5 [g DM/m^2/h])
                    (k_half 1 [g DM/m^2])
                    (C_per_N_factor C_per_N_factor)  ; Has default value.
                    (T_factor T_factor)              ; Has default value.
                    (respiration 0.5 [<fraction>])
                    (distribution distribution)      ; Has default value. >
```

- *AOM*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (AOM "AOM-SLOW-BIOINCORPORATION"
       "AOM-FAST")
  ```

  Parameter description:
  Incorporated AM parameters.

- *R_max*: number [**g DM/m^2/h**]
  Parameter (default 0.5)
  Maximal speed of incorporation.

- *k_half*: number [**g DM/m^2**]
  Parameter (default 1)
  Halflife constant.

- *C_per_N_factor*: plf [**(g C/cm^2)/(g N/cm^2)** → <**none**>]
  Parameter (has default value with 3 points)

  ```
  (C_per_N_factor (50 1) (100 0.1) (120 0.01))
  ```

  Parameter description:
  Limiting factor for high C/N ratio.

- *T_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (T_factor (4 0) (6 1))
  ```

  Parameter description:
  Limiting factor for low temperature.

- *respiration*: number [<**fraction**>]
  Parameter (default 0.5)
  Fraction of C lost in respiration.

- *distribution*: plf [**cm** → <**none**>]
  Parameter (has default value with 3 points)

  ```
  (distribution (-80 0) (-18 100) (0 100))
  ```

  Parameter description:
  Distribution of incorporated matter in the soil. (X, Y), where X is the depth (negative numbers), and Y is the relative weight in that depth. To get the fraction in a specific interval [a:b], we integrate the plf over that interval, and divide by the integration over the whole profile.

Log Variables

- *speed*: number [**g DM/m^2/h**]
  Fraction of litter incorporated this hour. The formula is speed = (R_max * litter) / (k_half + litter).
- *DM*: number [**g DM/m^2/h**]
  DM removed from surface.
- *C_removed*: number [**g C/m^2/h**]
  C removed from surface.
- *N_removed*: number [**g N/m^2/h**]
  N removed from surface.
- *CO2*: number [**g C/m^2/h**]
  C respirated.
- *C_added*: number [**g C/cm^3/h**] soil cells
  C added to soil.
- *N_added*: number [**g N/cm^3/h**] soil cells
  N added to soil.

85.4 Harvest

Log of all harvests during the simulation.

```
< Harvest  (time time)                                  ; Has partial value.
           (crop crop)
           (column column)
           (water_stress_days water_stress_days)
           (nitrogen_stress_days nitrogen_stress_days)
           (stem_DM stem_DM)
           (stem_N stem_N)
           (stem_C stem_C)
           (dead_DM dead_DM)
           (dead_N dead_N)
           (dead_C dead_C)
           (leaf_DM leaf_DM)
           (leaf_N leaf_N)
           (leaf_C leaf_C)
           (sorg_DM sorg_DM)
           (sorg_N sorg_N)
           (sorg_C sorg_C)
           (water_productivity water_productivity) >
```

- *time*: **Time** fixed component (see section 85.21)
  Submodel (has partially specified default value)
  Time of the harvest operation.
- *crop*: string (see section 1.4.5)
  State variable
  Name of crop that was harvested.
- *column*: string (see section 1.4.5)
  State variable
  Name of column where the yield were harvested.

- *water_stress_days*: number [**d**]
  State variable
  Production days lost due to water stress.
- *nitrogen_stress_days*: number [**d**]
  State variable
  Production days lost due to water stress.
- *stem_DM*: number [$\mathbf{g/m^2}$]
  State variable
  Total stem dry matter in harvest.
- *stem_N*: number [$\mathbf{g/m^2}$]
  State variable
  Total stem nitrogen in harvest.
- *stem_C*: number [$\mathbf{g/m^2}$]
  State variable
  Total stem carbon in harvest.
- *dead_DM*: number [$\mathbf{g/m^2}$]
  State variable
  Total dead leaf dry matter in harvest.
- *dead_N*: number [$\mathbf{g/m^2}$]
  State variable
  Total dead leaf nitrogen in harvest.
- *dead_C*: number [$\mathbf{g/m^2}$]
  State variable
  Total dead leaf carbon in harvest.
- *leaf_DM*: number [$\mathbf{g/m^2}$]
  State variable
  Total leaf dry matter in harvest.
- *leaf_N*: number [$\mathbf{g/m^2}$]
  State variable
  Total leaf nitrogen in harvest.
- *leaf_C*: number [$\mathbf{g/m^2}$]
  State variable
  Total leaf carbon in harvest.
- *sorg_DM*: number [$\mathbf{g/m^2}$]
  State variable
  Total storage organ dry matter in harvest.
- *sorg_N*: number [$\mathbf{g/m^2}$]
  State variable
  Total storage organ nitrogen in harvest.
- *sorg_C*: number [$\mathbf{g/m^2}$]
  State variable
  Total storage organ carbon in harvest.
- *water_productivity*: number [$\mathbf{kg\ DM/m^3\ H2O}$]
  State variable
  Storage organ harvested per evapotranspiration.

85.5 Snow

Simulate snow pack on surface. _Snow Hydrology_, U.S. Corps of Engineers, 1956.

```
< Snow  (Ssnow 0 [mm])
        (Swater 0 [mm])
        (age 0 [h])
        (dZs 0 [m])
        (mf 10 [m^-1])
        (mtprime 0.0833333 [kg/m^2/h C])
        (mrprime 1.5e-007 [kg/J])
        (m1 2 [kg/J])
        (m2 0.00416667 [h^-1])
        (rho_s 100 [kg/m^3])
        (f_c 0.07 [])
        (rho_1 200 [kg/m^3])
        (rho_2 0.5 [m^-1])
        (Psa 0.208333 [mm])
        (fsa 0.9 [])
        (K_snow_factor 2.86e-006 [W m^5/kg^2/dg C]) >
```

- *Ssnow*: number [**mm**]
  State variable (default 0)
  Snow storage expressed as water.

- *Swater*: number [**mm**]
  State variable (default 0)
  Water in snow storage.

- *age*: number [**h**]
  State variable (default 0)
  Time since last snow.

- *dZs*: number [**m**]
  State variable (default 0)
  Depth of snow layer.

- *mf*: number [$\mathbf{m^{-1}}$]
  Parameter (default 10)
  Snow pack depth melting factor.

- *mtprime*: number [$\mathbf{kg/m^2/h\ C}$]
  Parameter (default 0.0833333)
  Air temperature melting factor.

- *mrprime*: number [**kg/J**]
  Parameter (default 1.5e-007)
  Radiation melting factor.

- *m1*: number [**kg/J**]
  Parameter (default 2)
  Radiation melting linear.

- *m2*: number [$\mathbf{h^{-1}}$]
  Parameter (default 0.00416667)
  Radiation melting exponential factor.

- *rho_s*: number [**kg/m**^{3}]
  Parameter (default 100)
  Density of newly fallen snow.

- *f_c*: number (dimensionless)
  Parameter (default 0.07)
  Water capacity in snow factor.

- *rho_1*: number [**kg/m**^{3}]
  Parameter (default 200)
  Water collapse factor.

- *rho_2*: number [**m**$^{-1}$]
  Parameter (default 0.5)
  Snow collapse factor.

- *Psa*: number [**mm**]
  Parameter (default 0.208333)
  Absolute amount of snow required for snow to become new.

- *fsa*: number (dimensionless)
  Parameter (default 0.9)
  Relative amount of snow required for snow to become new.

- *K_snow_factor*: number [**W m**^{5}**/kg**^{2}**/dg C**]
  Parameter (default 2.86e-006)
  Factor related to thermal conductivity for snow water mix.

Log Variables

- *EvapSnowPack*: number [**mm/h**]
  Evaporation from snowpack.

- *q_s*: number [**mm/h**]
  Leaking water.

85.6 SoilHeat

Temperature and heat flux in soil.

```
< SoilHeat  (S S)
            (T T ...)
            (initial_T initial_T ...)
            (h_frozen -15000 [cm^-1])
            (enable_ice false)
            (T_top T_top) >
```

- *S*: number [**erg/cm**^{3}**/h**]
  Optional state variable
  External heat source, by default zero.

- *T*: number [**dg C**] soil cells
  Optional state variable
  Soil temperature.

- *initial_T*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'T' parameter. The initial value is given as a sequence of

(END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

< *end value* >

- *end*: number [**cm**]
  Parameter
  End point of this layer (a negative number).
- *value*: number [**dg C**]
  Parameter
  Soil temperature.

- *h_frozen*: number [$\mathbf{cm}^{-1}$]
  Parameter (default -15000)
  Pressure below which no more water will freeze.
- *enable_ice*: boolean (see section 1.4.2)
  Parameter (default false)
  Disable this to prevent water from freezing.
- *T_top*: number [**dg C**]
  Optional state variable
  Surface temperature at previous time step.

Log Variables

- *state*: number (dimension not specified) soil cells
  Current freezing/melting state.
- *q*: number [$\mathbf{erg/cm^2/h}$] soil edges
  Heat flux.
- *conductivity*: number [**erg/cm/dg C/h**]
  Heat conductivity.
- *capacity*: number [$\mathbf{erg/cm^3/dg\ C}$]
  Heat capacity.
- *T_freezing*: number [**dg C**] soil cells
  Freezing point depression for freezing.
- *T_thawing*: number [**dg C**] soil cells
  Freezing point depression for thawing.

85.7 SoilWater

Keep track of water and pressure in the soil matrix.

```
< SoilWater  (h h ...)
             (Theta Theta ...)
             (S_permanent)
             (max_exfiltration_gradient max_exfiltration_gradient)
             (max_sink_change 0.1 [])
             (initial_h initial_h ...)
             (initial_Theta initial_Theta ...)
             (X_ice X_ice ...)
             (X_ice_buffer X_ice_buffer ...) >
```

- *h*: number [**cm**] soil cells
  Optional state variable
  Soil water pressure.

- *Theta*: number [<**fraction**>] soil cells
  Optional state variable
  Soil water content.

- *S_permanent*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  State variable (default: an empty sequence)
  Permanent water sink, e.g. subsoil irrigation.

- *max_exfiltration_gradient*: number [**cm/cm**]
  Optional parameter
  Maximal pressure gradient for calculating exfiltration. The gradient is assumed from center of top node to surface of top node. By default, there is no maximum.

- *max_sink_change*: number (dimensionless)
  Parameter (default 0.1)
  Largest change to available water within a timestep. This is used for calculating the suggested timestep. The suggested timestep will be small enough that the change water due to forward calculated sinks (S_forward) alone is less than the specified value.

  Plant available water is defined as the difference between saturation and wilting point.

- *initial_h*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'h' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [**cm**]
    Parameter
    Soil water pressure.

- *initial_Theta*: submodel (see section 1.4.7) sequence
  Optional submodel
  Initial value of the 'Theta' parameter. The initial value is given as a sequence of (END VALUE) pairs, starting from the top and going down. The parameter will be initialized to VALUE from the END of the previous layer, to the END of the current layer.

  < *end value* >

  - *end*: number [**cm**]
    Parameter
    End point of this layer (a negative number).
  - *value*: number [<**fraction**>]
    Parameter
    Soil water content.

- *X_ice*: number [<**fraction**>] soil cells
  Optional state variable
  Ice volume fraction in soil.

- *X_ice_buffer*: number (dimensionless) soil cells
  Optional state variable
  Ice volume that didn't fit the soil durin freezing.

Log Variables

- K: number [**cm/h**] soil cells
  Hydraulic conductivity.

- *tillage*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Changes in water content due to tillage operations.

- *dt*: number [**h**]
  Suggested timestep length (based on S_forward). The absolute value is used, negative numbers indicate source based limits.

- *S_drain*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Water sink due to soil drainage.

- *S_root*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Water sink due to root uptake.

- *sink_cell*: integer
  Cell with largest forward sink compared to available water.

- *Theta_primary*: number [$\mathbf{cm^3/cm^3}$] soil cells
  Water content in primary matrix system. Conventionally, this is the intra-aggregate pores.

- *Theta_secondary*: number [$\mathbf{cm^3/cm^3}$] soil cells
  Water content in secondary matrix system. Conventionally, this is the inter-aggregate pores.

- *S_sum*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Total water sink (due to root uptake and macropores).

- *S_incorp*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Incorporated water sink, typically from subsoil irrigation.

- *S_p*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Water sink (due to macropores).

- *S_ice*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Ice sink due to thawing or freezing.

- *S_ice_water*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Water sink due to thawing or freezing.

- *S_forward_total*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Sink (including source terms) at beginning of timestep. Used for limiting size of timestep. Currently this includes drain and tertiary domain (biopores).

- *S_forward_sink*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Sink (excluding source terms) at beginning of timestep. Used for limiting size of timestep. Currently this includes drain and tertiary domain (biopores).

- *h_ice*: number (dimensionless) soil cells
  Pressure at which all air is out of the matrix. When there are no ice, this is 0.0. When there are ice, the ice is presummed to occupy the large pores, so it is h (Theta_sat - X_ice).

- *q*: number [**cm/h**] soil edges
  Matrix water flux (positive numbers mean upward).

- *q_primary*: number [**cm/h**] soil edges
  Primary domain water flux (positive numbers mean upward).

- *q_secondary*: number [**cm/h**] soil edges
  Secondary domain water flux (positive numbers mean upward).

- *q_p*: number [**cm/h**] soil edges
  Water flux in macro pores (positive numbers mean upward). Only the surface flux is accurate for models with tertiary storage. Use S_p instead, which is always accurate.

- *table_low*: number [**cm**]
  Groundwater table estimated by presure in lowest unsaturated cell. If there are multiple unsaturated cells in the same depth, the one with the lowest pressure will be used.

- *table_high*: number [**cm**]
  Groundwater table estimated by presure in highest saturated cell. If there are multiple saturated cells in the same depth, the one with the highest pressure will be used.

85.8 Surface

Keep track of things on the soil surface.

```
< Surface  (ridge ridge)
           (EpFactor 0.6 [])
           (albedo_dry 0.15 [])
           (albedo_wet 0.08 [])
           (forced_pressure forced_pressure)
           (forced_flux forced_flux)
           (pond_section pond_section ...)
           (DetentionCapacity 1000 [mm])
           (ReservoirConstant 1 [h^-1])
           (LocalDetentionCapacity 10 [mm])
           (z_mixing 0.1 [cm])
           (R_mixing 1e+009 [h/mm]) >
```

- *ridge*: **Ridge** fixed component (see section 85.12)
  Optional submodel
  Active ridge system, if any.

- *EpFactor*: number (dimensionless)
  Parameter (default 0.6)
  Value description: See figure 4 in the cited paper.

  The autumn value can be lower, due to muching. With a crop factor of 1.2 a combined Kc of 1.15 is reached at LAI=5. See also [Kjaersgaard et al., 2008] Convertion of reference evapotranspiration to potential evaporation for bare soil.

- *albedo_dry*: number (dimensionless)
  Parameter (default 0.15)
  Albedo of dry soil (pF >= 3)

- *albedo_wet*: number (dimensionless)
  Parameter (default 0.08)
  Albedo of wet soil (pf <= 1.7)

- *forced_pressure*: number [**mm**]
  Optional parameter
  Set this to force a permanent pressure top.

- *forced_flux*: number [**mm/h**]
  Optional parameter
  Set this to force a permanent flux top. Positive upwards (exfiltration).

- *pond_section*: number [**mm**] sequence
  Optional state variable
  Amount of ponding on each section of the surface. By default, there will be no ponding. In an 1D simulation, there will only be one section. In general, there will be a section for each numeric cell in the soil matrix with an edge towards the surface.

- *DetentionCapacity*: number [**mm**]
  State variable (default 1000)
  Amount of ponding the surface can retain. If ponding in any part of the surface is above this, exceed will runoff.

- *ReservoirConstant*: number [$\mathbf{h}^{-1}$]
  Parameter (default 1)
  Fraction of ponding above DetentionCapacity that runoffs each hour.

- *LocalDetentionCapacity*: number [**mm**]
  State variable (default 10)
  Amount of ponding the surface can retain locally. If ponding in any part of the surface is above this, exceed will be distributed to the rest of the surface.

- *z_mixing*: number [**cm**]
  Parameter (default 0.1)
  Depth of mixing layer in the top of the soil. The mixing layer affect exchange between soil coloids, soil water and the surface, especially in connection with intense rainfall.

- *R_mixing*: number [**h/mm**]
  Parameter (default 1e+009)
  Resistance to mixing inorganic compounds between soil and ponding.

Log Variables

- *T*: number [**dg C**]
  Temperature of water or air directly above the surface.

- *pond*: number [**mm**]
  Amount of ponding on the surface. Negative numbers indicate soil exfiltration.

- *EvapSoilSurface*: number [**mm/h**]
  Water evaporated from the surface, including the pond and exfiltration.

- *Eps*: number [**mm/h**]
  Potential evaporation from the surface.

- *runoff*: number [**mm/h**]
  Amount of water runoff from ponding this hour.

85.9 SoilLayer

A location and content of a soil layer. The layers apply to the soil section not covered by the 'zones' parameter.

Used by Soil @ horizons (see 85.11, page 421) .

< `SoilLayer` *`end horizon`* >

- *end*: number [**cm**]
  Parameter
  End point of this layer (a negative number).

- *horizon*: **horizon** component (see chapter 35)
  Soil properties of this layer.

85.10 SoilZone

A location and content of a soil zone. If several zones cover the same soil, the first one listed is used. If no zones cover the soil, the 'horizons' parameter is used.

With regard to the numeric discretization, the whole cell is assumed to be of the soil found in the cell center.

Used by Soil @ zones (see 85.11, page 421) .

< `SoilZone` *`volume horizon`* >

- *volume*: **volume** component (see chapter 81)
  Volume covered by this zone.

- *horizon*: **horizon** component (see chapter 35)
  Soil properties of this zone.

85.11 Soil

The soil submodel provides the numeric and physical properties of the soil.

```
< Soil  (horizons horizons ...)
        (zones)
        (MaxRootingDepth MaxRootingDepth)
        (dispersivity 5 [cm])
        (dispersivity_transversal dispersivity_transversal)
        (border -100) >
```

- *horizons*: **SoilLayer** fixed component (see section 85.9) sequence
  Layered description of the soil properties. The horizons can be overlapped by the 'zones' parameter. Some groundwater models, specifically 'pipe', may cause an extra horizon to be added below the one specified here if you do not also specify an explicit geometry.

- *zones*: **SoilZone** fixed component (see section 85.10) sequence
  Submodel (default: an empty sequence)
  Zones with special soil properties. This overrules the 'horizons' paramter.

- *MaxRootingDepth*: number [**cm**]
  Parameter
  Depth at the end of the root zone (a positive number).

- *dispersivity*: number [**cm**]
  Parameter (default 5)
  Dispersion length.

- *dispersivity_transversal*: number [**cm**]
  Optional parameter
  Transversal dispersion length. By default, this is 0.1 times the dispersivity.

- *border*: number [**cm**] sequence
  Parameter (has default value with length 1)

  ```
  (border -100 [cm])
  ```

  Parameter description:
  List of flux depths where a mass balance should be possible when logging. This attribute is ignored if the geometry is specified explicitly.

85.12 Ridge

Surface model after ridging.
Used by Surface @ ridge (see 85.8, page 419) .

```
< Ridge  (z z)
         (R_crust R_crust)
         (switch 0.333333 [<fraction>]) >
```

- *z*: plf [<**fraction**> → **cm**]
  Parameter
  The basic ridge parameter is the height, formulated as z (x), where x is the relative distance from the middle of the ridge. x = 0.0 is in the middle of a ridge, while x = 1.0 is at the maximal distance. The ridge is assumed to be symmetric. z (x) is measured in centimeter above the unridged soil surface, which means it is in the same reference system as the rest of the model.

- *R_crust*: number [**h**]
  Parameter
  Resistance in crust.

- *switch*: number [<**fraction**>]
  Parameter (default 0.333333)
  Fraction of ridge height where we switch from bottom regime to wall regime.

Log Variables

- *h*: number [**cm**]
  Soil water pressure.

- *Theta*: number [$\mathbf{cm}^3/\mathbf{cm}^3$]
  Soil water content.

- *Theta_pre*: number [$\mathbf{cm}^3/\mathbf{cm}^3$]
  Soil water content before transport.

- *z_pond*: number [**cm**]
  Internal free water height.

- *x_pond*: number []
  Water to soil point.

- *internal_ponding*: number [**cm**]
  Distance from ridge bottom to water surface.

- *R_bottom*: number [**h**]
  Resistance in ridge bottom.

- *R_wall*: number [**h**]
  Resistance in ridge wall.

- *I_bottom*: number [**cm/h**]
  Infiltration through ridge bottom.

- *I_wall*: number [**cm/h**]
  Infiltration through ridge wall.

- *I*: number [**cm/h**]
  Total infiltration.

85.13 RootSystem

Standard root system model.

```
< RootSystem  (rootdens rootdens)
              (ABAprod ABAprod)                    ; Default none value.
              (DptEmr 10 [cm])
              (PenPar1 0.25 [cm/dg C/d])
              (PenPar2 4 [dg C])
              (PenClayFac PenClayFac)              ; Has default value.
              (MaxPen 100 [cm])
              (MaxWidth MaxWidth)
              (Rad 0.005 [cm])
              (h_wp -15000 [cm])
              (MxNH4Up 2.5e-007 [g/cm/h])
              (MxNO3Up 2.5e-007 [g/cm/h])
              (Rxylem 10 [])
              (PotRtDpt PotRtDpt)
              (Depth Depth)
              (ABAConc 0 [g/cm^3])
              (h_x 0 [cm])
              (partial_soil_temperature 0 [dg C h])
              (partial_day 0 [h])
              (soil_temperature 0 [dg C])
              (water_stress_days 0 [d]) >
```

- *rootdens*: **rootdens** component (see chapter 56)
  Optional component
  Root density model.

- *ABAprod*: **ABAproduction** component (see chapter 2)
  Component (default 'none')
  ABA production model.

- *DptEmr*: number [**cm**]
  Parameter (default 10)
  Penetration at emergence.

- *PenPar1*: number [**cm/dg C/d**]
  Parameter (default 0.25)
  Penetration rate parameter, coefficient.

- *PenPar2*: number [**dg C**]
  Parameter (default 4)
  Penetration rate parameter, threshold.

- *PenClayFac*: plf [<**fraction**> → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (PenClayFac (0 1) (1 1))
  ```

  Parameter description:
  Clay dependent factor to multiply 'PenPar1' with.

- *MaxPen*: number [**cm**]
  Parameter (default 100)
  Maximum penetration depth.

- *MaxWidth*: number [**cm**]
  Optional parameter
  Maximum horizontal distance of roots from plant.

- *Rad*: number [**cm**]
  Parameter (default 0.005)
  Root radius.

- *h_wp*: number [**cm**]
  Parameter (default -15000)
  Matrix potential at wilting point.

- *MxNH4Up*: number [**g/cm/h**]
  Parameter (default 2.5e-007)
  Maximum NH4 uptake per unit root length.

- *MxNO3Up*: number [**g/cm/h**]
  Parameter (default 2.5e-007)
  Maximum NO3 uptake per unit root length.

- *Rxylem*: number (dimensionless)
  Parameter (default 10)
  Transport resistence in xyleme.

- *PotRtDpt*: number [**cm**]
  Optional state variable
  Potential root penetration depth.

- *Depth*: number [**cm**]
  Optional state variable
  Rooting Depth.

- *ABAConc*: number [$\mathbf{g/cm^3}$]
  State variable (default 0)
  ABA concentration in water uptake.

- *h_x*: number [**cm**]
  State variable (default 0)
  Root extraction at surface.

- *partial_soil_temperature*: number [**dg C h**]
  State variable (default 0)
  Soil temperature hours this day, so far.

- *partial_day*: number [**h**]
  State variable (default 0)
  Hours we have accumulated soil temperature this day.

- *soil_temperature*: number [**dg C**]
  State variable (default 0)
  Average soil temperature yesterday.

- *water_stress_days*: number [**d**]
  State variable (default 0)
  Number of days production has halted due to water stress. This is the sum of water stress for each hour, multiplied with the fraction of the radition of that day that was received that hour.

Log Variables

- *production_stress*: number (dimensionless)
  SVAT induced stress, or -1 if not applicable.

- *Density*: number [$\mathbf{cm/cm^3}$] soil cells
  Root density in soil layers.

- *H2OExtraction*: number [$\mathbf{cm^3/cm^3/h}$] soil cells
  Extraction of H2O in soil layers.

- *NH4Extraction*: number [$\mathbf{g\ N/cm^3/h}$] soil cells
  Extraction of NH4-N in soil layers.

- *NO3Extraction*: number [$\mathbf{g\ N/cm^3/h}$] soil cells
  Extraction of NO3-N in soil layers.

- *ABAExtraction*: number [$\mathbf{g/cm^3/h}$] soil cells
  Extraction of ABA in soil layers.

- *water_stress*: number (dimensionless)
  Fraction of requested water we didn't get.

- *Ept*: number [**mm/h**]
  Potential transpiration.

- *H2OUpt*: number [**mm/h**]
  H2O uptake.

- *NH4Upt*: number [$\mathbf{g\ N/m^2/h}$]
  NH4-N uptake.

- *NO3Upt*: number [$\mathbf{g\ N/m^2/h}$]
  NO3-N uptake.

85.14 CanopyStandard

Standard canopy model.

```
< CanopyStandard  (Height 0 [cm])
                  (PARref 0.06 [])
                  (PARext 0.6 [])
                  (NIRref 0.51 [])
                  (NIRext 0.18 [])
                  (EPext 0.5 [])
                  (IntcpCap 0.5 [mm])
                  (EpFac 1.2 [])
                  (EpFacWet EpFacWet)
                  (EpFacDS EpFacDS)                ; Has default value.
                  (rs_max 100000 [s/m])
                  (rs_min 100 [s/m])
                  (leaf_width leaf_width)          ; Has default value.
                  (SpLAI SpLAI)
                  (LeafAIMod LeafAIMod)            ; Has default value.
                  (SpSOrgAI 0 [(m^2/m^2)/(g DM/m^2)])
                  (SOrgAIMod SOrgAIMod)            ; Has default value.
                  (SOrgPhotEff 1 [])
                  (SpStemAI 0 [(m^2/m^2)/(g DM/m^2)])
                  (StemAIMod StemAIMod)            ; Has default value.
                  (StemPhotEff 1 [])
                  (HvsDS HvsDS)
                  (HvsWStem HvsWStem)              ; Has default value.
                  (LAIDist0 LAIDist0 ...)
                  (LAIDist1 LAIDist1 ...)
                  (PARrel 0.05 [])
                  (Offset 0 [cm])
                  (LeafAI 0 [m^2/m^2])
                  (StemAI 0 [m^2/m^2])
                  (SOrgAI 0 [m^2/m^2])
                  (LADm -9999.99 [cm^2/cm^3]) >
```

- *Height*: number [**cm**]
  State variable (default 0)
  Crop height.

- *PARref*: number (dimensionless)
  Parameter (default 0.06)
  PAR reflectance.

- *PARext*: number (dimensionless)
  Parameter (default 0.6)
  PAR extinction coefficient.

- *NIRref*: number (dimensionless)
  Parameter (default 0.51)
  NIR reflectance. NIRref = 0.51 (Ross, 1975)

- *NIRext*: number (dimensionless)
  Parameter (default 0.18)
  NIR extinction coefficient. NIRext = 0.18 (Jones, 1983)

- *EPext*: number (dimensionless)
  Parameter (default 0.5)
  EP extinction coefficient.

- *IntcpCap*: number [**mm**]
  Parameter (default 0.5)
  Interception capacity.

- *EpFac*: number (dimensionless)
  Parameter (default 1.2)
  Value description: See figure 4 in the cited paper.

  With a bare soil factor of 0.6 a combined Kc of 1.15 is reached at LAI=5. See also [Kjaersgaard et al., 2008]
  Potential evapotranspiration factor.

- *EpFacWet*: number (dimensionless)
  Optional parameter
  Potential evapotranspiration factor for wet surface. By default this is identical to EpFac.

- *EpFacDS*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (EpFacDS (0 1) (1 1))
  ```

  Parameter description:
  DS dependent potential evapotranspiration factor.

- *rs_max*: number [**s/m**]
  Parameter (default 100000)
  Maximum transpiration resistance.

- *rs_min*: number [**s/m**]
  Parameter (default 100)
  Minimum transpiration resistance.

- *leaf_width*: plf [**DS** → **cm**]
  Parameter (has default value with 2 points)

  ```
  (leaf_width (0 3) (2 3))
  ```

  Parameter description:
  Leaf width.

- *SpLAI*: number [**(m^2/m^2)/(g DM/m^2)**]
  Parameter
  Specific leaf weight.

- *LeafAIMod*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (LeafAIMod (0 1) (2 1))
  ```

  Parameter description:
  Specific leaf weight modifier. Used only after the intital phase.

- *SpSOrgAI*: number [**(m^2/m^2)/(g DM/m^2)**]
  Parameter (default 0)
  Specific storage organ weight. Used only after the intital phase.

- *SOrgAIMod*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (SOrgAIMod (0 1) (2 1))
  ```

  Parameter description:
  Specific storage organ weight modifier. Used only after the intital phase.

- *SOrgPhotEff*: number (dimensionless)
  Parameter (default 1)
  Relative photosynthetic efficiency of storage organ. Used only after the intital phase.

- *SpStemAI*: number [**(m^2/m^2)/(g DM/m^2)**]
  Parameter (default 0)
  Specific stem weight. Used only after the intital phase.

- *StemAIMod*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (StemAIMod (0 1) (2 1))
  ```

  Parameter description:
  Specific stem weight modifier. Used only after the intital phase.

- *StemPhotEff*: number (dimensionless)
  Parameter (default 1)
  Relative photosynthetic efficiency of stem. Used only after the intital phase.

- *HvsDS*: plf [<**none**> → **cm**]
  Parameter
  Crop height as function of DS.

- *HvsWStem*: plf [**g DM/m^2** → <**fraction**>]
  Parameter (has default value with 2 points)

  ```
  (HvsWStem (0 0.1) (200 1))
  ```

  Parameter description:
  Relative crop height as function of stem weight. By default, it needs 200 g DM/m^2 to reach full height.

- *LAIDist0*: number (dimensionless) array of length 3
  Parameter
  Relative CAI distribution at DS=0.

- *LAIDist1*: number (dimensionless) array of length 3
  Parameter
  Relative CAI distribution at DS=1.

- *PARrel*: number (dimensionless)
  Parameter (default 0.05)
  Relative PAR below the canopy. If the relative PAR get below this, the bottom leaves will start dying.

- *Offset*: number [**cm**]
  State variable (default 0)
  Extra height after harvest.

- *LeafAI*: number [$\mathbf{m^2/m^2}$]
  State variable (default 0)
  Leaf Area Index.

- *StemAI*: number [$\mathbf{m^2/m^2}$]
  State variable (default 0)
  Stem Area Index.

- *SOrgAI*: number [$\mathbf{m^2/m^2}$]
  State variable (default 0)
  Storage Organ Area Index.

- *LADm*: number [$\mathbf{cm^2/cm^3}$]
  State variable (default -9999.99)
  Maximal Leaf Area Density.

Log Variables

- *CAI*: number [$\mathbf{m^2/m^2}$]
  Crop Area Index.

- *LAIvsH*: plf [$\mathbf{cm \rightarrow m^2/m^2}$]
  Accumulated Leaf Area Index at Height.

- *ForcedCAI*: number [$\mathbf{m^2/m^2}$]
  CAI forced upon us by vegetation module.

- *SimCAI*: number [$\mathbf{m^2/m^2}$]
  CAI simulated by crop model.

- *CAImRat*: number (dimensionless)
  (CAIm - CAI) / CAIm.

85.15 CanopySimple

Simple canopy model.

```
< CanopySimple  (Height 0 [cm])
                (PARref 0.06 [])
                (PARext 0.6 [])
                (NIRref 0.51 [])
                (NIRext 0.18 [])
                (EPext 0.5 [])
                (IntcpCap 0.5 [mm])
                (EpFac 1.2 [])
                (EpFacWet EpFacWet)
                (EpFacDS EpFacDS)        ; Has default value.
                (rs_max 100000 [s/m])
                (rs_min 100 [s/m])
                (leaf_width leaf_width)  ; Has default value. >
```

- *Height*: number [**cm**]
  State variable (default 0)
  Crop height.

- *PARref*: number (dimensionless)
  Parameter (default 0.06)
  PAR reflectance.

- *PARext*: number (dimensionless)
  Parameter (default 0.6)
  PAR extinction coefficient.

- *NIRref*: number (dimensionless)
  Parameter (default 0.51)
  NIR reflectance. NIRref = 0.51 (Ross, 1975)

- *NIRext*: number (dimensionless)
  Parameter (default 0.18)
  NIR extinction coefficient. NIRext = 0.18 (Jones, 1983)

- *EPext*: number (dimensionless)
  Parameter (default 0.5)
  EP extinction coefficient.

- *IntcpCap*: number [**mm**]
  Parameter (default 0.5)
  Interception capacity.

- *EpFac*: number (dimensionless)
  Parameter (default 1.2)
  Value description: See figure 4 in the cited paper.

  With a bare soil factor of 0.6 a combined Kc of 1.15 is reached at LAI=5. See also [Kjaersgaard et al., 2008]
  Potential evapotranspiration factor.

- *EpFacWet*: number (dimensionless)
  Optional parameter
  Potential evapotranspiration factor for wet surface. By default this is identical to EpFac.

- *EpFacDS*: plf [**DS** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (EpFacDS (0 1) (1 1))
  ```

  Parameter description:
  DS dependent potential evapotranspiration factor.

- *rs_max*: number [**s/m**]
  Parameter (default 100000)
  Maximum transpiration resistance.

- *rs_min*: number [**s/m**]
  Parameter (default 100)
  Minimum transpiration resistance.

- *leaf_width*: plf [**DS** → **cm**]
  Parameter (has default value with 2 points)

  ```
  (leaf_width (0 3) (2 3))
  ```

  Parameter description:
  Leaf width.

Log Variables

- *CAI*: number [$\mathbf{m^2/m^2}$]
  Crop Area Index.

- *LAIvsH*: plf [$\mathbf{cm \rightarrow m^2/m^2}$]
  Accumulated Leaf Area Index at Height.

85.16 Harvesting

Information about what happens to the crop at harvest and cut.

```
< Harvesting  (Root Root ...)                    ; Has default value.
              (Stem Stem ...)                    ; Has default value.
              (Leaf Leaf ...)                    ; Has default value.
              (Dead Dead ...)                    ; Has default value.
              (SOrg SOrg ...)                    ; Has default value.
              (EconomicYield_W 1 [])
              (EconomicYield_N EconomicYield_N)
              (DSmax 0.8 [])
              (DSnew DSnew)
              (last_cut last_cut)
              (production_delay 0 [d])
              (cut_delay cut_delay)              ; Has default value.
              (total_water_use 0 [kg H2O])
              (sorg_height sorg_height) >
```

- *Root*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (Root "AOM-SLOW"
        "AOM-FAST")
  ```

  Parameter description:
  Root AM parameters.

- *Stem*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (Stem "AOM-SLOW"
        "AOM-FAST")
  ```

  Parameter description:
  Stem AM parameters.

- *Leaf*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (Leaf "AOM-SLOW"
        "AOM-FAST")
  ```

  Parameter description:
  Leaf AM parameters.

- *Dead*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

```
(Dead "AOM-SLOW"
      "AOM-FAST")
```

Parameter description:
Dead leaves AM parameters.

- *SOrg*: **AOM** component (see chapter 3) sequence
  Component (has default value with length 2)

  ```
  (SOrg "AOM-SLOW"
        "AOM-FAST")
  ```

  Parameter description:
  Storage organ AM parameters.

- *EconomicYield_W*: number (dimensionless)
  Parameter (default 1)
  Valuable fraction of storage organ (DM), e.g. grain or tuber.

- *EconomicYield_N*: number (dimensionless)
  Optional parameter
  Valuable fraction of storage organ (N). By default the value for DM is used.

- *DSmax*: number (dimensionless)
  Parameter (default 0.8)
  Maximal development stage for which the crop survives harvest.

- *DSnew*: number (dimensionless)
  Optional parameter
  New development stage after harvest. If not specified, use the DS where an uncut crop would first reach the height it now has after the cut. I.e. it uses the inverse function of the HvsDS Canopy parameter to find the new DS.

- *last_cut*: **Time** fixed component (see section 85.21)
  Optional submodel
  Date of last cut. Used for calculating cut delay.

- *production_delay*: number [**d**]
  State variable (default 0)
  production delay caused by last cut

- *cut_delay*: plf [**kg DM/ha** → **d**]
  Parameter (has default value with 2 points)

  ```
  (cut_delay (0 0) (1 0))
  ```

  Parameter description:
  Production and development delay in days as a function of the shoot DM removed by harvest. By default, there is no delay.

- *total_water_use*: number [**kg H2O**]
  State variable (default 0)
  Total evapotranspiration since emergence.

- *sorg_height*: number [**cm**]
  Optional parameter
  Vertical location of storage organ. Set this to a negative number for root fruits, this will cause harvesting to imply a suitable tillage operation, and guarentee that harvest will kill the plant. By default, the storage organ is assumed to be located far above ground.

Log Variables

- *cut_stress*: number [<**fraction**>]
  Stress induced due to last cut.

85.17 Production

Crop production in the default crop model.

```
< Production  (ShldResC 0 [<fraction>])
              (ReMobilDS 1.2 [])
              (ReMobilRt 0.1 [d^-1])
              (StemRes 0 [g DM/m^2])
              (CH2OReleaseRate 0.04 [h^-1])
              (E_Root 0.69 [])
              (E_Leaf 0.68 [])
              (E_Stem 0.66 [])
              (E_SOrg E_SOrg)
              (r_Root 0.015 [])
              (r_Leaf r_Leaf)
              (r_Stem r_Stem)
              (r_SOrg r_SOrg)
              (ExfoliationFac 1 [])
              (LfDR LfDR)
              (RtDR RtDR)
              (Large_RtDR 0.05 [d^-1])
              (RtDR_T_factor RtDR_T_factor)   ; Has default value.
              (IntDSRelRtRes 0.8 [])
              (EndDSRelRtRes 0.8 [])
              (RelRateRtRes 0.05 [d^-1])
              (LfRtRelRtRes 0.8 [])
              (CH2OPool 0.001 [g CH2O/m^2])
              (WLeaf 0.001 [g DM/m^2])
              (WStem 0 [g DM/m^2])
              (WRoot 0.001 [g DM/m^2])
              (WSOrg 0 [g DM/m^2])
              (WDead 0 [g DM/m^2])
              (NLeaf 0 [g N/m^2])
              (NStem 0 [g N/m^2])
              (NRoot 0 [g N/m^2])
              (NSOrg 0 [g N/m^2])
              (NDead 0 [g N/m^2])
              (NCrop NCrop)
              (C_AM 0 [g C/m^2])
              (N_AM 0 [g N/m^2])
              (DailyNetRoot 0 [g DM/m^2])
              (DailyNetShoot 0 [g DM/m^2]) >
```

- *ShldResC*: number [<**fraction**>]
  Parameter (default 0)
  Capacity of shielded reserves (fraction of stem DM).

- *ReMobilDS*: number (dimensionless)
  Parameter (default 1.2)
  Remobilization, Initial DS.

- *ReMobilRt*: number [$\mathbf{d}^{-1}$]
  Parameter (default 0.1)
  Remobilization, release rate.

- *StemRes*: number [$\mathbf{g\ DM/m}^2$]
  State variable (default 0)
  Shielded reserves in stems.

- *CH2OReleaseRate*: number [$\mathbf{h}^{-1}$]
  Parameter (default 0.04)
  CH2O Release Rate constant.

- *E_Root*: number (dimensionless)
  Parameter (default 0.69)
  Conversion efficiency, root.

- *E_Leaf*: number (dimensionless)
  Parameter (default 0.68)
  Conversion efficiency, leaf.

- *E_Stem*: number (dimensionless)
  Parameter (default 0.66)
  Conversion efficiency, stem.

- *E_SOrg*: number (dimensionless)
  Parameter
  Conversion efficiency, storage organ.

- *r_Root*: number (dimensionless)
  Parameter (default 0.015)
  Maintenance respiration coefficient, root.

- *r_Leaf*: number [$\mathbf{d}^{-1}$]
  Parameter
  Maintenance respiration coefficient, leaf.

- *r_Stem*: number [$\mathbf{d}^{-1}$]
  Parameter
  Maintenance respiration coefficient, stem.

- *r_SOrg*: number [$\mathbf{d}^{-1}$]
  Parameter
  Maintenance respiration coefficient, storage organ.

- *ExfoliationFac*: number (dimensionless)
  Parameter (default 1)
  Exfoliation factor, 0-1.

- *LfDR*: plf [$\mathbf{DS} \rightarrow\ \mathbf{d}^{-1}$]
  Parameter
  Death rate of Leafs.

- *RtDR*: plf [$\mathbf{DS} \rightarrow\ \mathbf{d}^{-1}$]
  Parameter
  Death rate of Roots.

- *Large_RtDR*: number [$\mathbf{d}^{-1}$]
  Parameter (default 0.05)
  Extra death rate for large root/shoot.

- *RtDR_T_factor*: plf [**dg C** → <**none**>]
  Parameter (has default value with 2 points)

  ```
  (RtDR_T_factor (0 1) (100 1))
  ```

  Parameter description:
  Temperature dependent factor for root death rate.

- *IntDSRelRtRes*: number (dimensionless)
  Parameter (default 0.8)
  Initial DS for the release of root reserves.

- *EndDSRelRtRes*: number (dimensionless)
  Parameter (default 0.8)
  End DS for the release of root reserves.

- *RelRateRtRes*: number [$\mathbf{d}^{-1}$]
  Parameter (default 0.05)
  Release rate of root reserves.

- *LfRtRelRtRes*: number (dimensionless)
  Parameter (default 0.8)
  Max Leaf:Root for the release of root res.

- *CH2OPool*: number [$\mathbf{g\ CH2O/m^2}$]
  State variable (default 0.001)
  CH2O Pool.

- *WLeaf*: number [$\mathbf{g\ DM/m^2}$]
  State variable (default 0.001)
  Leaf dry matter weight.

- *WStem*: number [$\mathbf{g\ DM/m^2}$]
  State variable (default 0)
  Stem dry matter weight.

- *WRoot*: number [$\mathbf{g\ DM/m^2}$]
  State variable (default 0.001)
  Root dry matter weight.

- *WSOrg*: number [$\mathbf{g\ DM/m^2}$]
  State variable (default 0)
  Storage organ dry matter weight.

- *WDead*: number [$\mathbf{g\ DM/m^2}$]
  State variable (default 0)
  Dead leaves dry matter weight.

- *NLeaf*: number [$\mathbf{g\ N/m^2}$]
  State variable (default 0)
  Nitrogen stored in the leaves.

- *NStem*: number [$\mathbf{g\ N/m^2}$]
  State variable (default 0)
  Nitrogen stored in the stem.

- *NRoot*: number [$\mathbf{g\ N/m^2}$]
  State variable (default 0)
  Nitrogen stored in the roots.

- *NSOrg*: number [**g N/m^2**]
  State variable (default 0)
  Nitrogen stored in the storage organ.
- *NDead*: number [**g N/m^2**]
  State variable (default 0)
  Nitrogen stored in dead leaves.
- *NCrop*: number [**g N/m^2**]
  Optional state variable
  Total crop nitrogen content. By default, this will start as the amount of N in the seed.
- *C_AM*: number [**g C/m^2**]
  State variable (default 0)
  Added C in plant material.
- *N_AM*: number [**g N/m^2**]
  State variable (default 0)
  Added N in plant material.
- *DailyNetRoot*: number [**g DM/m^2**]
  State variable (default 0)
  Root growth minus root respiration so far this day.
- *DailyNetShoot*: number [**g DM/m^2**]
  State variable (default 0)
  Leaf growth minus leaf respiration so far this day.

Log Variables

- *CLeaf*: number [**g C/m^2**]
  Leaf C weight.
- *CStem*: number [**g C/m^2**]
  Stem C weight.
- *CRoot*: number [**g C/m^2**]
  Root C weight.
- *CSOrg*: number [**g C/m^2**]
  Storage organ C weight.
- *CDead*: number [**g C/m^2**]
  Dead leaves C weight.
- *CCrop*: number [**g C/m^2**]
  Crop C weight.
- *PotCanopyAss*: number [**g CH2O/m^2/h**]
  Potential canopy assimilation, i.e. stressfree production.
- *CanopyAss*: number [**g CH2O/m^2/h**]
  Canopy assimilation.
- *NetPhotosynthesis*: number [**g CO2/m^2/h**]
  Net Photosynthesis.
- *AccNetPhotosynthesis*: number [**g CO2/m^2**]
  Accumulated Net Photosynthesis.

- *Respiration*: number [**g CH2O/m^2/h**]
  Crop Respiration.
- *MaintRespiration*: number [**g CH2O/m^2/h**]
  Maintenance Respiration.
- *GrowthRespiration*: number [**g CH2O/m^2/h**]
  Growth Respiration.
- *LeafRespiration*: number [**g CO2/m^2/h**]
  Total Leaf Respiration.
- *StemRespiration*: number [**g CO2/m^2/h**]
  Total Stem Respiration.
- *SOrgRespiration*: number [**g CO2/m^2/h**]
  Total SOrg Respiration.
- *RootRespiration*: number [**g CO2/m^2/h**]
  Total Root Respiration.
- *LeafMaintRespiration*: number [**g CO2/m^2/h**]
  Leaf Maintenance Respiration.
- *StemMaintRespiration*: number [**g CO2/m^2/h**]
  Stem Maintenance Respiration.
- *SOrgMaintRespiration*: number [**g CO2/m^2/h**]
  SOrg Maintenance Respiration.
- *RootMaintRespiration*: number [**g CO2/m^2/h**]
  Root Maintenance Respiration.
- *LeafGrowthRespiration*: number [**g CO2/m^2/h**]
  Leaf Growth Respiration.
- *StemGrowthRespiration*: number [**g CO2/m^2/h**]
  Stem Growth Respiration.
- *SOrgGrowthRespiration*: number [**g CO2/m^2/h**]
  SOrg Growth Respiration.
- *RootGrowthRespiration*: number [**g CO2/m^2/h**]
  Root Growth Respiration.
- *IncWLeaf*: number [**g DM/m^2/h**]
  Leaf growth.
- *IncWStem*: number [**g DM/m^2/h**]
  Stem growth.
- *IncWSOrg*: number [**g DM/m^2/h**]
  Storage organ growth.
- *IncWRoot*: number [**g DM/m^2/h**]
  Root growth.
- *DeadWLeaf*: number [**g DM/m^2/h**]
  Leaf DM removed.

- *DeadNLeaf*: number [**g N/m2/h**]
  Leaf N removed.

- *DeadWRoot*: number [**g DM/m^2/h**]
  Root DM removed.

- *DeadNRoot*: number [**g N/m2/h**]
  Root N removed.

- *C_Loss*: number [**g C/m^2/h**]
  C lost from the crop

85.18 Partition

Assimilate partitioning in the default crop model. The 'Root' parameter determine what fraction of the assimilate for growth goes to roots at a given development stage. The remaining assimilate goes to the shoot. The 'Leaf' and 'Stem' parameters determine what fraction of the shoot assimilate goes to the leaf and stem respectively. The remaining shoot assimilate will go to the storage organ.

```
< Partition  (Root Root)
             (Stem Stem)
             (Leaf Leaf)
             (RSR RSR)
             (nitrogen_stress_limit 1 []) >
```

- *Root*: plf [**DS** → <**fraction**>]
  Parameter
  Fraction of assimilate for growth that goes to the roots, as a function of the crop development stage. The remaining growth assimilate goes to the shoot.

- *Stem*: plf [**DS** → <**fraction**>]
  Parameter
  Fraction of shoot assimilate that goes to the stem.

- *Leaf*: plf [**DS** → <**fraction**>]
  Parameter
  Fraction of shoot assimilate that goes to the leafs.

- *RSR*: plf [**DS** → <**none**>]
  Parameter
  Maximal root/shoot ratio as a function of development state. If the root/shoot ratio is above this, the roots will start dying.

- *nitrogen_stress_limit*: number (dimensionless)
  Parameter (default 1)
  If nitrogen stress is above this number and DS is above 1, allocate all assimilate to the storage organ.

85.19 CrpN

Default crop nitrogen parameters.

```
< CrpN  (PtLeafCnc PtLeafCnc)
        (CrLeafCnc CrLeafCnc)
        (NfLeafCnc NfLeafCnc)
        (PtStemCnc PtStemCnc)
        (CrStemCnc CrStemCnc)
        (NfStemCnc NfStemCnc)
        (PtSOrgCnc PtSOrgCnc)
        (CrSOrgCnc CrSOrgCnc)
        (NfSOrgCnc NfSOrgCnc)
        (PtRootCnc PtRootCnc)
        (CrRootCnc CrRootCnc)
        (NfRootCnc NfRootCnc)
        (TLLeafEff TLLeafEff)                ; Has default value.
        (TLRootEff TLRootEff)                ; Has default value.
        (NO3_root_min 0 [g N/cm^3])
        (NH4_root_min 0 [g N/cm^3])
        (nitrogen_stress_days 0 [d])
        (DS_fixate 42000 [])
        (DS_cut_fixate 0 [])
        (fixate_factor 0.8 [])
        (DS_start_fixate DS_start_fixate) >
```

- *PtLeafCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Upper limit for N-concentration in leaves.

- *CrLeafCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Critical limit for N-concentration in leaves.

- *NfLeafCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Non-functional limit for N-concentration in leaves.

- *PtStemCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Upper limit for N-concentration in stem.

- *CrStemCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Critical limit for N-concentration in stem.

- *NfStemCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Non-functional limit for N-concentration in stem.

- *PtSOrgCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Upper limit for N-concentration in storage organ.

- *CrSOrgCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Critical limit for N-concentration in storage organ.

- *NfSOrgCnc*: plf [**DS** → **g N/g DM**]
  Parameter
  Non-functional limit for N-concentration in storage organ.

- *PtRootCnc*: plf [**DS → g N/g DM**]
  Parameter
  Upper limit for N-concentration in roots.

- *CrRootCnc*: plf [**DS → g N/g DM**]
  Parameter
  Critical limit for N-concentration in roots.

- *NfRootCnc*: plf [**DS → g N/g DM**]
  Parameter
  Non-functional lim for N-concentration in roots.

- *TLLeafEff*: plf [**DS →** <**fraction**>]
  Parameter (has default value with 2 points)

  ```
  (TLLeafEff (0 0.9) (2 0.9))
  ```

  Parameter description:
  Translocation effiency, Leaf.

- *TLRootEff*: plf [**DS →** <**fraction**>]
  Parameter (has default value with 2 points)

  ```
  (TLRootEff (0 0.1) (2 0.1))
  ```

  Parameter description:
  Translocation effiency, Root.

- *NO3_root_min*: number [**g N/cm**^{3}]
  Parameter (default 0)
  Minimum nitrate concentration near roots for uptake.

- *NH4_root_min*: number [**g N/cm**^{3}]
  Parameter (default 0)
  Minimum ammonium concentration near roots for uptake.

- *nitrogen_stress_days*: number [**d**]
  State variable (default 0)
  Number of days production has halted due to nitrogen stress. This is the sum of nitrogen stress for each hour, multiplied with the action of the radition of that day that was received that hour.

- *DS_fixate*: number (dimensionless)
  Parameter (default 42000)
  DS at which to start fixation of atmospheric N.

- *DS_cut_fixate*: number (dimensionless)
  Parameter (default 0)
  Restore fixation this DS after cut.

- *fixate_factor*: number (dimensionless)
  Parameter (default 0.8)
  Fraction of needed N fixated by day.

- *DS_start_fixate*: number (dimensionless)
  Optional state variable
  Development stage at which to restart fixation after a cut.

Log Variables

- *PtNCnt*: number [$\mathbf{g/m^2}$]
  Potential nitrogen content in crop.
- *CrNCnt*: number [$\mathbf{g/m^2}$]
  Critical nitrogen content in crop.
- *NfNCnt*: number [$\mathbf{g/m^2}$]
  Non-functional nitrogen content in crop.
- *nitrogen_stress*: number (dimensionless)
  Nitrogen stress factor.
- *Fixated*: number [$\mathbf{g\ N/m^2/h}$]
  N fixation from air.
- *AccFixated*: number [$\mathbf{g\ N/m^2}$]
  Accumuated N fixation from air.

85.20 DOM

A single Dissolved Organic Matter pool.

```
< DOM   (N N)                                          ; Has partial value.
        (C C)                                          ; Has partial value.
        (diffusion_coefficient diffusion_coefficient)
        (heat_factor heat_factor)
        (water_factor water_factor)
        (turnover_rate turnover_rate)
        (turnover_halftime turnover_halftime)
        (efficiency efficiency ...)
        (fractions fractions ...) >
```

- *N*: **DOM-Element** fixed component (see section 85.25)
  Submodel (has partially specified default value)
  Nitrogen content of DOM pool.
- *C*: **DOM-Element** fixed component (see section 85.25)
  Submodel (has partially specified default value)
  Carbon content of DOM pool.
- *diffusion_coefficient*: number [$\mathbf{cm^2/s}$]
  Parameter
  Diffusion coefficient.
- *heat_factor*: plf [$\mathbf{dg\ C \rightarrow <none>}$]
  Optional parameter
  Heat factor. If empty, use default from 'OrganicMatter'.
- *water_factor*: plf [$\mathbf{cm \rightarrow <none>}$]
  Optional parameter
  Water potential factor. If empty, use default from 'OrganicMatter'.
- *turnover_rate*: number [$\mathbf{h^{-1}}$]
  Optional parameter
  Fraction converted to other pools each hour. You must specify either this or 'turnover_halftime'.

- *turnover_halftime*: number [**h**]
  Optional parameter
  Time until half had been converted to other pools. You must specify either this or 'turnover_rate'.

- *efficiency*: number [<**fraction**>] sequence
  Parameter
  the efficiency this pool can be digested by each of the SMB pools.

- *fractions*: number [<**fraction**>] sequence
  Parameter
  Fraction of this pool that ends up in each SMB pools

85.21 Time

Year, month, day and hour, minute, second and microsecond.

Used by Weatherdata @ Begin (see 85.2, page 407) , Harvest @ time (see 85.4, page 412) , and Harvesting @ last_cut (see 85.16, page 431) .

```
< Time  year month mday hour
        (minute 0)
        (second 0)
        (microsecond 0) >
```

- *year*: integer
  State variable
  Current year.

- *month*: integer
  State variable
  Current month.

- *mday*: integer
  State variable
  Current day in the month.

- *hour*: integer
  State variable (default 0)
  Current hour.

- *minute*: integer
  State variable (default 0)
  Current minute.

- *second*: integer
  State variable (default 0)
  Current second.

- *microsecond*: integer
  State variable (default 0)
  Current microsecond.

Log Variables

- *yday*: integer
  Current Julian day.

- *week*: integer
  Current week.

- *wday*: string (see section 1.4.5)
  Current weekday. Monday is 1, Sunday is 7.

85.22 HorHeat

Heat capacity and conductivity per horizon.
Used by horizon component HorHeat (see 35, page 183) .

```
< HorHeat   (quarts_form_factor 2 [])
            (mineral_form_factor 4 [])
            (intervals 100)
            (C_soil C_soil)
            (K_water K_water ...)
            (K_ice K_ice ...) >
```

- *quarts_form_factor*: number (dimensionless)
  Parameter (default 2)
  Gemetry factor used for conductivity calculation.

- *mineral_form_factor*: number (dimensionless)
  Parameter (default 4)
  Gemetry factor used for conductivity calculation.

- *intervals*: integer
  Parameter (default 100)
  Number of numeric intervals to use in the heat coductivity table.

- *C_soil*: number [**erg/cm^3/dg C**]
  Optional parameter
  The soils heat capacity. By default, this is calculated from the soil constituents.

- *K_water*: number [**erg/s/cm/dg C**] soil cells
  Optional parameter
  Heat conductivity table for water in soil. By default, this is calculated from the soil constituents.

- *K_ice*: number [**erg/s/cm/dg C**] soil cells
  Optional parameter
  Heat conductivity table for solid frozen soil. By default, this is calculated from the soil constituents.

85.23 Fetch

A summary file line.

```
< Fetch   tag   >
```

- *tag*: string (see section 1.4.5)
  Parameter
  The tag of a column in the log file to summarize in this line.

85.24 Geometry1D

A one dimensional discretization of the soil.
< `Geometry1D (zplus` *zplus* `...)` >

- *zplus*: number [**cm**] soil cells
  Optional parameter
  Depth of each numeric layer (a negative number). The end points are listed descending from the surface to the bottom.

85.25 DOM-Element

A single element in a Dissolved Organic Matter pool.
Used by DOM @ N (see 85.20, page 441) .
< `"DOM-Element" (M` *M* `...)` >

- *M*: number [$\mathbf{g/cm^3}$] soil cells
  State variable
  Mass in water and soil.

Log Variables

- *S*: number [$\mathbf{g/cm^3/h}$] soil cells
  Combined source term.

- *C*: number [$\mathbf{g/cm^3}$] soil cells
  Concentration in water.

- *S_drain*: number [$\mathbf{g/cm^3/h}$] soil cells
  Source term (soil drainage only).

- *J_matrix*: number [$\mathbf{g/cm^2/h}$] soil edges
  Transportation in matrix (positive up).

- *J_tertiary*: number [$\mathbf{g/cm^2/h}$] soil edges
  Transportation outside matrix (positive up).

- *S_p*: number [$\mathbf{g/cm^3/h}$] soil cells
  Source term (macropore transport only).

85.26 GeometryRect

A rectangular discretization of the soil.
< `GeometryRect (zplus` *zplus* `...)`
`(xplus` *xplus* `...)` >

- *zplus*: number [**cm**] sequence
  Parameter
  Depth of each numeric layer (a negative number). The end points are listed descending from the surface to the bottom.

- *xplus*: number [**cm**] sequence
  Parameter
  Horizontal end of each numeric layer (a positive number). The end points are listed ascending from left (0.0) to right.

85.27 Toplevel

The top level syntax for a Daisy setup file.

The main parameter is 'run', which determines the program to run. It is also the only real parameter, the rest are more appropriately called 'commands', as the have an immediate effect. These are mostly concerned with reading additionally files ('input'), and where to find those files ('directory' and 'path').

Also found at top level, but not listed here, are the named parameterizations. The format of those are:

(defCOMP NEW OLD "DOC" PARS...)

Here COMP is the the component for which you want to define a new parameterization, NEW is the name of the new paramterization, OLD the name of an already defined parameterization or model you want to base the new parameterization on, and PARS... is a list of parameters and values you want to set for this paramaterization. "DOC" is an optional decsription of the paramterization.

You can declare new parameters both at the top level, and inside the PARS... list in the paramterization definition described above. Once you have declared a new parameter, you can set it just like the build-in parameters. The syntax for declaring a new parameter is

(declare PAR [SIZE] TYPE "DOC")

Here PAR is the name of the new parameter we want to declare, [SIZE] is either missing, in which case the new parameter is a singleton, [], in which case the new parameter is a sequence with an arbitrary length, of SIZE is an integer, in which case the sequence must hold exactly that number of values. TYPE is either a name of a component, in which case the parameter must hold a paramterization of a model within that component, or String, Integer, or Number, in which case the value must be a primitive of that type. For Number, you can also specify a dimension in square brackets afterwards. DOC is a non-optional description of the new parameter.

```
< Toplevel  (ui ui)
            (input input)
            (path path ...)                    ; Has default value.
            (directory directory)
            (run run)
            (allow_old_units true)
            (install_directory "C:/daisy") >
```

- *ui*: **ui** component (see chapter 74)
  Optional component
  Top level user interface.

- *input*: **parser** component (see chapter 47)
  Optional component
  Command to add more information about the simulation.

- *path*: string (see section 1.4.5) sequence
  Parameter (has default value with length 3)

  ```
  (path "." "../lib" "../sample")
  ```

  Parameter description:
  List of directories to search for input files in. The special value "." means the current directory.

  By default, this variable will be initialised from the DAISYPATH environment variable if it exists. The value of the variable should be a list of directories to search for input files in, seperated by semicolon on MS Windows, or colon on

other systems. If the DAISYPATH environment variable is not set, the path will be initialized to the working directory followed by the standard parameter libraries.

- *directory*: string (see section 1.4.5)
  Optional parameter
  Run program in this directory. This can affect both where input files are found and where log files are generated.

- *run*: **program** component (see chapter 52)
  Optional component
  Program to run.

  If this option is specified, all the 'Daisy' specific top-level attributes will be ignored. If unspecified, run 'Daisy' on the current top-level attributes.

- *allow_old_units*: boolean (see section 1.4.2)
  Parameter (default true)
  OBSOLETE: Set this to true to enable the old system of build-in unit conversation.

- *install_directory*: string (see section 1.4.5)
  Parameter (default 'C:/daisy')
  Directory where Daisy has been installed.

  This is used for looking up files that came with the installation, in particular the parameter library. By default, the value of the DAISYHOME environment variable is used. If DAISYHOME is not set, and the program is running under MS Windows, the value of the "Install Directory" registry key is used. If that is not set either (or we are not running MS Windows), a hardcoded value is used. This is "C:/daisy" under MS Windows, or "/usr/local/daisy" on other systems.

  The value found in the manual corresponds to the system where the manual was generated.

85.28 FetchPretty

A summary file line.

```
< FetchPretty   tag
                (name name)
                (factor 1 []) >
```

- *tag*: string (see section 1.4.5)
  Parameter
  The tag of a column in the log file to summarize in this line.

- *name*: string (see section 1.4.5)
  Optional parameter
  Name to use for this line. By default use the tag.

- *factor*: number (dimensionless)
  Parameter (default 1)
  Factor to multiply with to get the sum. Typically 1.0 to add this line, or -1.0 to subtract it.

85.29 Irrigation

Keep track of active irrigation events. Usually not set explicitly, but may be found in a checkpint.

< `Irrigation (event)` >

- *event*: submodel (see section 1.4.7) sequence
  Submodel (default: an empty sequence)
  Currently active irrigation events.

  < `(solute` *solute* `...)`
  `(volume` *volume*`)`
  `(flux` *flux*`)`
  `(temperature` *temperature*`)`
  `(time_left` *time_left*`)`
  `(target` *target*`)`
  `(silence` *silence*`)` >

  – *solute*: submodel (see section 1.4.7) sequence
    Solutes in irrigation water.

    < *name value* >

    * *name*: string (see section 1.4.5)
      State variable
      Name of chemical.
    * *value*: number [**g/cm^2/mm**]
      State variable
      Value for chemical.

  – *volume*: **volume** component (see chapter 81)
    Optional component
    Soil volume to apply for subsoil irrigation. Ignored for overhead and surface irrigation.

  – *flux*: number [**mm/h**]
    State variable
    Water applied.

  – *temperature*: number [**dg C**]
    Optional state variable
    Irrigation temperature. By default, use daily air temperature. Ignored for subsoil irrigation.

  – *time_left*: number [**h**]
    State variable
    Time left of this irrigation event.

  – *target*: string (see section 1.4.5)
    State variable
    Where to apply the irrigation.
    overhead: Above crop canopy. surface: On soil surface, below crop canopy. subsoil: In the soil. The 'volume' parameter will specify where.

  – *silence*: boolean (see section 1.4.2)
    State variable
    True if event should not declare when it is over.

85.30 StringerCondClause

If condition is true, return value.

< `StringerCondClause` *`condition value`* >

- *condition*: **boolean** component (see chapter 15)
  Condition to test for.
- *value*: string (see section 1.4.5)
  Parameter
  Value to return.

85.31 IntegerCondClause

If condition is true, return value.

< `IntegerCondClause` *`condition value`* >

- *condition*: **boolean** component (see chapter 15)
  Condition to test for.
- *value*: integer
  Parameter
  Value to return.

Version

Daisy version 4.93. LaTeX manual generated: Tue Feb 15 14:05:19 2011

Appendix A

Daisy Weather File Format

The Daisy Weather File format (*dwf* for short) is a flexible format for specifying weather data. The easiest way to create a dwf file is to edit an existing file. Here, we will describe the format for reference purposes.

First some general syntax. Empty lines and lines beginning with '`#`' are ignored, and can occur anywhere except at the very first line. Line starting with '`#`' are called *comment lines*. All words are case-sensitive, and must be written exactly as specified.

The first line *must* begin with the string '`dwf-0.0`', followed by whitespace. The rest of the line is ignored.

After the first line, the keyword section follows. Each line (except blank lines and comment lines) in the keyword section have the general format:

keyword: *value*

or

keyword: *value dimension*

All keywords, with the exception of '`Note`' and '`Timestep`', must occur exactly once. The sequence doesn't matter.

Here is a list of the recognized keywords, and the dimensions allowed for them:

Station The rest of the line (after the colon) contains the name of the weather station.

Elevation Height above sea level of station, given in `[m]`.

Longitude Longitude of station, given in either `[dgEast]` or `[dgWest]`.

Latitude Latitude of station, given in either `[dgNorth]` or `[dgSouth]`.

TimeZone Time zone used for the data, given in either `[dgEast]` or `[dgWest]`.

Surface Measurement conditions at surface, the value should either be '`reference`' (short grass) or '`field`' if the data have been measured at the simulated field.

ScreenHeight Measurement height above surface, given in `[m]`.

PrecipCorrect Factors to multiply to the precipitation for each month. Should have the form of twelve numbers, seperated by spaces, the first number representing the factor for Janurary, and the last for December. So if the first number is 1.20 and the weather data for a given day in Janurary specifies 10 mm precipitation, Daisy will calculate with 12 mm precipitation for that day.

Begin First data point, given in the format `yyyy-mm-dd` or `yyyy-mm-dd:hh`.

End Last data point, given in the format `yyyy-mm-dd` or `yyyy-mm-dd:hh`. Note that the date specified by `End` should be a few days before the actual end of the weather data, to prevent Daisy from reading past the end of the file. Also, make sure the simulation ends before the weather data, or the results may be lost.

Timestep Time between data points, given in `[hours]`. You can leave out this keyword, in that case the time for each data point must be specified. Leaving it out allows for varying timesteps.

Note You can have any number of these, but they must come in sequence. That is, no other keywords between two notes. You can have any text after the keyword.

TAverage Average temperature for the location, given in `[dgC]`.

TAmplitude Amplitude of yearly temperature variation, given in `[dgC]`.

MaxTDay The Julian day with the highest temperature (on average), given in `[yday]`.

There are two wayes of specifying deposition. Directly

NH4WetDep NH_4 deposition in precipitation, given in `[ppm]`.

NH4DryDep NH_4 deposition from air, given in `[kgN/ha/year]`.

NO3WetDep NO_3 deposition in precipitation, given in `[ppm]`.

NO3DryDep NO_3 deposition from air, given in `[kgN/ha/year]`.

Or alternatively, indirectly

Deposition Mean yearly deposition, given in `[kgN/ha/year]`.

PAverage Total yearly precipitation, given in `[mm]`.

DepDry Dry fraction of total deposition (from air). By default 40%. The remainder will fall with the precipitation.

DepDryNH4 Fraction of NH_4 in deposition from air. By default 60%. The remainder will fall in the form of NO_3.

DepWetNH4 Fraction of NH_4 in deposition from rain and snow. By default 50%. The remainder will fall in the form of NO_3.

You cannot specify both. If you use the indirect specification, these numbers will be used for calculating the equivalents of the direct specification of deposition.

After the keywords there should be a line consisting solely of hyphens. It marks the beginning of the data section. The lines in the data section are divided into columns by whitespace. All lines must have the same number of columns. The first line contains the name of the data type specified in each column, the second line the dimension of that data, and the following lines the actual measurement data in chronological order. The following is a list of the possible data type names, as well as the recognized dimensions.

Year Given in `[year]`.

Month Given in `[month]`.

Day Given in `[mday]`.

Hour Given in `[hour]`.

GlobRad Global radiation, given in `[W/m^2]`.

AirTemp Air temperature, given in `[dgC]`.

T_min Daily minimum air temperature, given in `[dgC]`.

T_max Daily maximum air temperature, given in `[dgC]`.

Precip Precipitation, given in `[mm/h]` or `[mm/d]`.

RefEvap Reference evapotranspiration, given in `[mm/h]` or `[mm/d]`.

VapPres Vapor pressure, given in `[Pa]`.

RelHum Relative humidity, given in `[fraction]` or `[%]`.

Wind Wind speed, given in `[m/s]`

The columns can be arranged in any sequence. GlobRad is mandatory, Year, Month, and Day are mandatory if no timestep have been specified. You cannot specify both VapPres and relHum. All other columns are optional.

Bibliography

[Abrahamsen, 1997] Abrahamsen, P. (1997). Daisy programmers guide. `http://www.dina.kvl.dk/~daisy/guide/guide.html`.

[Abrahamsen, 1998] Abrahamsen, P. (1998). The daisy c API. `http://www.dina.kvl.dk/~daisy/c-api/c-api.html`.

[Abrahamsen, 2003] Abrahamsen, P. (2003). *Daisy Tutorial*. Distributed with Daisy.

[Abrahamsen and Hansen, 2000] Abrahamsen, P. and Hansen, S. (2000). Daisy: An open soil-crop-atmosphere model. *Environmental Modelling and Software*, 15(3):313–330.

[Ahmadi et al., 2009] Ahmadi, S. H., Andersen, M. N., Poulsen, R. T., Plauborg, F., and Hansen, S. (2009). A quantitative approach to developing more mechanistic gas exchange models for field grown potato: A new insight into chemical and hydraulic signalling. *Agricultural and Forest Meteorology*, 149(9):1541 – 1551.

[Ball et al., 1987] Ball, J., Woodrow, I., and Berry, J. (1987). A model predicting stomatal conductance and its contribution to the control of photosynthesis under different environmental conditions. *Progress in Photosynthesis Research*, 4:221–224.

[Børgesen et al., 2006] Børgesen, C., Jacobsen, O., Hansen, S., and Schaap, M. (2006). Soil hydraulic properties near saturation, an improved conductivity model. *Journal of Hydrology*, 324(1-4):40–50.

[Brown and Foster, 1987] Brown, L. and Foster, G. (1987). Storm erosivity using idealized intensity distributions. *Transactions of the ASAE - American Society of Agricultural Engineers*, 30(2):379–386.

[Brubaker et al., 1992] Brubaker, S. C., Holzhey, C. S., and Brasher, B. R. (1992). Estimating the water-dispersible clay content of soils. *Soil Science Society of America Journal*, 56(4):1226–1232.

[Bruun et al., 2003] Bruun, S., Christensen, B. T., Hansen, E. M., Magid, J., and Jensen, L. S. (2003). Calibration and validation of the soil organic matter dynamics of the daisy model with data from the Askov long-term experiments. *Soil Biology & Biochemistry*, 35(1):67–76.

[Burdine, 1953] Burdine, N. (1953). Relative permeability calculations from pore-size distribution data. *Trans. AIME*, 198:71–77.

[CAMPBELL, 1974] CAMPBELL, G. (1974). A simple method for determining unsaturated conductivity from moisture retention data. *Soil Science*, 117(6):311.

[de Pury and Farquhar, 1997] de Pury, D. and Farquhar, G. (1997). Simple scaling of photosynthesis from leaves to canopies without the errors of big-leaf models. *Plant Cell and Environment*, 20(5):537–557.

[der Molen and Wesseling, 1991] der Molen, W. V. and Wesseling, J. (1991). A solution in closed form and a series solution to replace the tables for the thickness of the equivalent layer in Hooghoudt's drain spacing formula. *Agricultural Water Management*, 19(1):1–16.

[Gerwitz and Page, 1974] Gerwitz, A. and Page, E. R. (1974). An empirical mathematical model to describe plant root systems. *The Journal of Applied Ecology*, 11(2):773–381.

[Hairsine and Rose, 1991] Hairsine, P. and Rose, C. (1991). Rainfall detachment and deposition: Sediment transport in the absence of flow-driven processes. *Soil Sci. Soc.Am.J.*, 55:320–324.

[Hansen, 2002] Hansen, S. (2002). Daisy, a flexible soil-plant-atmosphere system model. `http://code.google.com/p/daisy-model/downloads/detail?name=DaisyDescription.pdf`.

[Hansen et al., 1990] Hansen, S., Jensen, H. E., Nielsen, N. E., and Svendsen, H. (1990). Daisy — soil plant atmostphere system model. Technical Report A10, Miljøstyrelsen.

[Hansen et al., 1991] Hansen, S., Jensen, H. E., Nielsen, N. E., and Svendsen, H. (1991). Simulation of nitrogen dynamics and biomass production in winter wheat using the Danish simulation model daisy. *Fertilizer Research*, 27:245–259.

[Hooghoudt, 1940] Hooghoudt, S. B. (1940). Algemene beschouwing van het probleem van de detailontwatering en de infiltratie door middel van parallel loopende drains, greppels, slooten en kanalen. Bijdragen tot de kennis van eenige natuurkundige grootheden van den grond 7, Rijksuitgeverij Dienst van de Nederlandse Staatscourant. 's-Gravenhage, Algemeene Landsdrukkerij.

[Jarvis et al., 1999] Jarvis, N. J., Villholth, K. G., and Ulén, B. (1999). Modelling particle mobilization and leaching in macroporous soil. *European Journal of Soil Science*, (50):621–632.

[Kjaersgaard et al., 2008] Kjaersgaard, J., Plauborg, F., Mollerup, M., Petersen, C., and Hansen, S. (2008). Crop coefficients for winter wheat in a sub-humid climate regime. *Agricultural Water Management*, 95(8):918–924.

[Larsbo and Jarvis, 2003] Larsbo, M. and Jarvis, N. (2003). *MACRO 5.0: a model of water flow and solute transport in macroporous soil: technical description.* Department of Soil Sciences, Swedish Univ. of Agricultural Sciences.

[Leuning, 1995] Leuning, R. (1995). critical appraisal of a combined stomatal-photosynthesis model for c3 plants. *Plant, Cell and Environment*, 18(4):339–355.

[Macena et al., 2003] Macena, F., Corbeels, M., Scopel, E., Affholder, F., and Pinto, H. (2003). Characterising effects of surface residues on evaporation for a simple water balance model. In *Proceedings 2nd World Congress on Conservation Agriculture, Foz do Iguaçu, Brazil*, volume 2, pages 522–524.

[Madsen and Platou, 1983] Madsen, H. and Platou, S. (1983). Land use planning in Denmark: The use of soil physical data in irrigation planning. *Nordic Hydrology*, 14(5).

[Moody, 1966] Moody, W. T. (1966). Nonlinear differential equation of drain spacing. *Journal of the Irrigation and Drainage Division, ASCE*, 92(1R2):1–9.

[Morgan et al., 1998] Morgan, R., Quinton, J., Smith, R., Govers, G., Poesen, J., Auerswald, K., Chisci, G., Torri, D., and Styczen, M. (1998). *EUROSEM: Documentation and user guide*. Cranfield University, 3.6 edition. The European Soil Erosion Model.

[Mualem, 1976] Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3).

[Müller et al., 1997] Müller, T., Jensen, L. S., Magid, J., and Nielsen, N. E. (1997). Temporal variation of C and N turnover in soil after oilseed rape incorporation in the field: Simulation with the soil-plant-atmosphere model daisy. *Ecological Modelling*, (99):247–262.

[Park et al., 1982] Park, S., Mitchell, J., and Scarborough, J. (1982). Soil erosion simulation on small watersheds: A modified ANSWERS model. *Transactions of the ASAE*, 25(6):1581–1588.

[Scopel et al., 1998] Scopel, E., Muller, B., Arreola Tostado, J., Chavez Guerra, E., and Maraux, F. (1998). Quantifying and modelling the effects of a light crop residue on the water balance: an application to rainfed maize in Western Mexico. In *XVI World Congress of Soil Science*.

[Scopel et al., 2004] Scopel, E., Silva, F. D., Corbeels, M., Affholder, F., and Maraux, F. (2004). Modelling crop residue mulching effects on water use and production of maize under semi-arid and humid tropical conditions. *Agronomie*, 24:383–395.

[Styczen and Høeg-Schmidt, 1988] Styczen, M. and Høeg-Schmidt, K. (1988). A new description of splash erosion in relation to raindrop sizes and vegetation. In Morgan, R. and Rickson, J., editors, *Erosion assessment and modelling*. Commission of the European Communities, EUR 10860.

[FOCUS, 2002] FOCUS (2002). Generic guidance for focus groundwater scenarios. Version 1.1.

[Wesseling, 1973] Wesseling, J. (1973). Subsurface flow into drains. in: Drainage principles and applications, vol. ii: Theories of field drainage and watershed runoff. Technical report, ILRI, Wageningen, The Netherlands.

[Wösten et al., 1999] Wösten, J., Lilly, A., Nemes, A., and Bas, C. L. (1999). Development and use of a database of hydraulic properties of European soils. *Geoderma*, 90:169–185.

Index