

Daisy Program Reference Manual

Per Abrahamsen <daisy@daisy-model.org>

December 11, 2012

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

I	The Framework	23
1	Introduction	25
2	Concepts	27
2.1	Components, Models, and Parameterizations	27
2.1.1	Parameterizations	28
2.2	Model Attributes	28
2.2.1	Optional and Default Parameters	28
2.2.2	Partial Parameterizations	28
2.2.3	Log Variables	29
2.3	Attribute Types	29
2.3.1	Ordered Attributes	29
3	Nomenclature	31
3.1	Name and Description	31
3.2	Sample	31
3.3	The Attribute List	32
3.3.1	The Type Line	32
3.3.2	The Category Line	32
3.3.3	The Textual Description	33
3.4	Submodels	33
4	The file parser model	35
4.1	Types and Formatting	35
4.1.1	Order and Size	35
4.1.2	Booleans	36
4.1.3	Integers	36
4.1.4	Numbers	36
4.1.5	Strings	37
4.1.6	PLF	37
4.1.7	Submodels and Fixed Components	38
4.1.8	Library Components	38
4.2	Declaring new parameters	39
4.3	Referring to parameters	39
4.4	Toplevel	39
5	Log File Generation	41
6	Error Messages	43
6.1	Parser errors	43
6.2	Completeness checks	43
6.3	Consistency checks	43
6.4	Runtime exceptions	44

6.5	Assertion failures	44
6.6	Program errors	44
II	The Components	45
7	ABApduction	47
7.1	none	47
7.2	root	47
7.3	soil	47
7.4	uptake	48
8	AOM	49
8.1	AOM-DIRECT	50
8.2	AOM-FAST	50
8.3	CROP-FAST	51
8.4	AOM-SLOW	51
8.5	AOM-SLOW-BIOINCORPORATION	51
8.6	AOM-SLOW-OLD	51
8.7	CROP-SLOW	51
8.8	Ryegrass-SLOW	51
8.9	Wclover-SLOW	51
9	ClayOM	53
9.1	biomod	53
9.2	old	54
10	MV_Crop	55
10.1	default	55
11	MV_Soil	57
11.1	default	57
12	SMB	59
12.1	SMB-FAST	60
12.2	SMB-SLOW	60
13	SOM	61
13.1	SOM-FAST	62
13.2	SOM-FAST-OLD	62
13.3	SOM-INERT	62
13.4	SOM-SLOW	62
13.5	SOM-SLOW-OLD	62
14	action	63
14.1	assert	63
14.2	cond	63
14.3	crop	64
14.4	emerge	70
14.5	error	70
14.6	extern	71
14.7	extern_fertigation	71
14.8	extern_subsoil	72
14.9	fertilize_base	73
14.10	fertilize	74

14.11	incorporate_fertilizer	74
14.12	harvest_base	74
14.13	cut	75
14.14	harvest	75
14.15	if	75
14.16	irrigate_base	76
14.17	irrigate_overhead	76
14.18	irrigate_subsoil	76
14.19	irrigate_surface	77
14.20	markvand	77
14.21	message	78
14.22	mix	79
14.23	disk_harrowing	79
14.24	rotavation	79
14.25	seed_bed_preparation	79
14.26	stubble_cultivation	79
14.27	nil	79
14.28	panic	79
14.29	pluck	80
14.30	progn	80
14.31	repeat	80
14.32	ridge	80
14.33	sequence	81
14.34	activity	81
14.35	set_heat_source	81
14.36	set_porosity	81
14.37	set_surface_detention_capacity	81
14.38	sow	82
14.39	plant	82
14.40	spray	82
14.41	stop	83
14.42	swap	83
14.43	plowing	83
14.44	t	83
14.45	table	83
14.46	wait	85
14.47	wait_mm_dd	86
14.48	wait_period	86
14.49	wait_days	86
14.50	wait_hours	86
14.51	warning	86
14.52	while	87
14.53	with-column	87
15	adsorption	89
15.1	Freundlich	89
15.2	Langmuir	89
15.3	full	90
15.4	linear	90
15.5	NH4	90
15.6	none	90
15.7	vS_S	91
16	am	93

16.1	base	93
16.2	initial	94
16.3	organic	94
16.4	slurry	95
16.5	Foulum_slurry	95
16.6	aender_gaes_dybstroelse	95
16.7	cattle_slurry	96
16.8	faar_geder_dybstroelse	96
16.9	hest_dybstroelse	96
16.10	hoens_dybstroelse	96
16.11	hoens_gylle	96
16.12	hoens_staldgoedning	96
16.13	kalkuner_dybstroelse	96
16.14	kartoffelfrugtsaft	96
16.15	komposteret_husholdningsaffald	97
16.16	kvaeg_ajle	97
16.17	kvaeg_dybstroelse	97
16.18	kvaeg_gylle	97
16.19	kvaeg_staldgoedning	97
16.20	kvaeggylle_separeret_tyk	97
16.21	kvaeggylle_separeret_tynd	97
16.22	kvaeggylle_usepareret	98
16.23	manure	98
16.24	cattle_manure	98
16.25	horse_manure	98
16.26	pig_manure	98
16.27	mink_raev_gylle	98
16.28	mink_raev_staldgoedning	98
16.29	novogro-30_slam	98
16.30	novogro_slam	98
16.31	pig_slurry	99
16.32	pressesaft_groentpillefabrik	99
16.33	slagtekyllinger_dybstroelse	99
16.34	slagtesvin_ajle	99
16.35	slagtesvin_dybstroelse	99
16.36	slagtesvin_gylle	99
16.37	slagtesvin_staldgoedning	99
16.38	soer_ajle	99
16.39	soer_dybstroelse	100
16.40	soer_gylle	100
16.41	soer_staldgoedning	100
16.42	spildevandsslam	100
16.43	svinegylle_separeret_tyk	100
16.44	svinegylle_separeret_tynd	100
16.45	svinegylle_usepareret	100
16.46	root	101
16.47	state	101
16.48	mineral	101
16.49	Ammonia	102
16.50	AmmoniumNitrate	102
16.51	CalciumNitrate	102
16.52	N25S	102
16.53	NP	102
16.54	NPK01	102

16.55	NPK02	103
16.56	Nitrate	103
17	average	105
17.1	arithmetic	105
17.2	geometric	105
17.3	harmonic	105
18	bioclimate	107
18.1	default	107
19	biopore	113
19.1	drain	114
19.2	matrix	114
20	boolean	117
20.1	false	117
20.2	not	117
20.3	numbers	117
20.4	<	117
20.5	<=	117
20.6	>	118
20.7	>=	118
20.8	operands	118
20.9	and	118
20.10	or	118
20.11	string-equal	118
20.12	true	118
20.13	xor	118
21	bound	119
21.1	finite	119
21.2	full	119
21.3	none	119
21.4	state	119
21.5	empty	120
22	chemical	121
22.1	default	121
22.2	common	130
22.3	pesticide	130
22.4	fungicide	130
22.5	herbicide	130
22.6	2,4-D	130
22.7	Atrazine	130
22.8	Bentazon	131
22.9	IPU	131
22.10	Ioxynil	131
22.11	MCPP	131
22.12	Pendimethalin	131
22.13	insecticide	131
22.14	DDT	131
22.15	Heptachlor	131
22.16	nutrient	131

22.17	N	131
22.18	NH ₄	132
22.19	NO ₃	132
22.20	solid	132
23	chemistry	133
23.1	default	133
23.2	N	133
23.3	pesticides	133
23.4	multi	134
23.5	none	135
23.6	nutrient	135
24	column	137
24.1	default	138
24.2	Askov	140
24.3	Foulum	140
24.4	Jyndevad	140
25	condedge	141
25.1	arithmetic	141
25.2	geometric	141
25.3	harmonic	141
25.4	pressure	141
26	condition	143
26.1	TSum_above	143
26.2	and	144
26.3	trafficable	144
26.4	check	144
26.5	crop_dm_over	144
26.6	crop_dm_sorg_over	144
26.7	crop_ds_after	145
26.8	daily	145
26.9	daily_air_temperature_above	145
26.10	daily_precipitation_above	145
26.11	every	145
26.12	extern	146
26.13	false	146
26.14	finished	146
26.15	hour	146
26.16	hourly	146
26.17	if	147
26.18	mday	147
26.19	minutely	147
26.20	mm_dd_base	147
26.21	after_mm_dd	148
26.22	before_mm_dd	148
26.23	mm_dd	148
26.24	month	148
26.25	monthly	148
26.26	not	148
26.27	or	148
26.28	periodic	149

26.29	running	149
26.30	secondly	149
26.31	soil_inorganic_N_above	149
26.32	soil_temperature_above	149
26.33	soil_water_content_above	150
26.34	soil_water_pressure_above	150
26.35	time	150
26.36	after	150
26.37	at	150
26.38	before	151
26.39	timestep	151
26.40	true	151
26.41	weekly	151
26.42	yday	151
26.43	year	151
26.44	yearly	151
27	crop	153
27.1	default	153
27.2	Aert	155
27.3	Beetroot	155
27.4	Broccoli	155
27.5	Broccoli - transplanted	155
27.6	Brussels sprouts	155
27.7	Brussels sprouts - transplanted	155
27.8	Celeriac	155
27.9	Celeriac - transplanted	156
27.10	Fodder Beet	156
27.11	Froegraes	156
27.12	Graes	156
27.13	Grass to grain	156
27.14	Grass	156
27.15	Maize	156
27.16	Ikuwala Maize	156
27.17	Silage Maize	156
27.18	Onion	156
27.19	Onion - planting of sets	157
27.20	Pea	157
27.21	Green Pea	157
27.22	Pioneer Maize	157
27.23	Potato; Koege	157
27.24	Potato; SCRI	157
27.25	Potato; FertOrgaNic	157
27.26	Potato; Agria	157
27.27	Potato; Folva	157
27.28	Potato	158
27.29	Potato; Triada	158
27.30	Potato; SCRI - AArhus	158
27.31	Early potato	158
27.32	Rug	158
27.33	Rye	158
27.34	Ryegrass	158
27.35	Silomajs	158
27.36	Spring Barley	158

27.37	Spring Barley; Foulum	159
27.38	Spring Rape	159
27.39	Spring Wheat	159
27.40	Sugar Beet	159
27.41	Vaarbyg	159
27.42	Vinterbyg	159
27.43	Vinterhvede	159
27.44	Vinterraps	159
27.45	Wclover	159
27.46	White cabbage	160
27.47	White cabbage - transplanted	160
27.48	Early white cabbage - transplanted	160
27.49	Winter Barley	160
27.50	Winter Barley; Foulum	160
27.51	Winter Barley; Koge	160
27.52	Winter Rape	160
27.53	Winter Wheat	160
27.54	Winter Wheat; Eest	160
27.55	Winter Wheat; Foulum	160
27.56	simple	161
28	depth	163
28.1	PLF	163
28.2	const	163
28.3	extern	163
28.4	file	164
29	difrad	165
29.1	DPF	165
29.2	weather	165
30	domsorp	167
30.1	default	167
31	drain	169
31.1	lateral	169
31.2	none	170
32	draineqd	171
32.1	MolenWesseling	171
32.2	Moody	171
32.3	Wesseling	171
32.4	none	171
33	element	173
33.1	atom	173
34	equilibrium	175
34.1	Langmuir	175
34.2	goal_A	175
34.3	linear	176
35	exchange	177
35.1	name	177
35.2	number	177

36 format	179
36.1 LaTeX	179
37 gnuplot	181
37.1 common	181
37.2 profile	182
37.3 soil	183
37.4 time	185
37.5 vector	185
37.6 xy	187
37.7 multi	189
38 groundwater	191
38.1 common	191
38.2 aquitard	191
38.3 deep	192
38.4 extern	192
38.5 file	192
38.6 fixed	192
38.7 flux	193
38.8 lysimeter	193
38.9 source	193
38.10 static	193
38.11 pipe	193
39 heatrect	197
39.1 Mollerup	197
39.2 linear	197
39.3 none	197
40 horizon	199
40.1 BSI3	201
40.2 DIN3	202
40.3 MIT3	202
40.4 BSI7	202
40.5 MIT7	203
40.6 DIN5	203
40.7 ISSS3	204
40.8 ISSS4	205
40.9 USDA3	206
40.10 FAO3	207
40.11 USDA7	207
40.12 FAO7	208
40.13 USPRA3	208
40.14 USPRA4	209
40.15 default	210
40.16 Ap_JB1	211
40.17 Ap_JB2	211
40.18 Ap_JB3	211
40.19 Ap_JB4	211
40.20 Ap_JB5	211
40.21 Ap_JB6	211
40.22 Ap_JB7	211
40.23 Askov Ap	211

40.24	Askov B	212
40.25	Askov C	212
40.26	B_JB1	212
40.27	B_JB2	212
40.28	B_JB3	212
40.29	B_JB4	212
40.30	B_JB5	212
40.31	B_JB6	212
40.32	B_JB7	212
40.33	C_JB1	212
40.34	C_JB2	212
40.35	C_JB3	213
40.36	C_JB4	213
40.37	C_JB5	213
40.38	C_JB6	213
40.39	C_JB7	213
40.40	Foulum AP	213
40.41	Foulum B	213
40.42	Foulum C	213
40.43	Jyndevad Ap	213
40.44	Jyndevad C	213
40.45	aquitard	214
40.46	numeric	214
41	hydraulic	215
41.1	B_BaC	215
41.2	B_BaC_Bimodal	216
41.3	B_C	217
41.4	B_C_inverse	218
41.5	B_vG	218
41.6	Cosby_et_al	219
41.7	MACRO	220
41.8	M_BaC	221
41.9	M_BaC_Bimodal	222
41.10	M_C	222
41.11	M_vG	223
41.12	M_vG_compact	224
41.13	M_vGp	225
41.14	hypres	226
41.15	mod_C	227
41.16	yolo	228
42	integer	229
42.1	*	229
42.2	+	229
42.3	-	229
42.4	cond	229
42.5	const	230
42.6	div	230
42.7	max	230
42.8	min	230
42.9	mod	230
42.10	sqr	230

43 litter	231
43.1 none	231
43.2 permanent	231
43.3 residue	232
43.4 Maize	232
43.5 Millet	232
44 log	233
44.1 checkpoint	233
44.2 harvest	233
44.3 select	234
44.4 extern	235
44.5 table	236
44.6 column	237
44.7 Field nitrogen	238
44.8 Field water	240
44.9 Soil nitrogen	241
44.10 Soil water	244
44.11 biopore	245
44.12 chemical	246
44.13 Field chemical	247
44.14 Soil chemical	249
44.15 crop	250
45 macro	253
45.1 default	253
45.2 none	254
46 mactrans	255
46.1 default	255
47 movement	257
47.1 solute	257
47.2 rectangle	257
47.3 vertical	258
48 net_radiation	261
48.1 brunt	261
48.2 brutsaert	262
48.3 idso_jackson	262
48.4 prata	262
48.5 satterlund	262
48.6 swinbank	262
49 nitrification	263
49.1 soil	263
49.2 solute	263
50 number	265
50.1 *	265
50.2 +	265
50.3 -	265
50.4 /	265
50.5 child	266
50.6 convert	266

50.7	dim	266
50.8	identity	266
50.9	const	267
50.10	initial_zero	267
50.11	zero_gradient	267
50.12	depth	267
50.13	depth_K	267
50.14	depth_Theta	268
50.15	exp	268
50.16	fetch	268
50.17	get	268
50.18	horizon	268
50.19	soil_K	269
50.20	soil_Theta	269
50.21	soil_heat_capacity	269
50.22	soil_heat_conductivity	269
50.23	if	269
50.24	initial_C	269
50.25	initial_NH4	269
50.26	initial_NO3	270
50.27	let	270
50.28	ln	270
50.29	log10	270
50.30	max	270
50.31	min	270
50.32	plf	271
50.33	pow	271
50.34	soil_h	271
50.35	source	272
50.36	source_average	272
50.37	source_increase	272
50.38	source_sum	272
50.39	source_unique	272
50.40	sqr	272
50.41	sqrt	272
50.42	x	273
51	organic	275
51.1	default	275
51.2	none	282
52	parser	283
52.1	file	283
53	pet	285
53.1	FAO_PM	285
53.2	Hargreaves	286
53.3	PM	286
53.4	makkink	286
53.5	weather	286
54	phenology	287
54.1	TSum	287
54.2	default	288

55 photosynthesis	291
55.1 Farquhar	291
55.2 FC_C3	293
55.3 FC_C4	295
55.4 GL	296
56 pond damp	299
56.1 EUROSEM	299
56.2 Hairsine91	299
56.3 Park82	299
56.4 none	299
57 program	301
57.1 AM_table	301
57.2 Daisy	301
57.3 GP2D	303
57.4 Osvaldo	304
57.5 batch	304
57.6 cd	305
57.7 cpedata	305
57.8 docmodel	307
57.9 document	307
57.10 gnuplot	307
57.11 hmovie	308
57.12 hydraulic	310
57.13 minimize	311
57.14 rootmatch	312
57.15 write	315
57.16 wse	315
58 raddist	317
58.1 default	317
58.2 sun-shade	317
59 rainergy	319
59.1 Brown87	319
59.2 EUROSEM	319
60 reaction	321
60.1 adsorption	321
60.2 bound_release	322
60.3 colgen	322
60.4 colgen_Jarvis99	323
60.5 colgen_Morgan98	324
60.6 colgen_Styczen88	325
60.7 default	325
60.8 denitrification	326
60.9 equilibrium	327
60.10 filter_velocity	328
60.11 nitrification	329
60.12 sorption	330
61 rootdens	333
61.1 Anders Pedersen	333

61.2	DS_Depth	334
61.3	DS_Rel	334
61.4	Depth_Depth	335
61.5	GP1D	335
61.6	GP2D	336
61.7	Gerwitz+Page74	337
61.8	growth	337
62	rubiscoNdist	339
62.1	exp	339
62.2	expr	339
62.3	forced	340
62.4	uniform	340
63	scope	341
63.1	exchange	341
64	scopesel	343
64.1	multi	343
64.2	name	343
64.3	null	343
65	secondary	345
65.1	alpha	345
65.2	cracks	345
65.3	pressure	346
65.4	none	346
66	seed	347
66.1	LAI	347
66.2	release	348
67	select	349
67.1	array	352
67.2	value	352
67.3	content	352
67.4	flow	353
67.5	flow_back	353
67.6	flow_bottom	353
67.7	flux_bottom	354
67.8	flow_front	354
67.9	flow_left	354
67.10	flow_right	354
67.11	flow_top	355
67.12	flux_top	355
67.13	index	355
67.14	number	356
67.15	volume_base	356
67.16	interval	358
67.17	volume	359
67.18	water	359
67.19	water_interval	359
67.20	water_volume	360
68	solute	361

69 solver	363
69.1 cxsparse	363
69.2 none	363
69.3 ublas	363
70 source	365
70.1 arithmetic	365
70.2 column	367
70.3 combine	370
70.4 merge	370
71 stomatacon	373
71.1 WSF	373
71.2 BB_base	373
71.3 BB	374
71.4 Leuning	374
71.5 MNA	374
71.6 SHA12	375
71.7 SHA14	376
72 string	377
72.1 cond	377
72.2 identity	377
72.3 number	377
72.4 dimension	377
72.5 value	378
73 summary	379
73.1 Rsqr	379
73.2 RsqrW	380
73.3 balance	380
73.4 Field crop nitrogen balance	381
73.5 Field mineral nitrogen balance	381
73.6 Field nitrogen balance	381
73.7 Field organic nitrogen balance	382
73.8 Field water balance	382
73.9 N_balance	382
73.10 Soil NH4+ balance	382
73.11 Soil NO3- balance	382
73.12 Soil mineral-N balance	382
73.13 Soil organic-N balance	382
73.14 Soil N balance	382
73.15 Soil water balance	383
73.16 simple	383
74 svat	385
74.1 PMSW	385
74.2 SSOC	391
74.3 none	393
75 tertiary	395
75.1 biopores	395
75.2 none	396
75.3 old	397

76 tortuosity	399
76.1 M_Q	399
76.2 linear	399
77 transform	401
77.1 equilibrium	401
78 transport	403
78.1 Hansen	403
78.2 Mollerup	403
78.3 convection	404
78.4 none	404
79 ui	405
79.1 none	405
79.2 progress	405
80 uifilter	407
80.1 base	407
80.2 raw	407
80.3 simple	407
81 uiitem	409
81.1 raw	409
82 unit	411
82.1 SI	411
82.2 SIfactor	412
82.3	412
82.4 %	412
82.5 <fraction>	412
82.6 <none>	413
82.7 A	413
82.8 J	413
82.9 J/cm ² /h	413
82.10 K	413
82.11 L/kg	413
82.12 MJ/d/m ²	413
82.13 MJ/h/m ²	413
82.14 MJ/m ² /d	413
82.15 MJ/m ² /h	413
82.16 MPa	414
82.17 MPa ⁻¹	414
82.18 Mg DM/ha	414
82.19 Mg DM/ha/h	414
82.20 Mg w.w./ha	414
82.21 Mg/ha	414
82.22 Pa	414
82.23 Pa ⁻¹	414
82.24 T w.w./ha	414
82.25 W	414
82.26 W m ⁻²	415
82.27 W/cm ²	415
82.28 W/m ²	415

82.29	ag/L	415
82.30	cd	415
82.31	cm	415
82.32	cm/d	415
82.33	cm/h	415
82.34	cm ⁻¹	415
82.35	cm ⁻²	415
82.36	cm ²	416
82.37	cm ³	416
82.38	cm ³ H ₂ O/cm ³	416
82.39	cm ³ /cm ³	416
82.40	cm ³ /g	416
82.41	cm ³ /ng	416
82.42	d	416
82.43	d ⁻¹	416
82.44	fg/L	416
82.45	g	416
82.46	g C/cm ²	417
82.47	g C/cm ³	417
82.48	g C/m ²	417
82.49	g DM/cm ²	417
82.50	g DM/m ²	417
82.51	g N/cm ²	417
82.52	g N/cm ² /h	417
82.53	g N/cm ³	417
82.54	g N/l	417
82.55	g N/m ²	417
82.56	g N/m ² /h	418
82.57	g w.w./m ²	418
82.58	g/cm/h	418
82.59	g/cm ²	418
82.60	g/cm ² /h	418
82.61	g/cm ² /mm	418
82.62	g/cm ³	418
82.63	g/ha	418
82.64	g/ha/d	418
82.65	g/ha/h	418
82.66	g/ha/mm	419
82.67	g/kg	419
82.68	g/m ²	419
82.69	g/m ² /h	419
82.70	g/mol	419
82.71	h	419
82.72	hPa	419
82.73	h ⁻¹	419
82.74	ha	419
82.75	hours	419
82.76	kPa	420
82.77	kg	420
82.78	kg C/ha	420
82.79	kg C/ha/d	420
82.80	kg C/ha/h	420
82.81	kg DM/ha	420
82.82	kg DM/m ²	420

82.83	kg N/ha	420
82.84	kg N/ha/d	420
82.85	kg N/ha/h	420
82.86	kg N/ha/mm	421
82.87	kg N/ha/y	421
82.88	kg m ⁻² s ⁻¹	421
82.89	kg w.w./ha	421
82.90	kg/Pa	421
82.91	kg/ha	421
82.92	kg/ha/h	421
82.93	kg/ha/mm	421
82.94	kg/ha/y	421
82.95	kg/m	421
82.96	kg/m/s	422
82.97	kg/m ²	422
82.98	kg/m ² /s	422
82.99	kg/m ³	422
82.100	kg/mol	422
82.101	kgN/ha/year	422
82.102	l/ha	422
82.103	l/ha/h	422
82.104	l/kg	422
82.105	m	422
82.106	m s ⁻¹	423
82.107	m/s	423
82.108	m ⁻¹	423
82.109	m ⁻²	423
82.110	m ⁻² kg s ⁻¹	423
82.111	m ²	423
82.112	m ³	423
82.113	m ³ /cm ³	423
82.114	m ³ /kg	423
82.115	mg	423
82.116	mg N/kg dry soil	424
82.117	mg N/l	424
82.118	mg/L	424
82.119	mg/g	424
82.120	mg/ha	424
82.121	mg/ha/d	424
82.122	mg/ha/h	424
82.123	mg/l	424
82.124	mg/m ²	424
82.125	minutes	424
82.126	ml/g	425
82.127	mm	425
82.128	mm/d	425
82.129	mm/h	425
82.130	mm/s	425
82.131	mm ⁻¹	425
82.132	mmol/kg	425
82.133	mmol/m ²	425
82.134	mmol/m ² /s	425
82.135	mol	425
82.136	mol/kg	426

82.137 mol/m ²	426
82.138 mol/m ² /s	426
82.139 ng/L	426
82.140 ng/MPa	426
82.141 ng/cm ³	426
82.142 ng/kPa	426
82.143 ng/l	426
82.144 ng/mm/h	426
82.145 none	426
82.146 pg/L	427
82.147 ppm	427
82.148 s	427
82.149 s ⁻¹	427
82.150 seconds	427
82.151 t/ha	427
82.152 ug	427
82.153 ug/L	427
82.154 ug/L/MPa	427
82.155 ug/ha	427
82.156 ug/ha/d	428
82.157 ug/ha/h	428
82.158 ug/l	428
82.159 ug/m ²	428
82.160 ug/m ² /h	428
82.161 ug/m ³	428
82.162 um	428
82.163 base	428
82.164 <error>	428
82.165 <unknown>	428
82.166 dgEast	429
82.167 dgNorth	429
82.168 dry soil fraction	429
82.169 rad	429
82.170 factor	429
82.171 dg	429
82.172 dgSouth	429
82.173 dgWest	429
82.174 new dg	430
82.175 ppm dry soil	430
82.176 offset	430
82.177 dg C	430
82.178 dg F	430
82.179 pF	430
83 uz1d	431
83.1 none	431
83.2 richards	431
84 uzmodel	433
84.1 lr	433
84.2 none	433
84.3 richards	433
85 uzrect	435

85.1	Mollerup	435
85.2	const	436
85.3	v+h	437
86	vegetation	439
86.1	crops	440
86.2	permanent	441
86.3	Bush	442
86.4	Coniferous	443
86.5	Grass	443
86.6	Hardwood	443
87	vernalization	445
87.1	default	445
87.2	none	445
88	volume	447
88.1	box	447
89	weather	449
89.1	weather	449
89.2	base	450
89.3	const	454
89.4	none	454
89.5	table	454
89.6	default	456
89.7	combine	458
89.8	indirect	459
89.9	time	460
90	wse	461
90.1	full	461
90.2	none	461
90.3	partial	461
91	xysource	463
91.1	arithmetic	463
91.2	combine	465
91.3	flux	466
91.4	inline	468
91.5	loop	469
91.6	merge	470
91.7	profile	471
91.8	xycombine	472
92	zone	475
92.1	box	475
92.2	polygon	476
93	Fixed Components	477
93.1	Timestep	477
93.2	Weatherdata	477
93.3	Bioincorporation	481
93.4	Harvest	482
93.5	Snow	484

93.6	SoilHeat	486
93.7	SoilWater	487
93.8	Surface	490
93.9	SoilLayer	491
93.10	SoilRegion	492
93.11	Soil	492
93.12	Ridge	493
93.13	RootSystem	494
93.14	CanopyStandard	496
93.15	CanopySimple	500
93.16	Harvesting	502
93.17	Production	504
93.18	Partition	509
93.19	CrpN	509
93.20	DOM	512
93.21	Time	513
93.22	HorHeat	514
93.23	Fetch	514
93.24	Geometry1D	515
93.25	DOM-Element	515
93.26	GeometryRect	515
93.27	Toplevel	516
93.28	FetchPretty	517
93.29	Irrigation	517
93.30	StringerCondClause	518
93.31	IntegerCondClause	518
	Version	519
A	Daisy Weather File Format	521
	Bibliography	524
	Index	528

Part I

The Framework

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, 2011].
- 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 519.

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.

Chapter 2

Concepts

In order to use the Daisy model, as well as this manual, you must already possess a fairly good understanding of the agronomic 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.

2.1 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 movement 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 84), while Richards Equation is a model (described in section 84.3).

Just about everything in Daisy are either components or models, not just what you naturally think of as physical processes. For example, crops (chapter 27), management (chapter 14), and even log files (chapter 44) 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 93.

2.1.1 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 (27.1) 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.

2.2 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.

2.2.1 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 means 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 22.1) 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 and the sorption characteristics, the ‘default’ chemical component is able to calculate the other assuming equilibrium.

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 86.1) 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.

2.2.2 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.

2.2.3 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 44). 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.

2.3 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 4.1.3) cannot hold a string (see section 4.1.5) 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 4.1.

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 *variable*, any number of values can appear. The attribute can also have an integer size, which means exactly that number of values should appear. There are also four additional symbolic sizes, *canopy cells*, *canopy edges*, *soil cells*, and *soil edges*, which means the variable must hold a number of values corresponding to the discretization of the canopy or soil, respectively. A *cell* is an interval, area, or volume, depending on whether we have a 1D, 2D, or 3D model, respectively. A cell sized variable usually hold the content or sometimes sink of the interval, area or volume of the cell. An *edge* is the connection between two neighbouring cells, and edge variables usually hold fluxes between cells.

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 4.1.4) will also have a unit, 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.

2.3.1 Ordered Attributes

Finally, some attributes may be *ordered* within the model. This order is used by the ‘file’ parser model (section 52.1) 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.

Chapter 3

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 93. 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 14.10 (that's on page 74). Section 14.10 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.

3.1 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 tells you that this is a specialization of 'fertilize_base', followed by a brief description of the model and what it does.

3.2 Sample

The next 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 non-slanted typewriter font 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 14.10, 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 2.3 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 white space. Section 14.2 has an example of that.

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

3.3 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 18.1) instead. It has plenty.

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

3.3.1 The Type Line

The first line (the line with the bullet) contains the name and the type (section 2.3, remember?) of the attribute. The name is written in slanted *italic*, to match the way it is written in the sample (section 3.2). Two types are treated specially: Numbers have a unit, 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 4.1 where they are explained.

3.3.2 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 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 have 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’.

3.3.3 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.

3.4 Submodels

Sometimes it is useful to combine attributes in a model into groups. One example of this is the ‘cond’ management action model (section 14.2), 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 27.1), 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 3.2. 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 3.3. You can even have submodels nested inside submodels, ad infinitum.

Chapter 4

The file parser model

This section describes Daisy setup files. A Daisy setup file is read by the ‘file’ model of the ‘parser’ component, hence the strange name.

4.1 Types and Formatting

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

1. Elements are separated by white space.
2. Grouping is done with parentheses. You can always (but need not) have white space before and after parentheses.
3. White space is line feeds, spaces, and tabs. You can have any number (greater than one) of these anywhere you can have white space.
4. Comments start with the character ‘;’ and ends at the end of the line. Comments are legal anywhere you can have white space.
5. All names (attribute names, component names, model names, and parameterization names) follow the same conventions as string values, described in section 4.1.5.

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.

4.1.1 Order and Size

Booleans, integers, numbers, and strings have a common format though. If *val* denotes the attribute value as described in sections 4.1.2, 4.1.3, 4.1.4, and 4.1.5, 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 4.1.

	singleton	sequence
ordered	<i>val</i>	<i>(val ...)</i>
unordered	<i>(name val)</i>	<i>(name val ...)</i>

Table 4.1: 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 4.1.6 and 4.1.7.

For sequences, you can use the special keyword `&old` anywhere in the specified value, and it will be replaced with the original value for the attribute. A typical example of its use is with the 'path' top level parameter, where you may add your own library files to the ones bundled with Daisy.

```
(path "C:/MyDaisyLib" &old)
```

	singleton	sequence
ordered	<code>(<i>val</i>)</code>	<code>((<i>val</i>)...)</code>
unordered	<code>(<i>name val</i>)</code>	<code>(<i>name (val)</i>...)</code>

Table 4.2: Format for time values, PLF's, submodels, and fixed components.

In general, library component values are written using the formatting conventions in table 4.2. However, there are two exceptions.

1. If you just specify the name of a complete parameterization, you do not need the extra parentheses, and can use the format conventions from table 4.1.
2. If there are no unordered attributes, then you do not need to put parentheses around the last attribute in the order. The action sequence is the last attribute in the 'clauses' submodel in the 'cond' action model.

4.1.2 Booleans

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

4.1.3 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 white space 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.

4.1.4 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 white space between any of the parts.

After the number, you may optionally specify the unit inside square brackets, like this [cm]. Fractions and other unitless numbers are specified as []. Daisy will check that the specified unit is also the unit it expects, or a simple conversion thereof (like [m] instead of [cm]). If the unit 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]. The

known units are listed in chapter 82, more units can be defined with the ‘defunit’ command.

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

In some contexts, numbers may be replaced by simple arithmetic expressions, using the operations defined in chapter 50.

4.1.5 Strings

Daisy accepts three different string formats. The first format is called ‘identifier’. Identifiers must start with a letter or a special characters (one of = < > _ + - * /) and can be followed by letters, digits or special characters only. They are conventionally used for attribute names.

The second 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.

The third format is intended for units. It begins with [and ends with]. The string consists of the text between the two delimiters, and can contain any character but].

For some string values, a sequence of the form ‘\${var}’ will be replaced with the value of parameter *var*, which must be a string, integer, or number. To include a ‘\$’ in such a string value, you must write ‘\$\$’. See section 4.3 for more information.

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")
```

You can instead use ordinary slashes in file names:

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

4.1.6 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 4.3.

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 4.3: Interpolated values for f

4.1.7 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 white space, 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 4.1, in particular 4.1.1.

The fixed components are found in chapter 93, 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 3.2.

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.

4.1.8 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 white space, and then the unordered attributes, just like you would for a submodel or fixed component (see section 4.1.7).

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 values you want to overwrite.

4.2 Declaring new parameters

You can declare new parameters most places where you can specify parameter values. 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, or *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 parameterization 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 must also specify a unit afterwards. *doc* is a non-optional description of the new parameter.

4.3 Referring to parameters

Instead of a parameter value, you can specify the name of another parameter.

```
(par1 $par2)
```

Here we specify the value of the parameter *par1* should be set to the value of the parameter *par2*. What makes this interesting is that *par2* does not have to be declared or have a value at the point in the setup file where you write this. The value of *par2* is only used at the time where the simulation starts, and must then be defined and have a value in an outer scope. What this means is best explained with an example. If you set the value of **clay** in a horizon to **\$MyClay**, then at the time where you start the simulation, Daisy will look for a parameter named **MyClay** in the horizon itself. If none is found, it will look for the parameter in the column where the horizon is used. If not declared there either, it will look for it in the Daisy program where the column is used. If it is still not found, it look for the parameter at the top level. And if that doesn't help, Daisy will give an error message.

Note that this substitution works on basis of a complete parameter value. You cannot substitute a single number in a PLF, nor single number in a sequence. The only exception in strings, where the sequence **\${par2}** anywhere in the string is substituted with the value of *par2*.

Also note than some parameters doesn't accept this kind of substitution at all. Only way to find out is to try.

4.4 Toplevel

A Daisy setup file consist of definitions of parameterizations and a small number of commands. The complete list of commands accepted can be found in 93.27. The two most important are 'input' and 'run'. The following short setup file demonstrates all three.

```
;;; test-refman.dai -- Sample file using the Daisy libraries.

(input file "dk-soil.dai")
(input file "dk-management.dai")

(defprogram RefSim Daisy
  "Simulation for use in reference manual."
  (weather default "dk-taastrup.dwf")
  (column Askov)
```

```

(manager "SBarley w. catch crop")
(time 1986 12 1 1)
(stop 1988 4 1 1)
(output harvest))

(run RefSim)

;;; test-refman.dai ends here.

```

The format will be described in more details in section 4.1, here we will concentrate on the three commands being used. The first is ‘input’. It directs Daisy to read commands from somewhere else. It takes a ‘parser’ component (see chapter 52, page 283) argument. The ‘file’ model (see section 52.1, page 283) will read the commands from the specified name, in the same format we are currently describing. Typically, these files contains standard parameterizations.

The next command is ‘defprogram’. It should be read as *define program*, or define a new parameterization of the ‘program’ component. This is followed by the name of the new parameterization, and the name of an existing parameterization or model to use as basis for the new parameterization. That is, if some attribute is not specified for the new parameterization, assume the value from old parameterization. Here, the new parameterization is called ‘RefSim’, and the original model to base it on is called ‘Daisy’.

In general, the format for library component values described in section 4.1.8 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 (4.1.8).

The last command is called ‘run’. It simply runs the specified program, ‘RefSim’. So, what happened here? Daisy, the program you start from Unix or Windows, actually itself contains different programs. A full list can be found in chapter 57, page 301. The main program is itself called ‘Daisy’, and will run a single Daisy simulation. We just created a parameterization containing all the mandatory parameters for a Daisy simulation, and called it ‘RefSim’. For a complete list of Daisy parameters, see section 57.2, page 301.

For compatibility with older version of Daisy (the Unix or Windows program) you can also specify the Daisy (the Daisy program) attributes directly on top level. In that case, the “run” command is not necessary.

Chapter 5

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 unit 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 44.5) 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 chapter 67) 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 unit 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 unit 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 57.2), 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.

Chapter 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.

6.1 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.

6.2 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
```

6.3 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
```

6.4 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
```

6.5 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 us.

6.6 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 7

ABAprduction

The 'ABAprduction' component calculates the prod of ABA in soil.

Used by RootSystem @ ABAprd (see 93.13, page 494) .

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

7.1 none

No ABA production.

Used by RootSystem @ ABAprd (see 93.13, page 494) .

7.2 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.

```
< root (expr expr)  
      ;; Shared parameters are described in chapter 7.  
      (description description)  
      (cite) >
```

- *expr*: **number** component (see chapter 50)
Expression to evaluate to ABA production per root length [g/cm/h]. The symbol 'h' will be bound to the water pressure [cm].

7.3 soil

ABA production based on soil location.

```
< soil (expr expr)
      ;; Shared parameters are described in chapter 7.
      (description description)
      (cite) >
```

- *expr*: **number** component (see chapter 50)
Expression to evaluate to ABA uptake [g/cm³/h]. The symbol 'h' will be bound to the water pressure [cm]. The symbol 'L' will be bound to the root density [cm/cm³]. The symbol 'S' will be bound to the water uptake [cm³/cm³/h].

7.4 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.

```
< uptake (expr expr)
        ;; Shared parameters are described in chapter 7.
        (description description)
        (cite) >
```

- *expr*: **number** component (see chapter 50)
Expression to evaluate to ABA concentration in water uptake [g/cm³]. The symbol 'h' will be bound to the water pressure [cm].

Chapter 8

AOM

A single Added Organic Matter pool.

Used by Bioincorporation @ AOM (see 93.3, page 481) , and Harvesting @ Root (see 93.16, page 502) .

```
< 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/cm2])
  (top_N 0 [g N/cm2]) >
```

- *N*: number [g N/cm³] soil cells
Optional state variable
Nitrogen in each soil interval.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *C*: number [g C/cm³] 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** \rightarrow **<none>**]
Optional parameter
Water potential factor. If empty, use default from 'OrganicMatter'.
- *C_per_N*: number [(**g C/cm³**)/(**g N/cm³**)] 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/cm²**]
State variable (default 0)
Carbon on top of soil.
- *top_N*: number [**g N/cm²**]
State variable (default 0)
Nitrogen on top of soil.

8.1 AOM-DIRECT

Third AOM pool of already decomposed material.

Used by am slurry om (see 16.4, page 95) .

8.2 AOM-FAST

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

Used by crop simple root.am (see 27.56, page 161) , vegetation permanent litter.am (see 86.2, page 441) , am root om (see 16.46, page 101) , am slurry om (see 16.4, page 95) , Bioincorporation @ AOM (see 93.3, page 481) , and Harvesting @ Root (see 93.16, page 502) .

8.3 CROP-FAST

A ‘AOM-FAST’ model (see 8.2, page 50) build into Daisy.

Parameterization used for fast pool of some crop residuals. See also [Bruun et al., 2003]

8.4 AOM-SLOW

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

Used by crop simple root.am (see 27.56, page 161) , vegetation permanent litter.am (see 86.2, page 441) , am root om (see 16.46, page 101) , am slurry om (see 16.4, page 95) , and Harvesting @ Root (see 93.16, page 502) .

8.5 AOM-SLOW-BIOINCORPORATION

A ‘AOM-SLOW’ model (see 8.4, page 51) build into Daisy.

Lower C/N ration for bioincorporated matter. See also [Bruun et al., 2003]

Used by Bioincorporation @ AOM (see 93.3, page 481) .

8.6 AOM-SLOW-OLD

A ‘AOM-SLOW’ model (see 8.4, page 51) build into Daisy.

Original parameterization of the slow AOM pool. See also [Müller et al., 1997]

8.7 CROP-SLOW

A ‘AOM-SLOW’ model (see 8.4, page 51) build into Daisy.

Parameterization used for slow pool of some crop residuals. See also [Bruun et al., 2003]

8.8 Ryegrass-SLOW

A ‘CROP-SLOW’ model (see 8.7, page 51) defined in ‘ryegrass.dai’.

Slow AOM pool modified for ryegrass. See also [Bruun et al., 2003]

8.9 Wclover-SLOW

A ‘CROP-SLOW’ model (see 8.7, page 51) defined in ‘wclover.dai’.

Slow AOM pool modified for Wclover. See also [Bruun et al., 2003]

Chapter 9

ClayOM

Find the effect of clay on organic matter processing.

9.1 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.

```
< biomod (factor factor) ; Has default value.  
      (a a)  
      (E_SMB E_SMB)  
      (f_SMB1 f_SMB1) >
```

- *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 going 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.

9.2 old

Traditional clay influence on organic matter.

Used by organic default ClayOM (see 51.1, page 275) .

`< old (factor factor) ; Has default value. >`

- *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.

Chapter 10

MV_Crop

Description of a crop for use by the MARKVAND model.

10.1 default

Standard MARKVAND crop model.

```
< default (S_F S_F ...)
          (A_F A_F ...)
          (L_gv L_gv)
          (L_ge L_ge)
          (L_gx L_gx)
          (L_gm L_gm)
          (L_ym L_ym)
          (S_Le S_Le)
          (S_Lx S_Lx)
          (S_Lr S_Lr)
          (S_Lm S_Lm)
          (z_0 z_0)
          (z_v z_v)
          (z_xA z_xA)
          (z_m z_m)
          (c_r c_r) >
```

- *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.
- *z₀*: 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 11

MV_Soil

Description of a soil for use by the MARKVAND model.

11.1 default

Standard MARKVAND soil model.

```
< default  (z_o z_o)
            (z_xJ z_xJ)
            (Theta_fo Theta_fo)
            (Theta_wo Theta_wo)
            (Theta_fu Theta_fu)
            (Theta_wu Theta_wu)
            (C_e C_e)
            (c_e c_e)
            (c_T c_T)
            (k_qr k_qr)
            (k_qb k_qb) >
```

- *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_{gr} : number (dimensionless)
Parameter
Drainage constant root zone.
- k_{qb} : number (dimensionless)
Parameter
Drainage constant subsone.

Chapter 12

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/cm³] soil cells
Optional state variable
Nitrogen in each soil interval.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *C*: number [g C/cm³] 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/cm³)/(g N/cm³)] 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.

12.1 SMB-FAST

Fast SMB pool parameterization.

Used by organic default smb (see 51.1, page 275) .

12.2 SMB-SLOW

Slow SMB pool parameterization.

Used by organic default smb (see 51.1, page 275) .

Chapter 13

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/cm³] soil cells
Optional state variable
Nitrogen in each soil interval.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *C*: number [g C/cm³] 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/cm³)/(g N/cm³)] 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.

13.1 SOM-FAST

Fast SOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]
Used by organic default som (see 51.1, page 275) .

13.2 SOM-FAST-OLD

A 'SOM-FAST' model (see 13.1, page 62) build into Daisy.
Original parameterization of the fast SOM pool. See also [Müller et al., 1997]

13.3 SOM-INERT

Inert SOM pool parameterization.
Used by organic default som (see 51.1, page 275) .

13.4 SOM-SLOW

Slow SOM pool parameterization by Sander Bruun. See also [Bruun et al., 2003]
Used by organic default som (see 51.1, page 275) .

13.5 SOM-SLOW-OLD

A 'SOM-SLOW' model (see 13.4, page 62) build into Daisy.
Original parameterization of the slow SOM pool. See also [Müller et al., 1997]

Chapter 14

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.

14.1 assert

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

```
< assert  condition
          (message message)  ; Has default value. >
```

- *condition*: **condition** component (see chapter 26)
Condition to check.
- *message*: string (see section 4.1.5)
Parameter (has default value with length 32)

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

Parameter description:
Error message to give iff assertion fails.

14.2 cond

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

```
< cond  clauses... >
```

- *clauses*: submodel (see section 4.1.7) 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 26)
Condition for performing the actions.
- *actions*: **action** component (see chapter 14) sequence
Actions to perform when condition is meet.

14.3 crop

Manage a specific crop or multicrop.

```
< crop (spray)
      (secondary secondary)
      (primary primary) ; Has partial value.
      (harvest_annual harvest_annual)
      (harvest_perennial harvest_perennial)
      (fertilize_at)
      (fertilize_at_index 0)
      (fertilize_incorporate false)
      (tillage)
      (tillage_index 0)
      (spray_index 0)
      (irrigation irrigation)
      (irrigation_rest irrigation_rest)
      (irrigation_year irrigation_year)
      (irrigation_delay irrigation_delay) >
```

- *spray*: submodel (see section 4.1.7) 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 4.1.5)
Parameter
Name of chemical to spray.
- *amount*: number [g/ha]
Parameter
Amount of chemical to spray.

- *secondary*: submodel (see section 4.1.7)
Optional submodel
Secondary crop, if any.

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

- *crop*: **crop** component (see chapter 27)
Crop to sow.
- *date*: submodel (see section 4.1.7)
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 4.1.2)
State variable (default false)
True iff the crop has been sowed.
- *primary*: submodel (see section 4.1.7)
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 27)
Crop to sow.
- *date*: submodel (see section 4.1.7)
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 4.1.2)
State variable (default false)
True iff the crop has been sowed.
- *harvest_annual*: submodel (see section 4.1.7)
Optional submodel
Harvest parameters for annual crops.

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

- *done*: boolean (see section 4.1.2)
State variable (default false)
True iff the crop has been sowed.
- *loss*: number [**<fraction>**]
Parameter
Fraction lost during harvest.
- *remove_residuals*: boolean (see section 4.1.2)
Parameter
Remove residuals at harvest.
- *latest*: submodel (see section 4.1.7)
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 4.1.7)
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 16) 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 4.1.7)
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 16) 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 4.1.7) 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 16)
Fertilizer to apply.

- *fertilize_at_index*: integer
State variable (default 0)
Next entry in 'fertilize_at' to execute.
- *fertilize_incorporate*: boolean (see section 4.1.2)
Parameter (default false)
Incorporate organic fertilizer in plowing zone.
- *tillage*: submodel (see section 4.1.7) 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 14)
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 4.1.7)
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 4.1.7)
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 4.1.7)
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 4.1.7)
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 4.1.7)
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 4.1.7)
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.

14.4 emerge

Force a crop to emerge.

< **emerge** *crop* >

- *crop*: string (see section 4.1.5)
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.

14.5 error

Write a error message to the user.

< **error** *message* >

- *message*: string (see section 4.1.5)
Parameter
Error message to give.

14.6 extern

Select an external scope, and perform action.

```
< extern  scope action  >
```

- *scope*: **scopesel** component (see chapter 64)
Scope to evaluate expressions in.
- *action*: **action** component (see chapter 14)
Action to perform if the condition is false.

14.7 extern_fertigation

Continues 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.

```
< extern_fertigation  (NO3 NO3)
                      (NH4 NH4)
                      (volume volume)           ; Default box value.
                      (scope scope)             ; Default null value.
                      (description description)
                      (cite)
                      (from -10 [cm])
                      (to to)
                      (surface surface)
                      (overhead overhead)
                      (subsoil subsoil) >
```

- *NO3*: **number** component (see chapter 50)
Amount of NO3 in irrigation.
- *NH4*: **number** component (see chapter 50)
Amount of NH4 in irrigation.
- *volume*: **volume** component (see chapter 88)
Component (default 'box')
Soil volume to add irrigation.
- *scope*: **scopesel** component (see chapter 64)
Component (default 'null')
Scope to evaluate expressions in.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

- *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 50)
Amount of surface irrigation applied.
- *overhead*: **number** component (see chapter 50)
Amount of overhead irrigation applied.
- *subsoil*: **number** component (see chapter 50)
Amount of subsoil irrigation applied.

14.8 extern_subsoil

Subsoil irrigation controlled externally.

```
< extern_subsoil (volume volume) ; Default box value.
                  (flux flux)
                  (scope scope)
                  (description description)
                  (cite)
                  (from from)
                  (to to)
                  (constituents constituents ...) >
```

- *volume*: **volume** component (see chapter 88)
Component (default 'box')
Soil volume to add irrigation.
- *flux*: **number** component (see chapter 50)
Amount of irrigation applied.
- *scope*: **scopesel** component (see chapter 64)
Scope to evaluate expressions in.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *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 4.1.5) 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.

14.9 fertilize_base

Shared parameters for all fertilize actions.

```
< fertilize_base (am am)
    (equivalent_weight equivalent_weight)
    (minimum_weight 0 [kg N/ha])
    (precision precision)
    (second_year_compensation false) >
```

- *am*: **am** component (see chapter 16)
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 4.1.7)
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 4.1.2)
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.

14.10 fertilize

A 'fertilize_base' model (see 14.9, page 73) build into Daisy.

Apply fertilizer to the soil surface.

```
< fertilize  am
    (from 0 [cm])
    (to 0 [cm])
    ;; Shared parameters are described in section 14.9.
    (equivalent_weight equivalent_weight)
    (minimum_weight 0 [kg N/ha])
    (precision precision)
    (second_year_compensation false) >
```

- *am*: **am** component (see chapter 16)
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.

14.11 incorporate_fertilizer

A 'fertilize_base' model (see 14.9, page 73) build into Daisy.

Incorporate fertilizer.

```
< incorporate_fertilizer  am
    (volume volume)
    ;; Shared parameters are described in section 14.9.
    (equivalent_weight equivalent_weight)
    (minimum_weight 0 [kg N/ha])
    (precision precision)
    (second_year_compensation false) >
```

- *am*: **am** component (see chapter 16)
The fertilizer you want to apply.
- *volume*: **volume** component (see chapter 88)
Soil volume to incorporate fertilizer in.

14.12 harvest_base

Common parameters for harvest operations.

```
< harvest_base  crop
    (combine false)
    (stub 0 [cm])
    (stem 1 [<fraction>])
    (leaf 1 [<fraction>])
    (sorg 1 [<fraction>]) >
```

- *crop*: string (see section 4.1.5)
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 4.1.2)
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.

14.13 cut

A 'harvest_base' model (see 14.12, page 74) build into Daisy.
Cut a crop.

14.14 harvest

A 'harvest_base' model (see 14.12, page 74) build into Daisy.
Harvest a crop.

14.15 if

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

```
< if  if then else >
```

- *if*: **condition** component (see chapter 26)
Condition determining which action to perform.
- *then*: **action** component (see chapter 14)
Action to perform if the condition is true.
- *else*: **action** component (see chapter 14)
Component (default 'nil')
Action to perform if the condition is false.

14.16 irrigate_base

Shared parameter for irrigate actions.

```
< irrigate_base  flux
                    (solute)
                    (hours hours)
                    (days 0)
                    (remaining_time remaining_time)
                    (temperature temperature) >
```

- *flux*: **number** component (see chapter 50)
Amount of irrigation applied.
- *solute*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Solutes in irrigation water.

```
<  name value  >
```

 - *name*: string (see section 4.1.5)
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).

14.17 irrigate_overhead

A 'irrigate_base' model (see 14.16, page 76) build into Daisy.
Irrigate the field from above.

14.18 irrigate_subsoil

A 'irrigate_base' model (see 14.16, page 76) build into Daisy.
Irrigate the field directly into the soil. Currently, the 'temperature' parameter is ignored.

```

< irrigate_subsoil  flux
                    (volume volume)                                ; Default box value.
                    (from from)
                    (to to)
                    ;; Shared parameters are described in section 14.16.
                    (solute)
                    (hours hours)
                    (days 0)
                    (remaining_time remaining_time)
                    (temperature temperature) >

```

- *flux*: **number** component (see chapter 50)
Amount of irrigation applied.
- *volume*: **volume** component (see chapter 88)
Component (default 'box')
Soil volume to add irrigation.
- *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.

14.19 irrigate_surface

A 'irrigate_base' model (see 14.16, page 76) build into Daisy.

Irrigate the field directly on the soil surface, bypassing the canopy.

14.20 markvand

Irrigate the field according to MARKVAND scheduling.

```

< markvand  (soil soil)
            (flux 2 [mm/h])
            (map map ...)
            (T_sum T_sum)
            (dt dt)
            (V_I 0 [mm])
            (V_r V_r)
            (V_e V_e)
            (C_u 0 [mm])
            (V_u 0 [mm])
            (V_b V_b) >

```

- *soil*: **MV_Soil** component (see chapter 11)
Soil type to schedule irrigation on.
- *flux*: number [**mm/h**]
Parameter (default 2)
Water application speed.

- *map*: submodel (see section 4.1.7) sequence
Map of Daisy crop names into MARKVAND crop descriptions.

```
< Daisy MARKVAND >
```

 - *Daisy*: string (see section 4.1.5)
Parameter
Name of Daisy crop.
 - *MARKVAND*: **MV_Crop** component (see chapter 10)
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 interceptor 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.

14.21 message

Write a message to the user.

```
< message message >
```

- *message*: string (see section 4.1.5)
Parameter
Message to give to the user.

14.22 mix

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

```
< mix   depth
      (penetration 1 [<fraction>]) >
```

- *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.

14.23 disk_harrowing

A ‘mix’ model (see 14.22, page 79) defined in ‘tillage.dai’.

14.24 rotavation

A ‘mix’ model (see 14.22, page 79) defined in ‘tillage.dai’.

14.25 seed_bed_preparation

A ‘mix’ model (see 14.22, page 79) defined in ‘tillage.dai’.

14.26 stubble_cultivation

A ‘mix’ model (see 14.22, page 79) defined in ‘tillage.dai’.

14.27 nil

This action does nothing, always done.

Used by action if else (see 14.15, page 75) .

14.28 panic

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

```
< panic   message >
```

- *message*: string (see section 4.1.5)
Parameter
Error message to give.

14.29 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.

```
< pluck  crop
      (stem 0 [<fraction>])
      (leaf 0 [<fraction>])
      (sorg 1 [<fraction>]) >
```

- *crop*: string (see section 4.1.5)
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.

14.30 progn

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

```
< progn  actions... >
```

- *actions*: **action** component (see chapter 14) sequence
List of actions to perform.

14.31 repeat

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

```
< repeat  repeat
      (do do) >
```

- *repeat*: **action** component (see chapter 14)
Action to perform repeatedly.
- *do*: **action** component (see chapter 14)
Optional component
Action currently being performed.

14.32 ridge

Ridge a ridge on the field.

```
< ridge  ridge >
```


- *ridge*: **Ridge** fixed component (see section 93.12)
Submodel (has partially specified default value)
Ridge parameters

14.33 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.

< **sequence** (do) >

- *do*: **action** component (see chapter 14) sequence
Component (default: an empty sequence)
Sequence of actions to perform.

14.34 activity

A ‘sequence’ model (see 14.33, page 81) build into Daisy.

14.35 set_heat_source

Set external point heat source at height to value.

< **set_heat_source** *height value* >

- *height*: number [**cm**]
Parameter
Height of heat source (a negative number).
- *value*: number [**W/m²**]
Parameter
Value of heat source.

14.36 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.

< **set_porosity** (depth 0 [cm])
(porosity *porosity*) >

- *depth*: number [**cm**]
Parameter (default 0)
A point in the horizon to modify.
- *porosity*: number [<**fraction**>]
Parameter
Non-solid fraction of soil.

14.37 set_surface_detention_capacity

Set amount of ponding the surface can retain.

< **set_surface_detention_capacity** *height* >

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

14.38 sow

Sow a crop on the field.

```
< sow  crop
      (seed seed)
      (row_width 0 [cm])
      (plant_distance plant_distance)
      (row_position 0 [cm])
      (plant_position plant_position) >
```

- *crop*: **crop** component (see chapter 27)
Crop to sow.
- *seed*: number [g w.w./m²]
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 parameter 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 orthogonal 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 parameter 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 orthogonal to the rows as is otherwise assumed by Daisy.

14.39 plant

A 'sow' model (see 14.38, page 82) build into Daisy.
'plant' is another name for 'sow'

14.40 spray

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

```
< spray  chemical amount >
```

- *chemical*: string (see section 4.1.5)
Parameter
Name of pesticide to spray.
- *amount*: number [g/ha]
Parameter
Amount of pesticide to spray.

14.41 stop

Stop the simulation.

14.42 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.

```
< swap (depth depth)
      (middle middle) >
```

- *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.

14.43 plowing

A 'swap' model (see 14.42, page 83) defined in 'tillage.dai'.

14.44 t

This action does nothing, never done.

14.45 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 line in 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 performed.

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

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

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.

```
< table (emerge emerge)
      (harvest harvest)
      (sow sow)
      (file file)
      (volume volume) ; Default box value.
      (flux 2 [mm/h])
      (from -5 [cm])
      (to -25 [cm])
      (missing missing ...) ; Has default value.
      (filter)
      (original original ...)
      (dim_line dim_line)
      (fertilizer fertilizer)
      (enable_irrigation true)
      (enable_fertilization true) >
```

- *emerge*: **action** component (see chapter 14)
Optional component
Emerge action.
- *harvest*: **action** component (see chapter 14)
Optional component
Harvest action.
- *sow*: **action** component (see chapter 14)
Optional component
Sow action.
- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *volume*: **volume** component (see chapter 88)
Component (default 'box')
Soil volume to add irrigation.
- *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 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:
List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

```
< tag allowed... >
```

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 16)
Optional component
The fertilizer you want to apply.
- *enable_irrigation*: boolean (see section 4.1.2)
Parameter (default true)
Set this to false to ignore any irrigation information in the file.
- *enable_fertilization*: boolean (see section 4.1.2)
Parameter (default true)
Set this to false to ignore any fertilization information in the file.

14.46 wait

Wait until the specified condition is true.

```
< wait condition >
```

- *condition*: **condition** component (see chapter 26)
Condition to wait for.

14.47 wait_mm_dd

Wait until a specific month and day in the year.

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

- *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.

14.48 wait_period

Waits the specified period.

```
< wait_period  (hours hours)
               (days days)
               (end_time end_time) >
```

- *hours*: integer
Parameter
Wait this number of hours.
- *days*: integer
Parameter
Wait this number of days.
- *end_time*: **Time** fixed component (see section 93.21)
Optional submodel
Wait until this date. Setting this overrides the 'days' and 'hours' parameters.

14.49 wait_days

A 'wait_period' model (see 14.48, page 86) build into Daisy.

Waits the specified number of days.

14.50 wait_hours

A 'wait_period' model (see 14.48, page 86) build into Daisy.

Waits the specified number of hours.

14.51 warning

Write a warning to the user.

```
< warning  message >
```

- *message*: string (see section 4.1.5)
Parameter
Warning to give to the user.

14.52 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.

```
< while actions... >
```

- *actions*: **action** component (see chapter 14) sequence
List of actions to perform.

14.53 with-column

Perform actions on a specific column.

```
< "with-column" column actions... >
```

- *column*: string (see section 4.1.5)
Parameter
Name of column to perform actions on.
- *actions*: **action** component (see chapter 14) sequence
Actions to perform on the specified column.

Chapter 15

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.

15.1 Freundlich

$M = \text{rho } K C^m + \text{Theta } C$
`< Freundlich (m m)`
 (K_clay K_clay)
 (K_OC K_OC) >

- *m*: number (dimensionless)
 Parameter
 Freundlich parameter
- *K_clay*: number $[(\text{g}/\text{cm}^3)^{-m}]$
 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 $[(\text{g}/\text{cm}^3)^{-m}]$
 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.

15.2 Langmuir

$M = \text{rho } (\text{my_max } C) / (K + C) + \text{Theta } C$
`< Langmuir (K K)`
 (my_max_clay my_max_clay)
 (my_max_OC my_max_OC) >

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

15.3 full

Full adsorption. Used for non-solutes, fully adsorped in the soil.

Used by chemical solid adsorption (see 22.20, page 132) .

15.4 linear

$M = \rho K C + \Theta C$

Used by chemical NH_4 adsorption (see 22.18, page 132) , chemical Atrazine adsorption (see 22.7, page 130) , chemical Bentazon adsorption (see 22.8, page 131) , chemical IPU adsorption (see 22.9, page 131) , chemical MCPP adsorption (see 22.11, page 131) , chemical Pendimethalin adsorption (see 22.12, page 131) , chemical Ioxynil adsorption (see 22.10, page 131) , chemical 2,4-D adsorption (see 22.6, page 130) , chemical Heptachlor adsorption (see 22.15, page 131) , and chemical DDT adsorption (see 22.14, page 131) .

```
< linear (K_clay K_clay)
        (K_OC K_OC) >
```

- *K_clay*: number [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 [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'.

15.5 NH_4

A 'linear' model (see 15.4, page 90) build into Daisy.

Adsorption of ammonium.

15.6 none

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

Used by chemical default adsorption (see 22.1, page 121) .

15.7 vS_S

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

Chapter 16

am

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

16.1 base

Common attributes for all added organic matter models.

```
< base (name name)  
      (initialized false)  
      (creation creation)  
      (lock lock)  
      (om om ...) >
```

- *name*: string (see section 4.1.5)
Optional state variable
A name given to this AOM so you can identify it in for example log files.
- *initialized*: boolean (see section 4.1.2)
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 93.21)
Optional submodel
Time this AM was created.
- *lock*: submodel (see section 4.1.7)
Optional submodel
This AM belongs to a still living plant

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

 - *crop*: string (see section 4.1.5)
State variable
Crop to which this am is locked
 - *part*: string (see section 4.1.5)
State variable
Crop part to which this am is locked
- *om*: **AOM** component (see chapter 8) sequence
Optional component
The individual AOM pools.

16.2 initial

A ‘base’ model (see 16.1, page 93) build into Daisy.

Initial added organic matter at the start of the simulation.

```
< initial (layers layers ...)
          ;; Shared parameters are described in section 16.1.
          (name name)
          (initialized false)
          (creation creation)
          (lock lock)
          (om om ...) >
```

- *layers*: submodel (see section 4.1.7) 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/m²]
Parameter
Carbon in this layer.

16.3 organic

A ‘base’ model (see 16.1, page 93) build into Daisy.

Organic fertilizer, typically slurry or manure from animals.

```
< organic (description description)
          (cite)
          (weight 0 [Mg w.w./ha])
          (total_C_fraction total_C_fraction)
          (total_N_fraction total_N_fraction)
          (second_year_utilization second_year_utilization)
          (first_year_utilization first_year_utilization)
          (dry_matter_fraction dry_matter_fraction)
          (NO3_fraction 0 [<fraction>])
          (NH4_fraction 0 [<fraction>])
          (volatilization 0 [<fraction>])
          ;; Shared parameters are described in section 16.1.
          (name name)
          (initialized false)
          (creation creation)
          (lock lock)
          (om om ...) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

- *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.

16.4 slurry

A ‘organic’ model (see 16.3, page 94) defined in ‘fertilizer.dai’.

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

16.5 Foulum_slurry

A ‘slurry’ model (see 16.4, page 95) defined in ‘fertilizer.dai’.

Parameters provided by DJF in Foulum, 1998. Added by <sha@kvl.dk>, 2000.

16.6 aender_gaes_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.7 cattle_slurry

A ‘slurry’ model (see 16.4, page 95) defined in ‘fertilizer.dai’.
Numbers found by <hsv@kv1.dk>, 2001.

16.8 faar_geder_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.9 hest_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.10 hoens_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.11 hoens_gylle

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.12 hoens_staldgoedning

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.13 kalkuner_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.14 kartoffelfrugtsaft

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.
Dansk Procesteknologi I/S (2003). RAPPORT TIL KARTOFFELAFGIFTSFONDEN: KALIUM I FRUGTSAFT - OVERSIGT. LSNINGSMETODER M.V. (Feb. 2003)

16.15 komposteret_husholdningsaffald

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

Jens Petersen (2001). Gdningsvrði 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>

16.16 kvaeg_ajle

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.17 kvaeg_dybstroelse

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.18 kvaeg_gylle

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.19 kvaeg_staldgoedning

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.20 kvaeggylle_separeret_tyk

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.21 kvaeggylle_separeret_tynd

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.

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

16.22 kvaeggylle_useparereret

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.23 manure

A ‘slurry’ model (see 16.4, page 95) defined in ‘fertilizer.dai’.

We have no specific information about manure.

16.24 cattle_manure

A ‘manure’ model (see 16.23, page 98) defined in ‘fertilizer.dai’.

Numbers found by <hsv@kvl.dk>, 2001.

16.25 horse_manure

A ‘manure’ model (see 16.23, page 98) defined in ‘fertilizer.dai’.

Numbers found by <hsv@kvl.dk>, 2001.

16.26 pig_manure

A ‘manure’ model (see 16.23, page 98) defined in ‘fertilizer.dai’.

Numbers found by <hsv@kvl.dk>, 2001.

16.27 mink_raev_gylle

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.28 mink_raev_staldgoedning

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.29 novogro-30_slam

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

Novozymes, Kalundborg: green accounts. <http://www.novonordisk.com/reports/press/environmental/er97/s>

16.30 novogro_slam

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

Novozymes, Kalundborg: green accounts. <http://www.novonordisk.com/reports/press/environmental/er97/s>

16.31 pig_slurry

A 'slurry' model (see 16.4, page 95) defined in 'fertilizer.dai'.
Numbers found by <hsv@kvl.dk>, 2001.

16.32 pressesaft_groentpillefabrik

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Albert Baumann (2003) Pers. medd. se side (Grntpiller) <abdangr@post.tele.dk>

16.33 slagtekyllinger_dybstroelse

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.34 slagtesvin_ajle

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.35 slagtesvin_dybstroelse

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.36 slagtesvin_gylle

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.37 slagtesvin_staldgoedning

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.38 soer_ajle

A 'slurry' model (see 16.4, page 95) defined in 'dk-fertilizer.dai'.
Baseret p DJF-rapport nr. 36. Kvlstof, fosfor og kalium i husdyrgdning - normal 2000.

16.39 soer_dybstroelse

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.40 soer_gylle

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.41 soer_staldgoedning

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

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

16.42 spildevandsslam

A ‘slurry’ model (see 16.4, page 95) 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

16.43 svinegylle_separeret_tyk

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringstoffraktioner fra separeringsprodukter. Indlg p Efterskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separeret kv- og svinegylle. Specialeprojekt, KVL.

16.44 svinegylle_separeret_tynd

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringstoffraktioner fra separeringsprodukter. Indlg p Efterskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separeret kv- og svinegylle. Specialeprojekt, KVL.

16.45 svinegylle_usepareret

A ‘slurry’ model (see 16.4, page 95) defined in ‘dk-fertilizer.dai’.

Peter Srensen og Mette Hjby Larsen (2002). Udnyttelse af nringstoffraktioner fra separeringsprodukter. Indlg p Efterskonferencen 2002; Mette Hjby Larsen (2002) Kvlstof mineralisering af separeret kv- og svinegylle. Specialeprojekt, KVL.

16.46 root

A ‘base’ model (see 16.1, page 93) build into Daisy.

Initialization of old root remains.

Used by organic default am (see 51.1, page 275) .

```
< root  (depth depth)
        (weight 1.2 [Mg DM/ha])
        (om om ...) ; Has default value.
        (dist 7 [cm])
        (total_C_fraction 0.4 [<fraction>])
        (total_N_fraction 0.01 [<fraction>])
        ;; Shared parameters are described in section 16.1.
        (name name)
        (initialized false)
        (creation creation)
        (lock lock) >
```

- *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

16.47 state

A ‘base’ model (see 16.1, page 93) build into Daisy.

Most AM models are only used for initialization, they will be converted 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.

16.48 mineral

Mineral fertilizer.

```
< mineral (description description)
        (cite)
        (weight 0 [kg N/ha])
        (NH4_fraction NH4_fraction)
        (volatilization 0 [<fraction>]) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *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.

16.49 Ammonia

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Pure NH₄. Added by <sha@kvl.dk>, 2000.

16.50 AmmoniumNitrate

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
A 50-50 mix of NH₄ and NO₃. Added by <sha@kvl.dk>, 2000.

16.51 CalciumNitrate

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Kalksalpeter

16.52 N25S

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
From _Kemira ’97_. Added by <sha@kvl.dk>, 2000.

16.53 NP

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Typical NP or NPS fertilizer

16.54 NPK01

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Various NPK and NS fertilizers

16.55 NPK02

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Various NPK fertilizers

16.56 Nitrate

A ‘mineral’ model (see 16.48, page 101) defined in ‘fertilizer.dai’.
Pure NO₃.

Chapter 17

average

Find the average of two numbers.

17.1 arithmetic

Arithmetic average $'(a+b)/2'$.

Used by uz1d richards `K_average` (see 83.2, page 431) .

17.2 geometric

Geometric average $'\sqrt{a*b}'$.

17.3 harmonic

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

Chapter 18

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.

18.1 default

The default bioclimate model.

Used by column default Bioclimate (see 24.1, page 138) .

```
< default  (svat svat)                                ; Default none value.
            (pet pet)                                    ;
            (difrad difrad)                                ;
            (raddist raddist)                              ; Default default value.
            (Snow Snow)                                    ; Has default value.
            (net_radiation net_radiation)                 ; Default brunt value.
            (NoOfIntervals 30)
            (irrigation_subsoil_permanent 0 [mm/h])
            (canopy_water_storage 0 [mm])
            (litter_water_storage 0 [mm])
            (max_svat_iterations 100)
            (max_svat_absolute_difference 0.01 [mm/h])
            (maxTdiff 1 [K])
            (maxEdiff 5 [Pa])
            (min_sun_angle 0.0628319 [rad]) >
```

- *svat*: **svat** component (see chapter 74)
Component (default 'none')
Soil Vegetation Atmosphere component.
- *pet*: **pet** component (see chapter 53)
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 data.

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 29)
Optional component
Diffuse radiation component.

By default, choose depending on available climate data.

If diffuse radiation is available in the climate data, Daisy will use these (the weather difrad model). Otherwise Daisy will use the DPF model.
- *raddist*: **raddist** component (see chapter 58)
Component (default 'default')
Radiation distribution model.
- *Snow*: **Snow** fixed component (see section 93.5)
Submodel (has fully specified default value)
Surface snow pack.
- *net_radiation*: **net_radiation** component (see chapter 48)
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 convergence.
- *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 potential 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 [W/m²]
Diffuse radiation above canopy.
- *total_PAR*: number [W/m²] canopy boundaries
Total PAR between canopy layers.
- *sun_PAR*: number [W/m²] canopy boundaries
Sun PAR between canopy layers.
- *total_NIR*: number [W/m²] canopy boundaries
Total NIR between canopy layers.
- *sun_NIR*: number [W/m²] canopy boundaries
Sun NIR between canopy layers.
- *sun_LAI_fraction*: number [<fraction>] canopy intervals
Sunlit LAI in canopy layers.
- *absorbed_total_PAR_canopy*: number [W/m²]
Canopy absorbed PAR (sun+shade)
- *absorbed_total_NIR_canopy*: number [W/m²]
Canopy absorbed NIR (sun+shade)
- *absorbed_total_Long_canopy*: number [W/m²]
Canopy absorbed long wave radiation (sun+shade)
- *absorbed_total_PAR_soil*: number [W/m²]
Soil absorbed PAR (sun+shade)

- *absorbed_total_NIR_soil*: number [\mathbf{W}/\mathbf{m}^2]
Soil absorbed NIR (sun+shade)
- *absorbed_total_Long_soil*: number [\mathbf{W}/\mathbf{m}^2]
Soil absorbed long wave radiation (sun+shade)
- *absorbed_sun_PAR_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed PAR on sunlit leaves
- *absorbed_sun_NIR_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed NIR on sunlit leaves
- *absorbed_sun_Long_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed long wave radiatio on sunlit leaves
- *absorbed_shadow_PAR_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed PAR on shadow leaves
- *absorbed_shadow_NIR_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed NIR on shadow leaves
- *absorbed_shadow_Long_canopy*: number [\mathbf{W}/\mathbf{m}^2]
Canopy absorbed long wave radiation on shadow leaves
- *incoming_Long_radiation*: number [\mathbf{W}/\mathbf{m}^2]
Incoming longwave radiation
- *incoming_PAR_radiation*: number [\mathbf{W}/\mathbf{m}^2]
Incoming PAR radiation
- *incoming_NIR_radiation*: number [\mathbf{W}/\mathbf{m}^2]
Incoming NIR radiation
- *incoming_Total_radiation*: number [\mathbf{W}/\mathbf{m}^2]
Incoming radiation, sum of shortwave and longwave

Chapter 19

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 50)
Biopore density [cm^{-2}] as a function of 'x' [cm].
- *diameter*: number [**cm**]
Parameter
Biopore diameter.

Log Variables

- *S*: number [**cm³/cm³/h**] soil cells
Total stream from matrix domain to biopore.
- *M2B*: number [**cm³/cm³/h**] soil cells
Stream from matrix domain to biopore. Never negative.
- *B2M*: number [**cm³/cm³/h**] soil cells
Stream from biopore to matrix domain. Never negative.
- *infiltration*: number [**cm/h**]
Surface infiltration.
- *solute_infiltration*: submodel (see section 4.1.7) sequence
Rate of solute infiltration through surface.

```
< name value > Log Variables
```

- *value*: number [**g/cm²/h**]
Value for chemical.
- *name*: string (see section 4.1.5)
Name of chemical.

19.1 drain

Biopores that ends in the drain pipes.

```
< drain (pipe_position pipe_position)
      ;; Shared parameters are described in chapter 19.
      (height_start height_start)
      (height_end height_end)
      (density density)
      (diameter diameter) >
```

- *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.

19.2 matrix

Biopores that ends in the matrix.

```
< matrix (solute)
      (debug 0)
      (max_iterations 50)
      (max_absolute_difference 0.02 [cm])
      (max_relative_difference 0.001 [])
      (xplus xplus ...)
      (K_wall_relative K_wall_relative)
      (h_bottom h_bottom ...)
      (allow_upward_flow true)
      ;; Shared parameters are described in chapter 19.
      (height_start height_start)
      (height_end height_end)
      (density density)
      (diameter diameter) >
```

- *solute*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Chemical concentration in biopore intervals.

```
< name value... >
```

- *name*: string (see section 4.1.5)

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 4.1.2)
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 [**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.

Chapter 20

boolean

Generic representation of booleans.

Used by `StringerCondClause @ condition` (see 93.30, page 518) , and `IntegerCondClause @ condition` (see 93.31, page 518) .

20.1 false

Always false.

20.2 not

True if and only if the operand is not true.

< `not` *operands...* >

- *operands*: **boolean** component (see chapter 20) array of length 1
The operand to check.

20.3 numbers

Base class for boolean expressions involving numbers.

< `numbers` *operands...* >

- *operands*: **number** component (see chapter 50) sequence
List of operands to compare.

20.4 <

A ‘numbers’ model (see 20.3, page 117) build into Daisy.

True iff each operand is smaller than the next.

20.5 <=

A ‘numbers’ model (see 20.3, page 117) build into Daisy.

True iff each operand is smaller than or equal to the next.

20.6 >

A ‘numbers’ model (see 20.3, page 117) build into Daisy.
True iff each operand is larger than the next.

20.7 >=

A ‘numbers’ model (see 20.3, page 117) build into Daisy.
True iff each operand is at least as large as the next.

20.8 operands

Base class for boolean expressions involving multiple boolean operands.

< **operands** *operands...* >

- *operands*: **boolean** component (see chapter 20) sequence
List of operands to compare.

20.9 and

A ‘operands’ model (see 20.8, page 118) build into Daisy.
True if and only if all operands are true.

20.10 or

A ‘operands’ model (see 20.8, page 118) build into Daisy.
True if and only if any operand is true.

20.11 string-equal

True iff the supplied strings are identical.

< "string-equal" *values...* >

- *values*: string (see section 4.1.5) sequence
Parameter
Strings to compare.

20.12 true

Always true.

Used by source arithmetic valid (see 70.1, page 365) , and xysource arithmetic valid (see 91.1, page 463) .

20.13 xor

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

< **xor** *operands...* >

- *operands*: **boolean** component (see chapter 20) array of length 2
The two operands to compare.

Chapter 21

bound

Specify one end of an interval boundary.

21.1 finite

Finite interval bound.

```
< finite bound >
```

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

21.2 full

Maximum value for the interval boundary.

21.3 none

No boundary specified.

21.4 state

Bound used for checkpoints.

Used by volume box bottom (see 88.1, page 447) , and zone box bottom (see 92.1, page 475) .

```
< state (bound bound)
      (type type) >
```

- *bound*: number [cm]
Optional state variable
Interval bound to use. Only valid for the 'finite' type.
- *type*: string (see section 4.1.5)
State variable
Bound type

21.5 empty

A 'state' model (see 21.4, page 119) build into Daisy.

A 'state' model set to 'none'.

Chapter 22

chemical

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

22.1 default

Read chemical properties as normal Daisy parameters.

```

< default  (initial_C initial_C ...)
            (adsorption adsorption)                                ; Def
            (initial initial)                                       ; Def
            (description description)
            (cite)
            (C C ...)
            (solubility 1 [g/cm3])
            (solubility_infiltration_factor 1 [])
            (crop_uptake_reflection_factor 1 [<fraction>])
            (canopy_dissipation_rate canopy_dissipation_rate)
            (canopy_dissipation_halftime canopy_dissipation_halftime)
            (canopy_dissipation_rate_coefficient canopy_dissipation_rate_coefficient)
            (canopy_washoff_coefficient canopy_washoff_coefficient)
            (surface_decompose_rate surface_decompose_rate)
            (surface_decompose_halftime surface_decompose_halftime)
            (diffusion_coefficient diffusion_coefficient)
            (decompose_rate decompose_rate)
            (decompose_halftime decompose_halftime)
            (decompose_heat_factor decompose_heat_factor)           ; Has
            (decompose_water_factor decompose_water_factor)         ; Has
            (decompose_CO2_factor decompose_CO2_factor)             ; Has
            (decompose_conc_factor decompose_conc_factor)           ; Has
            (decompose_depth_factor decompose_depth_factor)         ; Has
            (decompose_lag_increment decompose_lag_increment)       ; Has
            (drain_secondary false)
            (C_below C_below)                                       ; Def
            (decompose_products)
            (snow_storage 0 [g/m2])
            (canopy_storage 0 [g/m2])
            (litter_storage 0 [g/m2])
            (surface_storage 0 [g/m2])
            (C_secondary C_secondary ...)
            (initial_C_secondary initial_C_secondary ...)
            (initial_C_primary initial_C_primary ...)
            (M M ...)
            (initial_M initial_M ...)
            (initial_M_secondary initial_M_secondary ...)
            (initial_M_primary initial_M_primary ...)
            (Ms Ms ...)
            (initial_Ms initial_Ms ...)
            (S_permanent)
            (lag lag ...) >

```

- *initial_C*: submodel (see section 4.1.7) 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 15)
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 50)
Component (default 'const')

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

Parameter description:

Initial content (M) if otherwise unspecified. [g/cm^3].

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *C*: number [g/cm^3] soil cells
Optional state variable
Concentration in water.
- *solubility*: number [g/cm^3]
Parameter (default 1)
Maximal concentration in water at 20 dg C.
- *solubility_infiltration_factor*: number (dimensionless)
Parameter (default 1)
Adjustment for maximum concentration in infiltrated water.
- *crop_uptake_reflection_factor*: number [fraction]
Parameter (default 1)
How much of the chemical is reflected at crop uptake.
- *canopy_dissipation_rate*: number [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 [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 [**h⁻¹**]
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.
- *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 [**cm²/s**]
Parameter
Diffusion coefficient.
- *decompose_rate*: number [**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/cm³/h** → **<none>**]
Parameter (has default value with 0 points)
CO2 development factor on decomposition.
- *decompose_conc_factor*: plf [**g/cm³ 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/cm³ → 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 4.1.2)
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 in the matrix is used. Using the secondary domain is more physically correct, but also more likely to give unstable results.
- *C_below*: **number** component (see chapter 50)
Component (default 'const')

(C_below const -1 [g/cm³])

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 4.1.7) sequence
Submodel (default: an empty sequence)
List of products from decomposition.

 < *fraction chemical* >
 – *fraction*: number [<fraction>]
 Parameter
 Fraction of decomposed matter that become this chemical.
 – *chemical*: string (see section 4.1.5)
 Parameter
 Chemical product of decomposed matter.
- *snow_storage*: number [g/m²]
State variable (default 0)
Stored in the snow pack.
- *canopy_storage*: number [g/m²]
State variable (default 0)
Stored on the canopy.
- *litter_storage*: number [g/m²]
State variable (default 0)
Stored in the litter (mulch, surface residuals).
- *surface_storage*: number [g/m²]
State variable (default 0)
Stored on the soil surface. This includes the mixing layer, and constitute 'surface_solute' and 'surface_immobile'.
- *C_secondary*: number [g/cm³] soil cells
Optional state variable
Concentration in secondary domain.
- *initial_C_secondary*: submodel (see section 4.1.7) 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 [**g/cm³**]
Parameter
Concentration in secondary domain.
- *initial_C_primary*: submodel (see section 4.1.7) 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/cm³**]
Parameter
Concentration in primary domain.
- *M*: number [**g/cm³**] soil cells
Optional state variable
Total mass per volume water, soil, and air.
- *initial_M*: submodel (see section 4.1.7) 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/cm³**]
Parameter
Total mass per volume water, soil, and air.
- *initial_M_secondary*: submodel (see section 4.1.7) sequence
Optional submodel
Initial value of the 'M_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 [**g/cm³**]
Parameter
Secondary domain mass per volume water, soil, and air.

- *initial_M_primary*: submodel (see section 4.1.7) 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/cm³]
 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 4.1.7) 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/cm³/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/m²/h]
Amount currently being applied.
- *tillage*: number [g/cm³/h] soil cells
Changes during tillage.
- *dt*: number [h]
Suggested timestep length based on sink terms.

- *deposit*: number $[\text{g}/\text{m}^2/\text{h}]$
Amount deposited from the atmosphere.
- *surface_tillage*: number $[\text{g}/\text{m}^2/\text{h}]$
Amount removed from surface due to tillage operations.
- *litter_tillage*: number $[\text{g}/\text{m}^2/\text{h}]$
Amount removed from litter due to tillage operations.
- *snow_in*: number $[\text{g}/\text{m}^2/\text{h}]$
Entering snow pack.
- *snow_out*: number $[\text{g}/\text{m}^2/\text{h}]$
Leaking from snow pack.
- *canopy_in*: number $[\text{g}/\text{m}^2/\text{h}]$
Entering canopy.
- *canopy_dissipate*: number $[\text{g}/\text{m}^2/\text{h}]$
Dissipating from canopy.
- *canopy_out*: number $[\text{g}/\text{m}^2/\text{h}]$
Falling through or off the canopy.
- *canopy_harvest*: number $[\text{g}/\text{m}^2/\text{h}]$
Amount removed with crop harvest.
- *litter_in*: number $[\text{g}/\text{m}^2/\text{h}]$
Entering litter .
- *litter_decompose*: number $[\text{g}/\text{m}^2/\text{h}]$
Decomposed from the litter.
- *litter_out*: number $[\text{g}/\text{m}^2/\text{h}]$
Leaking from litter.
- *surface_solute*: number $[\text{g}/\text{m}^2]$
Stored in the soil water of the mixing layer. This is part of 'surface_storage'.
- *surface_immobile*: number $[\text{g}/\text{m}^2]$
Bound to soil particles in the mixing layer. This is part of 'surface_storage'.
- *surface_in*: number $[\text{g}/\text{m}^2/\text{h}]$
Falling on the bare soil surface.
- *surface_runoff*: number $[\text{g}/\text{m}^2/\text{h}]$
Removed through lateral movement on the soil.
- *surface_decompose*: number $[\text{g}/\text{m}^2/\text{h}]$
Decomposed from the surface.
- *surface_transform*: number $[\text{g}/\text{m}^2/\text{h}]$
Added through transformation to surface.
- *surface_mixture*: number $[\text{g}/\text{m}^2/\text{h}]$
Entering the soil through mixture with ponded water.
- *surface_out*: number $[\text{g}/\text{m}^2/\text{h}]$
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 use it.

Note that the value is relative to the current timestep.
- *top_storage*: number [**g/m²**]
Sum of above ground (surface, liter, snow, canopy) storage.
- *top_loss*: number [**g/m²/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/cm³**] soil cells
Concentration in primary domain.
- *M_secondary*: number [**g/cm³**] soil cells
Mass in secondary domain.
- *M_primary*: number [**g/cm³**] soil cells
Primary domain mass per volume water, soil, and air.
- *M_error*: number [**g/cm³**] soil cells
Mass subtracted to avoid negative values.
- *S_secondary*: number [**g/cm³/h**] soil cells
Secondary matrix source term.
- *S_primary*: number [**g/cm³/h**] soil cells
Primary matrix source term.
- *S_tertiary*: number [**g/cm³/h**] soil cells
Source term for tertiary (macropore) domain.
- *S_exchange*: number [**g/cm³/h**] soil cells
Exchange from primary to secondary domain.
- *S_drain*: number [**g/cm³/h**] soil cells
Source term (soil drainage only).
- *S_external*: number [**g/cm³/h**] soil cells
External source, such as incorporated fertilizer.
- *S_root*: number [**g/cm³/h**] soil cells
Source term (root uptake only, always negative).
- *S_decompose*: number [**g/cm³/h**] soil cells
Source term for decompose, is never positive.
- *S_decompose_primary*: number [**g/cm³/h**] soil cells
Source term for decompose in primary domain, is never positive.
- *S_decompose_secondary*: number [**g/cm³/h**] soil cells
Source term for decompose in secondary domain, is never positive.

- *S_transform*: number $[\text{g}/\text{cm}^3/\text{h}]$ soil cells
Source term for transformations other than sorption.
- *J_primary*: number $[\text{g}/\text{cm}^2/\text{h}]$ soil edges
Transportation in primary matrix water (positive up).
- *J_secondary*: number $[\text{g}/\text{cm}^2/\text{h}]$ soil edges
Transportation in secondary matrix water (positive up).
- *J_matrix*: number $[\text{g}/\text{cm}^2/\text{h}]$ soil edges
Transportation in matrix (positive up).
- *J_tertiary*: number $[\text{g}/\text{cm}^2/\text{h}]$ soil edges
Transportation in tertiary water (positive up).
- *sink_cell*: integer
Cell with largest forward sink compared to available matter.

22.2 common

A ‘default’ model (see 22.1, page 121) defined in ‘chemistry-base.dai’.
For chemicals where we know no better. See also [FOCUS, 2002]

22.3 pesticide

A ‘common’ model (see 22.2, page 130) defined in ‘chemistry-base.dai’.
This stuff protects plants. See also [FOCUS, 2002]

22.4 fungicide

A ‘pesticide’ model (see 22.3, page 130) defined in ‘chemistry-base.dai’.
This stuff kills swamps. See also [FOCUS, 2002]

22.5 herbicide

A ‘pesticide’ model (see 22.3, page 130) defined in ‘chemistry-base.dai’.
This stuff kills plants. See also [FOCUS, 2002]

22.6 2,4-D

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Selective herbicide, kill weeds but not crops. See also [FOCUS, 2002]
Used by chemistry pesticides trace (see 23.3, page 133) .

22.7 Atrazine

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.8 Bentazon

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.9 IPU

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.10 Ioxynil

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.11 MCPP

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.12 Pendimethalin

A ‘herbicide’ model (see 22.5, page 130) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.13 insecticide

A ‘pesticide’ model (see 22.3, page 130) defined in ‘chemistry-base.dai’.
This stuff kills insects. See also [FOCUS, 2002]

22.14 DDT

A ‘insecticide’ model (see 22.13, page 131) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.15 Heptachlor

A ‘insecticide’ model (see 22.13, page 131) defined in ‘chemistry.dai’.
Used by chemistry pesticides trace (see 23.3, page 133) .

22.16 nutrient

A ‘default’ model (see 22.1, page 121) build into Daisy.
Plants eat this stuff.

22.17 N

A ‘nutrient’ model (see 22.16, page 131) build into Daisy.
Non-organic nitrogen.

22.18 NH₄

A 'N' model (see 22.17, page 131) build into Daisy.

Ammonium-N.

Used by chemistry N trace (see 23.2, page 133) .

22.19 NO₃

A 'N' model (see 22.17, page 131) build into Daisy.

Nitrate-N.

Used by chemistry N trace (see 23.2, page 133) .

22.20 solid

A 'default' model (see 22.1, page 121) defined in 'chemistry-base.dai'.

Non-dissolvable chemicals

Chapter 23

chemistry

Pesticides and other chemicals.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

23.1 default

Handle chemicals and reactions.

```
< default (reaction)  
          (trace)  
          ;; Shared parameters are described in chapter 23.  
          (description description)  
          (cite) >
```

- *reaction*: **reaction** component (see chapter 60) sequence
Component (default: an empty sequence)
List of chemical reactions you want to simulate.
- *trace*: **chemical** component (see chapter 22) sequence
Component (default: an empty sequence)
List of chemicals you want to trace in the simulation.

23.2 N

A ‘default’ model (see 23.1, page 133) build into Daisy.

Inorganic nitrogen.

Used by chemistry nutrient combine (see 23.6, page 135) .

23.3 pesticides

A ‘default’ model (see 23.1, page 133) defined in ‘chemistry.dai’.

Trace all known pesticides.

23.4 multi

Handle multile chemistries.

Used by column default Chemistry (see 24.1, page 138) .

```
< multi  (combine combine ...)
          (ignore)
          (max_sink_total 0.5 [])
          (max_sink_solute 0.9 [])
          (max_sink_secondary 1.5 [])
          (min_sink_total 0.01 [])
          ;; Shared parameters are described in chapter 23.
          (description description)
          (cite) >
```

- *combine*: **chemistry** component (see chapter 23) sequence
List of chemistry parameterizations you want to combine.
- *ignore*: string (see section 4.1.5) 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 22) sequence
List of chemicals in nested chemistries.

23.5 none

A ‘multi’ model (see 23.4, page 134) build into Daisy.

No active chemistries.

23.6 nutrient

A ‘multi’ model (see 23.4, page 134) build into Daisy.

Include ‘N’ chemistry so organic matter and plants will work.

Chapter 24

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 [m2])  
            (location) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *area*: number [m²]
State variable (default 1)
Area covered by this column. When logging multiple columns, the values are weighted by relative area.
- *location*: submodel (see section 4.1.7) 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.

24.1 default

Hansen et.al. 1990. with generic movement in soil.

```
< default  (weather weather)
            (scope scope)                ; Default null value.
            (SoilHeat SoilHeat)           ; Has default value.
            (SoilWater SoilWater)         ; Has default value.
            (Surface Surface)             ; Has default value.
            (Soil Soil)                   ; Has partial value.
            (Irrigation Irrigation)        ; Has default value.
            (Movement Movement)          ; Default vertical value.
            (Vegetation Vegetation)       ; Default crops value.
            (Litter Litter)               ; Default none value.
            (Bioclimate Bioclimate)       ; Default default value.
            (Drain Drain)                 ; Default none value.
            (Groundwater Groundwater)
            (Chemistry Chemistry)         ; Default multi value.
            (OrganicMatter OrganicMatter) ; Default default value.
            (second_year_utilization 0 [kg N/ha])
            (tillage_age tillage_age ...)
            ;; Shared parameters are described in chapter 24.
            (description description)
            (cite)
            (area 1 [m2])
            (location) >
```

- *weather*: **weather** component (see chapter 89)
Optional component
Weather model for providing climate information during the simulation. If unspecified, used global weather.
- *scope*: **scopesel** component (see chapter 64)
Component (default ‘null’)
Scope to evaluate expressions in.
- *SoilHeat*: **SoilHeat** fixed component (see section 93.6)
Submodel (has fully specified default value)
Soil heat capacity and transportation.
- *SoilWater*: **SoilWater** fixed component (see section 93.7)
Submodel (has fully specified default value)
Soil water content and transportation.
- *Surface*: **Surface** fixed component (see section 93.8)
Submodel (has fully specified default value)
The upper border of the soil.
- *Soil*: **Soil** fixed component (see section 93.11)
Submodel (has partially specified default value)
The numeric and physical soil properties.
- *Irrigation*: **Irrigation** fixed component (see section 93.29)
Submodel (has fully specified default value)
Active irrigation events.
- *Movement*: **movement** component (see chapter 47)
Component (default ‘vertical’)
Discretization and movement of water, heat and solutes in the soil.

- *Vegetation*: **vegetation** component (see chapter 86)
Component (default 'crops')
The crops on the field.
- *Litter*: **litter** component (see chapter 43)
Component (default 'none')
The litter layer below the canopy.
- *Bioclimate*: **bioclimate** component (see chapter 18)
Component (default 'default')
The water and energy distribution among the crops.
- *Drain*: **drain** component (see chapter 31)
Component (default 'none')
Drainage.
- *Groundwater*: **groundwater** component (see chapter 38)
The groundwater level.
- *Chemistry*: **chemistry** component (see chapter 23)
Component (default 'multi')

(Chemistry multi (combine N))

Parameter description:
Chemical compounds in the system.

- *OrganicMatter*: **organic** component (see chapter 51)
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 [g/m²/h]
Amount of DM removed by harvest this hour.
- *harvest_N*: number [g/m²/h]
Amount of nitrogen removed by harvest this hour.
- *harvest_C*: number [g/m²/h]
Amount of carbon removed by harvest this hour.
- *residuals_DM*: number [g/m²/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 [$\text{g}/\text{m}^2/\text{h}$]
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 [$\text{g}/\text{m}^2/\text{h}$]
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 [$\text{g}/\text{cm}^3/\text{h}$] 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 [$\text{g}/\text{cm}^3/\text{h}$] 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 [$\text{g}/\text{m}^2/\text{h}$]
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 [$\text{g}/\text{m}^2/\text{h}$]
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.

24.2 Askov

A ‘default’ model (see 24.1, page 138) defined in ‘dk-soil.dai’.
This is a JB6 soil with free drainage macropores.

24.3 Foulum

A ‘default’ model (see 24.1, page 138) defined in ‘dk-soil.dai’.
Provided by DJF as part if the FertOrgaNic project.

24.4 Jyndevad

A ‘default’ model (see 24.1, page 138) defined in ‘dk-soil.dai’.
This is a JB1 soil with free drainage

Chapter 25

condedge

Find the hydraulic conductivity between two cells.

25.1 arithmetic

Use the arithmetic average of the conductivity in the two cells.

Used by uzrect Mollerup `K_average` (see 85.1, page 435) .

25.2 geometric

Geometric average `'sqrt(a*b)'`.

25.3 harmonic

Use harmonic average of the conductivity of the two cells. This corresponds to using the average hydraulic resistance.

25.4 pressure

Pressure dependent average of the two cells. Use harmonic average of the conductivity of unsaturated cells. This corresponds to using the average hydraulic resistance. For saturated cells, water may stream into unsaturated neighbor 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'`.

```
< pressure (h_lim h_lim)
           (allow_sideways true)
           (use_h_old true) >
```

- `h_lim`: number [cm]
Parameter
Lower pressure limit for fast downward flow.
- `allow_sideways`: boolean (see section 4.1.2)
Parameter (default true)
Allow water to flow fast from saturated cells to all neighbor cells. Not just the cell below.

- *use_h_old*: boolean (see section 4.1.2)
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 26

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 67, page 349) .

26.1 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_above  TSum_limit
               (check_hour 6)
               (reset_mday 1)
               (reset_month 3)
               (TSum_now TSum_now) >
```

- *TSum_limit*: number [dg C d]
Parameter
Temperature 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 temperature sum since last reset.

26.2 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.

```
< and  operands...  >
```

- *operands*: **condition** component (see chapter 26) sequence
Conditions to test.

26.3 trafficable

A ‘and’ model (see 26.2, page 144) defined in ‘tillage.dai’.

26.4 check

Test if a boolean expression is true.

```
< check  expr  >
```

- *expr*: **boolean** component (see chapter 20)
Expression to evaluate.

26.5 crop_dm_over

True iff the crop has reached the specified amount of dry matter.

```
< crop_dm_over  crop weight
                (height 0 [cm])  >
```

- *crop*: string (see section 4.1.5)
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.

26.6 crop_dm_sorg_over

True iff the storage organ has reached the specified amount of dry matter.

```
< crop_dm_sorg_over  crop weight  >
```

- *crop*: string (see section 4.1.5)
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.

26.7 crop_ds_after

True iff the crop has reached development stage 'ds'.

```
< crop_ds_after  crop  ds  >
```

- *crop*: string (see section 4.1.5)
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].

26.8 daily

True at the end of each day.

26.9 daily_air_temperature_above

Test if the daily air is warmer than the specified temperature.

```
< daily_air_temperature_above  temperature  >
```

- *temperature*: number [dg C]
Parameter
Lowest air temperature for which the condition is true.

26.10 daily_precipitation_above

Test if the daily precipitation is warmer than the specified value.

```
< daily_precipitation_above  precipitation  >
```

- *precipitation*: number [mm]
Parameter
Lowest precipitation for which the condition is true.

26.11 every

Matches simulation with fixed time intervals.

```
< every  (seconds 0)
         (minutes 0)
         (hours 0)
         (days 0)
         (microseconds 0)
         (next next) >
```

- *seconds*: integer
State variable (default 0)
Number of seconds.
- *minutes*: integer
State variable (default 0)
Number of minutes.

- *hours*: integer
State variable (default 0)
Number of hours.
- *days*: integer
State variable (default 0)
Number of days.
- *microseconds*: integer
State variable (default 0)
Number of microseconds.
- *next*: **Time** fixed component (see section 93.21)
Optional submodel
Time for next match.

26.12 extern

Test if a boolean expression is true, using extern log.

`< extern scope expr >`

- *scope*: **scopesel** component (see chapter 64)
Scope to evaluate expression in.
- *expr*: **boolean** component (see chapter 20)
Expression to evaluate.

26.13 false

Always false.

26.14 finished

True iff the simulation has finished.

Used by log checkpoint when (see 44.1, page 233) .

26.15 hour

True, at the specified hour.

`< hour at >`

- *at*: integer
Parameter
Hour when the condition is true [0-23].

26.16 hourly

True at the end of each hour.

Used by log Soil water when (see 44.10, page 244) , log Field water when (see 44.8, page 240) , log Field nitrogen when (see 44.7, page 238) , log Soil nitrogen when (see 44.9, page 241) , log Field chemical when (see 44.13, page 247) , and log Soil chemical when (see 44.14, page 249) .

26.17 if

If the first condition is true, return the value of the second condition, else return the value of the third condition.

```
< if if then else >
```

- *if*: **condition** component (see chapter 26)
Condition to test for.
- *then*: **condition** component (see chapter 26)
Condition to use if the 'if' test was true.
- *else*: **condition** component (see chapter 26)
Condition to use if the 'if' test was false.

26.18 mday

True, at the specified day in the month.

```
< mday at >
```

- *at*: integer
Parameter
Day in the month when the condition is true [1-31].

26.19 minutely

True at the end of each minute.

26.20 mm_dd_base

Conditions based on month and day.

```
< mm_dd_base month day  
              (hour 8)  
              (minute 0)  
              (second 0) >
```

- *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.

26.21 after_mm_dd

A ‘mm_dd_base’ model (see 26.20, page 147) build into Daisy.
True after specific month, day and hour in the year.

26.22 before_mm_dd

A ‘mm_dd_base’ model (see 26.20, page 147) build into Daisy.
True before specific month, day and hour in the year.

26.23 mm_dd

A ‘mm_dd_base’ model (see 26.20, page 147) build into Daisy.
True a specific month, day and hour in the year.

26.24 month

True, at the specified month.

< month *at* >

- *at*: integer
Parameter
Month when the condition is true [1-12].

26.25 monthly

True at the end of each month.

26.26 not

True iff the operand is not true.

Used by condition trafficable operands (see 26.3, page 144) .

< not *operand* >

- *operand*: **condition** component (see chapter 26)
Condition to test.

26.27 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.

< or *operands...* >

- *operands*: **condition** component (see chapter 26) sequence
Conditions to test.

26.28 periodic

True if more than a specified walltime has passed since last time it was true.

Used by program Daisy print_time (see 57.2, page 301) .

< periodic *period* >

- *period*: integer
Parameter (default 1)
Number of walltime seconds between success.

26.29 running

True iff the simulation is still running.

26.30 secondly

True at the end of each second.

26.31 soil_inorganic_N_above

Test if the soil contains more mineral nitrogen than the specified amount.

< soil_inorganic_N_above *amount*
(from 0 [cm])
(to *to*) >

- *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.

26.32 soil_temperature_above

Test if the soil is warmer than the specified temperature.

Used by condition trafficable operands (see 26.3, page 144) .

< soil_temperature_above (height *height*)
(temperature *temperature*) >

- *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.

26.33 soil_water_content_above

Test if the soil contains more water than the specified amount.

```
< soil_water_content_above  water
                             (from 0 [cm])
                             (to to) >
```

- *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.

26.34 soil_water_pressure_above

Test if the soil is wetter than the specified pressure potential.

```
< soil_water_pressure_above  (height height)
                             (potential 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.

26.35 time

Conditions based on a specific time.

```
< time  time  >
```

- *time*: **Time** fixed component (see section 93.21)
Submodel (has partially specified default value)
Fixed time to test for.

26.36 after

A ‘time’ model (see 26.35, page 150) build into Daisy.

True, iff the simulation time is after the specified time.

26.37 at

A ‘time’ model (see 26.35, page 150) build into Daisy.

True, iff the simulation time is at the specified time.

26.38 before

A ‘time’ model (see 26.35, page 150) build into Daisy.

True, iff the simulation time is before the specified time.

26.39 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.

`< timestep operand timestep >`

- *operand*: **condition** component (see chapter 26)
Condition to use.
- *timestep*: string (see section 4.1.5)
Parameter
Timestep to use.

26.40 true

Always true.

Used by program Daisy activate_output (see 57.2, page 301) .

26.41 weekly

True at the end of each week.

26.42 yday

True, at the specified julian day.

`< yday at >`

- *at*: integer
Parameter
Julian day when the condition is true [1-366].

26.43 year

True, at the specified year.

`< year at >`

- *at*: integer
Parameter
Year when the condition is true.

26.44 yearly

True at the end of each year.

Chapter 27

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 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

27.1 default

Standard Daisy crop model. Hansen, 1999.

```
< default (Harvest Harvest) ; Has default value.  
          (CrpN CrpN) ; Has partial value.  
          (Seed Seed) ; Default LAI value.  
          (Root Root) ; Has default value.  
          (Canopy Canopy) ; Has partial value.  
          (Prod Prod) ; Has partial value.  
          (last_time last_time)  
          (Devel Devel)  
          (Partit Partit) ; Has partial value.  
          (Vernal Vernal) ; Default none value.  
          (LeafPhot LeafPhot) ; Default GL value.  
          (water_stress_effect water_stress_effect)  
          (enable_N_stress enable_N_stress)  
          (min_light_fraction 0 [<fraction>])  
          ;; Shared parameters are described in chapter 27.  
          (description description)  
          (cite) >
```

- *Harvest*: **Harvesting** fixed component (see section 93.16)
Submodel (has fully specified default value)
Harvest parameters.

- *CrpN*: **CrpN** fixed component (see section 93.19)
Submodel (has partially specified default value)
Nitrogen parameters.
- *Seed*: **seed** component (see chapter 66)
Component (default 'LAI')
Initial crop growth.
- *Root*: **RootSystem** fixed component (see section 93.13)
Submodel (has fully specified default value)
Root system.
- *Canopy*: **CanopyStandard** fixed component (see section 93.14)
Submodel (has partially specified default value)
Canopy.
- *Prod*: **Production** fixed component (see section 93.17)
Submodel (has partially specified default value)
Production.
- *last_time*: **Time** fixed component (see section 93.21)
Optional submodel
The time of the previous timestep.
- *Devel*: **phenology** component (see chapter 54)
Development and phenology.
- *Partit*: **Partition** fixed component (see section 93.18)
Submodel (has partially specified default value)
Assimilate partitioning.
- *Vernal*: **vernalization** component (see chapter 87)
Component (default 'none')
Vernalization.
- *LeafPhot*: **photosynthesis** component (see chapter 55)
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 90)
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 4.1.2)
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 minimum 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 55)
Leaf photosynthesis for sunlit leaves. This will be zero if the selected radiation distribution model does not distinguish between sunlit and shadow leaves.
- *reserved*: **photosynthesis** component (see chapter 55)
Leaf photosynthesis for reserved leaves. This is used for simulating "patches" in multi-crop systems, such as a clover-grass mixture. This is controlled by the 'min_light_fraction' parameter.

27.2 Aert

A 'default' model (see 27.1, page 153) defined in 'dk-pea.dai'.

rt med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.3 Beetroot

A 'default' model (see 27.1, page 153) defined in 'beetroot.dai'.

For Aarhus County - Samsøe Project

27.4 Broccoli

A 'default' model (see 27.1, page 153) defined in 'broccoli.dai'.

For Aarhus County - Samsøe Project

27.5 Broccoli - transplanted

A 'Broccoli' model (see 27.4, page 155) defined in 'broccoli.dai'.

27.6 Brussels sprouts

A 'default' model (see 27.1, page 153) defined in 'brusselssprouts.dai'.

For Aarhus County - Samsøe Project

27.7 Brussels sprouts - transplanted

A 'Brussels sprouts' model (see 27.6, page 155) defined in 'brusselssprouts.dai'.

27.8 Celeriac

A 'default' model (see 27.1, page 153) defined in 'celeriace.dai'.

For Aarhus County - Samsøe Project

27.9 Celeriac - transplanted

A ‘Celeriac’ model (see 27.8, page 155) defined in ‘celeriace.dai’.

27.10 Fodder Beet

A ‘default’ model (see 27.1, page 153) defined in ‘fodderbeet.dai’.

27.11 Froegraes

A ‘default’ model (see 27.1, page 153) defined in ‘dk-grass.dai’.

Frgrs med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.12 Graes

A ‘Froegraes’ model (see 27.11, page 156) defined in ‘dk-grass.dai’.

Grs med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.13 Grass to grain

A ‘default’ model (see 27.1, page 153) defined in ‘grass.dai’.

27.14 Grass

A ‘Grass to grain’ model (see 27.13, page 156) defined in ‘grass.dai’.

27.15 Maize

A ‘default’ model (see 27.1, page 153) defined in ‘maize.dai’.

Roskilde 74-76 ukendt varietet

27.16 Ikuwala Maize

A ‘Maize’ model (see 27.15, page 156) defined in ‘maize.dai’.

SASA Project

27.17 Silage Maize

A ‘Maize’ model (see 27.15, page 156) defined in ‘maize.dai’.

LG11

27.18 Onion

A ‘default’ model (see 27.1, page 153) defined in ‘onion.dai’.

For AArhus County - Samsoe Project

27.19 Onion - planting of sets

A ‘Onion’ model (see 27.18, page 156) defined in ‘onion_plantingofsets.dai’.

27.20 Pea

A ‘default’ model (see 27.1, page 153) defined in ‘pea.dai’.

27.21 Green Pea

A ‘Pea’ model (see 27.20, page 157) defined in ‘pea.dai’.

27.22 Pioneer Maize

A ‘default’ model (see 27.1, page 153) defined in ‘maize.dai’.

27.23 Potato; Koege

A ‘default’ model (see 27.1, page 153) defined in ‘potato.dai’.

Kge Project

27.24 Potato; SCRI

A ‘default’ model (see 27.1, page 153) defined in ‘potato.dai’.

Potato parameterized by sha@kvl.dk – 2003. It is based on experimental data from SCRI from 1983 and 1984.

27.25 Potato; FertOrgaNic

A ‘Potato; SCRI’ model (see 27.24, page 157) 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 variety specific calibration.

The calibration was performed by Charlotte Tofteng <cto@kvl.dk> in 2006.

27.26 Potato; Agria

A ‘Potato; FertOrgaNic’ model (see 27.25, page 157) 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).

27.27 Potato; Folva

A ‘Potato; FertOrgaNic’ model (see 27.25, page 157) defined in ‘potato.dai’.

Recalibration of Agria potato performed by Tove Heidman in 2006 based on experimental results from DJF.

27.28 Potato

A ‘Potato; Folva’ model (see 27.27, page 157) defined in ‘potato.dai’.

27.29 Potato; Triada

A ‘Potato; FertOrgaNic’ model (see 27.25, page 157) defined in ‘potato.dai’.

Recalibration of Triada potato performed by Charlotte Tofteng <cto@kvl.dk> based on experimental results from Jadwisin, Poland 2003-2005.

27.30 Potato; SCRI - AArhus

A ‘default’ model (see 27.1, page 153) defined in ‘earlypotato.dai’.

Potato parameterized by sha@kvl.dk – 2003. It is based on experimental data from SCRI from 1983 and 1984.

27.31 Early potato

A ‘Potato; SCRI - AArhus’ model (see 27.30, page 158) defined in ‘earlypotato.dai’.

For AArhus County - Samsøe Project

27.32 Rug

A ‘default’ model (see 27.1, page 153) defined in ‘dk-rye.dai’.

Rug med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.33 Rye

A ‘default’ model (see 27.1, page 153) defined in ‘rye.dai’.

27.34 Ryegrass

A ‘default’ model (see 27.1, page 153) defined in ‘ryegrass.dai’.

ryegrass v.27 Henning Hgh Jensen

27.35 Silomajs

A ‘default’ model (see 27.1, page 153) 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.

27.36 Spring Barley

A ‘default’ model (see 27.1, page 153) defined in ‘sbarley.dai’.

27.37 Spring Barley; Foulum

A ‘Spring Barley’ model (see 27.36, page 158) defined in ‘sbarley.dai’.
RS-Model Projekt

27.38 Spring Rape

A ‘default’ model (see 27.1, page 153) defined in ‘srape.dai’.

27.39 Spring Wheat

A ‘default’ model (see 27.1, page 153) defined in ‘swheat.dai’.
Kge Project

27.40 Sugar Beet

A ‘default’ model (see 27.1, page 153) defined in ‘sugarbeet.dai’.
Neuenkirchen Project

27.41 Vaarbyg

A ‘default’ model (see 27.1, page 153) defined in ‘dk-sbarley.dai’.
Vrbyg med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.42 Vinterbyg

A ‘default’ model (see 27.1, page 153) defined in ‘dk-wbarley.dai’.
Vinterbyg med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.43 Vinterhvede

A ‘default’ model (see 27.1, page 153) defined in ‘dk-wwheat.dai’.
Vinterhvede med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet.

27.44 Vinterraps

A ‘default’ model (see 27.1, page 153) defined in ‘dk-wrape.dai’.
Vinterraps med rsudbytter der svarer til gennemsnittet for dansk landbrug. Skabt af Christian Thirup for Daisy standardiseringsprojektet

27.45 Wclover

A ‘default’ model (see 27.1, page 153) defined in ‘wclover.dai’.
v.30 Developed by Henning Hgh Jensen

27.46 White cabbage

A ‘default’ model (see 27.1, page 153) defined in ‘whitecabbage.dai’.
For AArhus County - Samsøe Project

27.47 White cabbage - transplanted

A ‘White cabbage’ model (see 27.46, page 160) defined in ‘whitecabbage.dai’.

27.48 Early white cabbage - transplanted

A ‘White cabbage - transplanted’ model (see 27.47, page 160) defined in ‘early-whitecabbage.dai’.

27.49 Winter Barley

A ‘default’ model (see 27.1, page 153) defined in ‘wbarley.dai’.

27.50 Winter Barley; Foulum

A ‘Winter Barley’ model (see 27.49, page 160) defined in ‘wbarley.dai’.

27.51 Winter Barley; Koge

A ‘default’ model (see 27.1, page 153) defined in ‘wbarley.dai’.
Kge Project

27.52 Winter Rape

A ‘default’ model (see 27.1, page 153) defined in ‘wrape.dai’.

27.53 Winter Wheat

A ‘default’ model (see 27.1, page 153) defined in ‘wwheat.dai’.

27.54 Winter Wheat; Eest

A ‘Winter Wheat’ model (see 27.53, page 160) defined in ‘wwheat.dai’.

27.55 Winter Wheat; Foulum

A ‘Winter Wheat’ model (see 27.53, page 160) defined in ‘wwheat.dai’.
RS-Model Project

27.56 simple

Forced growth crop model.

```
< simple (day 0 [d])
      (T_sum 0 [dg C d])
      (Root Root) ; Has default value.
      (Canopy Canopy) ; Has default value.
      (LAIvsTS LAIvsTS)
      (LAIvsDay LAIvsDay)
      (forced_LAI 0 [m2/m2])
      (height_max 80 [cm])
      (spring spring ...) ; Has default value.
      (spring_LAI 0.1 [m2/m2])
      (root_DM 2 [Mg DM/ha])
      (root_N 20 [kg N/ha])
      (root_am root_am ...) ; Has default value.
      (potential_N potential_N)
      (N_actual 0 [g N/m2])
      (N_b 10 [kg N/ha])
      (N_flowering 0.9 [<fraction>])
      ;; Shared parameters are described in chapter 27.
      (description description)
      (cite) >
```

- *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 93.13)
Submodel (has fully specified default value)
Root system.
- *Canopy*: **CanopySimple** fixed component (see section 93.15)
Submodel (has fully specified default value)
Canopy.
- *LAIvsTS*: plf [dg C d → m²/m²]
Optional parameter
LAI as a function of T_sum
- *LAIvsDay*: plf [d → m²/m²]
Optional parameter
LAI as a function of number of days since sowing.
- *forced_LAI*: number [m²/m²]
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 [m^2/m^2]
Parameter (default 0.1)
Set 'forced_LAI' to this after spring clearance 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 8) 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 [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 28

depth

Find the depth of two numbers.

28.1 PLF

Linear interpolation of depth.

`< PLF table... >`

- *table*: submodel (see section 4.1.7) 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 93.21)
Submodel (has partially specified default value)
Time.
- *value*: number [**cm**]
Parameter
Depth.

28.2 const

Constant depth.

`< const value >`

- *value*: number [**cm**]
Parameter
Constant depth.

28.3 extern

Look up depth in an scope.

`< extern (value value)
(initial_value initial_value) >`

- *value*: **number** component (see chapter 50)
Expression that evaluates to a depth.

- *initial_value*: number [**cm**]
Optional parameter
Initial depth.

28.4 file

Linear interpolation of depth read from file.

< file *file* >

- *file*: string (see section 4.1.5)
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.

Chapter 29

difrad

The 'difrad' component should calculate the diffuse radiation from meteorological data.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

29.1 DPF

Diffuse radiation calculated using the model of De Pury and Farquhar, 1997. See also [de Pury and Farquhar, 1997]

```
< DPF (cite cite ...) ; Has default value.  
      (a 0.84 [])  
      (fa 0.5 [<fraction>])  
      ;; Shared parameters are described in chapter 29.  
      (description description) >
```

- *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.

29.2 weather

Diffuse radiation using weather data.

Chapter 30

domsorp

Sorption and desorption of DOM to SOM.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

30.1 default

Transformation between two soil chemicals.

```
< default (transform transform)  
          (dom_pool dom_pool)  
          (som_pool som_pool)  
          ;; Shared parameters are described in chapter 30.  
          (description description)  
          (cite) >
```

- *transform*: **transform** component (see chapter 77)
Transformation 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/cm³/h] soil cells
Carbon converted from DOM to SOM (may be negative).
- *S_N*: number [g N/cm³/h] soil cells
Carbon converted from DOM to SOM (may be negative).

Chapter 31

drain

Lateral transport of water.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

31.1 lateral

Pipe drainage. See also [Hooghoudt, 1940]

```
< lateral (x x)  
          (L 1800 [cm])  
          (rad 3.5 [cm])  
          (height height)  
          (cite cite ...) ; Has default value.  
          (pipe_position -110 [cm])  
          (K_to_pipes K_to_pipes)  
          (eq_depth eq_depth) ; Default MolenWesseling value.  
          (pipe_outlet pipe_outlet)  
          (pipe_level pipe_level)  
          ;; Shared parameters are described in chapter 31.  
          (description description) >
```

- *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 32)
Component (default 'MolenWesseling')
Model for calculating equivalent depth for drains.
- *pipe_outlet*: **depth** component (see chapter 28)
Optional component
Water table in drain pipe outlet.

By default this will be identical to 'pipe_position', meaning free flow of water through drains. Specifying a lower water level will not affect the simulation. Specifying a higher water level is functionally equivalent to temporarily increasing the 'pipe_position', lowering the ability of the drain system to drain the soil.

Currently, there is no possibility of water flowing from the pipe outlet to soil.
- *pipe_level*: number [**cm**]
Optional state variable
Current effective pipe position (a negative number).

Log Variables

- *S*: number [**cm³/cm³/h**] soil cells
Pipe drainage.
- *DrainFlow*: number [**cm/h**]
Drain flow to pipes.
- *EqDrnFlow*: number [**cm/h**]
Equilibrium drain flow to pipes.

31.2 none

No lateral transport.

Used by column default Drain (see 24.1, page 138) .

Chapter 32

draineqd

Find the equilibrium drain depth for the Hooghoudt drainage model.

```
< component (description description)  
            (cite cite ...) ; Has default value. >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (has default value with length 1)

```
(cite hooghoudt)
```

Parameter description:

BibTeX keys that would be relevant for this model or parameterization.

32.1 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 31.1, page 169) .

32.2 Moody

Equivalent depth calculated with model by Moody. See also [Moody, 1966]

32.3 Wesseling

Equivalent depth calculated with model by Wesseling. See also [Wesseling, 1973]

32.4 none

No modifications. See also [Hooghoudt, 1940]

Chapter 33

element

An element of a compound.

33.1 atom

An atom.

`< atom (mass mass) >`

- *mass*: number [g/mol]
Parameter
Atomic mass.

Chapter 34

equilibrium

Find equilibrium between two soil chemicals.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

34.1 Langmuir

$A = (\text{my_max } B) / (K + B)$

```
< Langmuir (K K)
          (my_max my_max)
          ;; Shared parameters are described in chapter 34.
          (description description)
          (cite) >
```

- *K*: **number** component (see chapter 50)
Half saturation constant [g/cm³].
- *my_max*: **number** component (see chapter 50)
Max equilibrium capacity [g/cm³].

34.2 goal_A

Attempt to maintain A at a fixed level.

```
< goal_A (goal_A goal_A)
        (A_solute A_solute)
        (min_B min_B)
        (B_solute B_solute)
        (debug_cell -1)
        ;; Shared parameters are described in chapter 34.
        (description description)
        (cite) >
```

- *goal_A*: **number** component (see chapter 50)
The desired level of A [g/cm³].
- *A_solute*: boolean (see section 4.1.2)
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 50)
Do not convert B to A if B is smaller than this [g/cm³].
- *B_solute*: boolean (see section 4.1.2)
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.

34.3 linear

```

A = K B
< linear (K K)
    ;; Shared parameters are described in chapter 34.
    (description description)
    (cite) >

```

- *K*: **number** component (see chapter 50)
The ratio A/B at equilibrium [].

Chapter 35

exchange

A named value to exchange with external models.

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

- *name*: string (see section 4.1.5)
Parameter
Name of value to exchange.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

35.1 name

Exchange a string value.

```
< name (value value)
      ;; Shared parameters are described in chapter 35.
      (name name)
      (description description)
      (cite) >
```

- *value*: string (see section 4.1.5)
Parameter
Current value to exchange.

35.2 number

Exchange a numeric value.

```
< number (value value)
         (dimension dimension)
         ;; Shared parameters are described in chapter 35.
         (name name)
         (description description)
         (cite) >
```

- *value*: number (dimension not specified)
Optional state variable
Current value to exchange.
- *dimension*: string (see section 4.1.5)
Parameter
Dimension of value to exchange.

Chapter 36

format

Text formatting component.

36.1 LaTeX

Format text as LaTeX.

Used by program document format (see 57.9, page 307) , and program docmodel format (see 57.8, page 307) .

Chapter 37

gnuplot

Plot a graph with gnuplot.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

37.1 common

Common parameters.

```
< common (where screen)  
        (title title)  
        (extra)  
        (device device)  
        (canvas canvas)  
        (size size)  
        (legend auto)  
        ;; Shared parameters are described in chapter 37.  
        (description description)  
        (cite) >
```

- *where*: string (see section 4.1.5)
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. *.emf: Enhanced Metafile.

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 4.1.5)
Optional parameter
Title of the plot, if any. Set it to an empty string to disable.

- *extra*: string (see section 4.1.5) 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 4.1.5)
Optional parameter
Output device. By default, this is derived from the file extension.
- *canvas*: string (see section 4.1.5)
Optional parameter
Canvas size. By default, this depend on the device.
- *size*: submodel (see section 4.1.7)
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 4.1.5)
Optional parameter (default 'auto')
Placement of legend. This can be one of the four corners, named by compass 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 connecting two datapoints, one of them outside the graph, may cross the legend.

37.2 profile

A 'common' model (see 37.1, page 181) build into Daisy.

Plot 2D soil profile.

```
< profile (column column)
    ;; Shared parameters are described in section 37.1.
    (description description)
    (cite)
    (where screen)
    (title title)
    (extra)
    (device device)
    (canvas canvas)
    (size size)
    (legend auto) >
```

- *column*: **column** component (see chapter 24)
Column whose soil profile to plot.

37.3 soil

A ‘common’ model (see 37.1, page 181) build into Daisy.

Generate a 2D gnuplot graph with soil content.

```
< soil (file file)
      (max max)
      (min min)
      (dimension dimension)
      (bottom bottom)
      (type block)
      (top 0 [cm])
      (left 0 [cm])
      (right right)
      (missing missing ...) ; Has default value.
      (filter)
      (original original ...)
      (dim_line dim_line)
      (when when) ; Has partial value.
      (samples 25)
      ;; Shared parameters are described in section 37.1.
      (description description)
      (cite)
      (where screen)
      (title title)
      (extra)
      (device device)
      (canvas canvas)
      (size size)
      (legend auto) >
```

- *file*: string (see section 4.1.5)
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 4.1.5)
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 4.1.5)
State variable (default ‘block’)
Plot type. Valid options are ‘block’ and ‘contour’.

- *top*: number [**cm**]
Parameter (default 0)
Highest 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 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:
List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

< tag allowed... >

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 93.21)
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.

37.4 time

A ‘common’ model (see 37.1, page 181) build into Daisy.

Generate a gnuplot graph with times series.

```
< time (source source ...)
      (begin begin)
      (end end)
      (ymin ymin)
      (ymax ymax)
      (y2min y2min)
      (y2max y2max)
      ;; Shared parameters are described in section 37.1.
      (description description)
      (cite)
      (where screen)
      (title title)
      (extra)
      (device device)
      (canvas canvas)
      (size size)
      (legend auto) >
```

- *source*: **source** component (see chapter 70) sequence
Time series to plot.
- *begin*: **Time** fixed component (see section 93.21)
Optional submodel
First date at x-axis.
- *end*: **Time** fixed component (see section 93.21)
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.

37.5 vector

A ‘common’ model (see 37.1, page 181) build into Daisy.

Generate a 2D gnuplot vector field.

```

< vector (file file)
        (dimension dimension)
        (factor factor)
        (bottom bottom)
        (top 0 [cm])
        (left 0 [cm])
        (right right)
        (missing missing ...) ; Has default value.
        (filter)
        (original original ...)
        (dim_line dim_line)
        (begin begin)
        (end end)
        (when when)
        ;; Shared parameters are described in section 37.1.
        (description description)
        (cite)
        (where screen)
        (title title)
        (extra)
        (device device)
        (canvas canvas)
        (size size)
        (legend auto) >

```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *dimension*: string (see section 4.1.5)
Optional parameter
Dimension for data. By default, use dimension from file.
- *factor*: number (dimension not specified)
Parameter
Multiply vector values with this factor.
- *bottom*: number [**cm**]
Optional parameter
Deepest z value in plot. By default, derive value from data file.
- *top*: number [**cm**]
Parameter (default 0)
Highest 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 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

`< tag allowed... >`

– *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.

– *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 93.21)
Optional submodel
Ignore values before this time.
- *end*: **Time** fixed component (see section 93.21)
Optional submodel
Ignore values after this time.
- *when*: **Time** fixed component (see section 93.21)
Optional submodel
Use value closest to this time.

37.6 xy

A 'common' model (see 37.1, page 181) build into Daisy.

Generate a gnuplot graph with up to two x-axes.

```

< xy (source source ...)
      (ymin ymin)
      (ymax ymax)
      (y2min y2min)
      (y2max y2max)
      (xmin xmin)
      (xmax xmax)
      (x2min x2min)
      (x2max x2max)
      ;; Shared parameters are described in section 37.1.
      (description description)
      (cite)
      (where screen)
      (title title)
      (extra)
      (device device)
      (canvas canvas)
      (size size)
      (legend auto) >

```

- *source*: **xy**source component (see chapter 91) 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.

37.7 multi

Generate multiple graphs for the gnuplot command file.

```
< multi  (before)
          (after)
          (graph graph ...)
          ;; Shared parameters are described in chapter 37.
          (description description)
          (cite) >
```

- *before*: string (see section 4.1.5) 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 4.1.5) 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 37) sequence
Graphs to plot.

Chapter 38

groundwater

The 'groundwater' component is responsible for specifying the groundwater table at each timestep.

38.1 common

All groundwater models can log height.

```
< common >
```

Log Variables

- *height*: number [**cm**]
Groundwater level. Positive numbers indicate free drainage.

38.2 aquitard

A 'common' model (see 38.1, page 191) build into Daisy.

Aquitard groundwater, free drainage.

```
< aquitard (K_aquitard 0.001 [cm/h])  
          (Z_aquitard 200 [cm])  
          (h_aquifer h_aquifer)  
          (pressure_table pressure_table) >
```

- *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 28)
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'.

38.3 deep

A 'common' model (see 38.1, page 191) build into Daisy.

Deep groundwater, free drainage.

Used by column Askov Groundwater (see 24.2, page 140) , and column Jyndevad Groundwater (see 24.4, page 140) .

38.4 extern

A 'common' model (see 38.1, page 191) build into Daisy.

Look up groundwater table in an scope.

```
< extern (table table)
        (initial_table initial_table) >
```

- *table*: **number** component (see chapter 50)
Expression that evaluates to groundwater table in.
- *initial_table*: number [**cm**]
Optional parameter
Groundwater level for initialization of soil water.

38.5 file

A 'common' model (see 38.1, page 191) build into Daisy.

Read groundwater table from a file.

```
< file file
      (offset 0 [cm]) >
```

- *file*: string (see section 4.1.5)
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.

38.6 fixed

A 'common' model (see 38.1, page 191) build into Daisy.

Fixed high groundwater level.

```
< fixed table >
```

- *table*: number [**cm**]
Parameter
Groundwater level (negative number below surface).

38.7 flux

A ‘common’ model (see 38.1, page 191) build into Daisy.

Flux groundwater, free drainage.

```
< flux flux >
```

- *flux*: number [cm/h]
Parameter
Constant flux to groundwater.

38.8 lysimeter

A ‘common’ model (see 38.1, page 191) build into Daisy.

Lysimeter bottom.

38.9 source

A ‘common’ model (see 38.1, page 191) build into Daisy.

Read groundwater table from a source.

```
< source source
      (offset 0 [cm]) >
```

- *source*: **source** component (see chapter 70)
Groundwater table time series.
- *offset*: number [cm]
Parameter (default 0)
Add this to depth from source.

38.10 static

A ‘common’ model (see 38.1, page 191) build into Daisy.

Static groundwater level. Provided for backward compatibility, use ‘deep’ or ‘fixed’ instead.

```
< static (table 1 [cm]) >
```

- *table*: number [cm]
Parameter (default 1)
Groundwater level. Positive numbers indicate free drainage.

38.11 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 groundwater level to sink into the aquitard. The model cannot handle groundwater levels below the last cell, or above the soil surface.

```

< pipe (x x)
      (L 1800 [cm])
      (height height)
      (K_aquitard 0.001 [cm/h])
      (Z_aquitard 200 [cm])
      (h_aquifer h_aquifer)
      (pressure_table pressure_table)
      (pipe_position -110 [cm])
      (K_to_pipes K_to_pipes) >

```

- *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 28)
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 aquitard 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 [$\text{cm}^3/\text{cm}^3/\text{h}$] 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 39

heatrect

Heat transport in rectangular grid.

39.1 Mollerup

Finite volume solution to heat transfer by Mikkel Mollerup.

Used by movement rectangle heat (see 47.2, page 257) .

```
< Mollerup (solver solver) ; Default cxsparse value.  
      (debug 0) >
```

- *solver*: **solver** component (see chapter 69)
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.

39.2 linear

Linear temperature interpolation between top and bottom.

39.3 none

No heat transport.

Chapter 40

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 93.9, page 491) , and SoilRegion @ horizon (see 93.10, page 492) .

```
< 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)
                  (r_pore_min 0.1 [um]) >
```

- *hydraulic*: **hydraulic** component (see chapter 41)
Component (default ‘hypres’)
The hydraulic propeties of the soil.
- *tortuosity*: **tortuosity** component (see chapter 76)
Component (default ‘M_Q’)
The soil tortuosity.
- *HorHeat*: **HorHeat** fixed component (see section 93.22)
Submodel (has fully specified default value)
Heat capacity and conductivity.
- *description*: string (see section 4.1.5)
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 4.1.5) 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/cm³]
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, typically slow, fast and inert. 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 49)
Component (default 'soil')
The soil nitrification process.
- *secondary_domain*: **secondary** component (see chapter 65)
Component (default 'none')
Secondary matrix domain for solute movement.
- *attributes*: submodel (see section 4.1.7) 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 4.1.5)
Parameter
Name of attribute.

- *value*: number [**<user>**]
Parameter
Value of attribute.
- *r_pore_min*: number [**um**]
Parameter (default 0.1)
Smallest pores in the soil.

40.1 BSI3

A horizon using BSI3 texture classification.

```
< BSI3  (clay clay)
        (silt silt)
        (sand sand)
        (humus humus)
        (normalize false)
        ;; Shared parameters are described in chapter 40.
        (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)
        (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)
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.

40.2 DIN3

A 'BSI3' model (see 40.1, page 201) build into Daisy.
'DIN3' is another name for 'BSI3'

40.3 MIT3

A 'BSI3' model (see 40.1, page 201) build into Daisy.
'MIT3' is another name for 'BSI3'

40.4 BSI7

A horizon using BSI7 texture classification.

```
< BSI7 (clay clay)
    (fine_sand fine_sand)
    (medium_sand medium_sand)
    (coarse_sand coarse_sand)
    (fine_silt fine_silt)
    (medium_silt medium_silt)
    (coarse_silt coarse_silt)
    (humus humus)
    (normalize false)
    ;; Shared parameters are described in chapter 40.
    (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)
    (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)
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.

40.5 MIT7

A 'BSI7' model (see 40.4, page 202) build into Daisy.

'MIT7' is another name for 'BSI7'

40.6 DIN5

A horizon using DIN5 texture classification.

```
< DIN5  (clay clay)
        (silt silt)
        (fine_sand fine_sand)
        (medium_sand medium_sand)
        (coarse_sand coarse_sand)
        (humus humus)
        (normalize false)
        ;; Shared parameters are described in chapter 40.
        (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)
        (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)
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.

40.7 ISSS3

A horizon using ISSS3 texture classification.

- ```
< ISSS3 (clay clay)
 (silt silt)
 (sand sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```
- *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 4.1.2)  
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.

## 40.8 ISSS4

A horizon using ISSS4 texture classification.

```
< ISSS4 (clay clay)
 (silt silt)
 (fine_sand fine_sand)
 (coarse_sand coarse_sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)  
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.

## 40.9 USDA3

A horizon using USDA3 texture classification.

```
< USDA3 (clay clay)
 (silt silt)
 (sand sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)  
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.

## 40.10 FAO3

A ‘USDA3’ model (see 40.9, page 206) build into Daisy.

‘FAO3’ is another name for ‘USDA3’

## 40.11 USDA7

A horizon using USDA7 texture classification.

```
< USDA7 (clay clay)
 (silt silt)
 (very_fine_sand very_fine_sand)
 (fine_sand fine_sand)
 (medium_sand medium_sand)
 (coarse_sand coarse_sand)
 (very_coarse_sand very_coarse_sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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\_coarse\_sand*: number [**<fraction>**]  
Parameter  
Mineral particles between 1000 [um] and 2000 [um].
- *humus*: number [**<fraction>**]  
Parameter  
Humus content of soil.
- *normalize*: boolean (see section 4.1.2)  
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.

## 40.12 FAO7

A 'USDA7' model (see 40.11, page 207) build into Daisy.  
'FAO7' is another name for 'USDA7'

## 40.13 USPRA3

A horizon using USPRA3 texture classification.

```
< USPRA3 (clay clay)
 (silt silt)
 (sand sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)  
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.

## 40.14 USPRA4

A horizon using USPRA4 texture classification.

```
< USPRA4 (clay clay)
 (silt silt)
 (fine_sand fine_sand)
 (coarse_sand coarse_sand)
 (humus humus)
 (normalize false)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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 4.1.2)  
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.

## 40.15 default

USDA/FAO texture classification.

The soil constituents are automatically normalized.

OBSOLETE: Use the USDA or FAO model instead.

```
< default (clay clay)
 (silt silt)
 (sand sand)
 (fine_sand fine_sand)
 (coarse_sand coarse_sand)
 (humus humus)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *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.

## 40.16 Ap\_JB1

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.17 Ap\_JB2

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.18 Ap\_JB3

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.19 Ap\_JB4

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.20 Ap\_JB5

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.21 Ap\_JB6

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.22 Ap\_JB7

A 'default' model (see 40.15, page 210) defined in 'dk-horizon.dai'.

## 40.23 Askov Ap

A 'default' model (see 40.15, page 210) 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 Planteavlfsforsg.

## 40.24 Askov B

A ‘default’ model (see 40.15, page 210) 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.

## 40.25 Askov C

A ‘default’ model (see 40.15, page 210) 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.

## 40.26 B\_JB1

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.27 B\_JB2

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.28 B\_JB3

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.29 B\_JB4

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.30 B\_JB5

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.31 B\_JB6

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.32 B\_JB7

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.33 C\_JB1

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

## 40.34 C\_JB2

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.35 C\_JB3**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.36 C\_JB4**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.37 C\_JB5**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.38 C\_JB6**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.39 C\_JB7**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-horizon.dai’.

**40.40 Foulum AP**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-soil.dai’.

Provided by DJF as part if the FertOrgaNic project.

**40.41 Foulum B**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-soil.dai’.

Provided by DJF as part if the FertOrgaNic project.

**40.42 Foulum C**

A ‘default’ model (see 40.15, page 210) defined in ‘dk-soil.dai’.

Provided by DJF as part if the FertOrgaNic project.

**40.43 Jyndevad Ap**

A ‘default’ model (see 40.15, page 210) 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.

**40.44 Jyndevad C**

A ‘default’ model (see 40.15, page 210) 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.

## 40.45 aquitard

A ‘default’ model (see 40.15, page 210) build into Daisy.  
Texture for implicit aquitard horizon.

## 40.46 numeric

A horizon using explicit texture classification.

```
< numeric (humus humus)
 (fractions fractions ...)
 (normalize false)
 (limits limits ...)
 ;; Shared parameters are described in chapter 40.
 (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)
 (r_pore_min 0.1 [um]) >
```

- *humus*: number [<fraction>]  
Parameter  
Humus content of soil.
- *fractions*: number [<fraction>] sequence  
Parameter  
Fraction of particles between the corresponding numerical limits.
- *normalize*: boolean (see section 4.1.2)  
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 41

## hydraulic

This component is responsible for specifying the soils hydraulic properties.

Used by horizon component hydraulic (see 40, page 199) .

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

- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.

### 41.1 B\_BaC

Brooks and Corey retention curve model with Burdine theory.

Used by horizon Jynde vad Ap hydraulic (see 40.43, page 213) , and horizon Jynde vad C hydraulic (see 40.44, page 213) .

```
< B_BaC (lambda lambda)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (h_b h_b)
 (K_at_h K_at_h)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- *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 4.1.7)  
Optional submodel  
Water conductivity at specified pressure.

< *h K* >

- *h*: number [**cm**]  
Parameter  
Soil water pressure.
- *K*: number [**cm/h**]  
Parameter  
Water conductivity.

## 41.2 B\_BaC\_Bimodal

Brooks and Corey retention curve model with Burdine theory. Bimodal hydraulic conductivity curve.

```
< B_BaC_Bimodal (lambda lambda)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (h_b h_b)
 (Theta_b Theta_b)
 (K_b K_b)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- *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'.

### 41.3 B\_C

Campbell retention curve model with Burdine theory. See also [Campbell, 1974, Burdine, 1953]

```
< B_C (cite cite ...) ; Has default value.
 (Theta_sat Theta_sat)
 (K_sat K_sat)
 (h_b h_b)
 (K_at_h K_at_h)
 (b b)
 (1 2 [])
 ;; Shared parameters are described in chapter 41.
 (description description) >
```

- *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 4.1.7)  
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.
- *l*: number (dimensionless)  
Parameter (default 2)  
Burdine form parameter.

## 41.4 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 conductivity curve is fully specified by a single point. See also [Campbell, 1974, Burdine, 1953]

```
< B_C_inverse (cite cite ...) ; Has default value.
 (Theta_sat Theta_sat)
 (K_sat K_sat)
 (K_at_h K_at_h)
 (Theta_fc Theta_fc)
 (Theta_wp Theta_wp)
 ;; Shared parameters are described in chapter 41.
 (description description) >
```

- *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 4.1.7)  
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]

## 41.5 B\_vG

van Genuchten retention curve model with Burdine theory.

```

< B_vG (alpha alpha)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (K_at_h K_at_h)
 (1 2 [])
 (n n)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >

```

- *alpha*: number [**cm**<sup>-1</sup>]  
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 4.1.7)  
Optional submodel  
Water conductivity at specified pressure.

< **h** *K* >

- *h*: number [**cm**]  
Parameter  
Soil water pressure.
- *K*: number [**cm/h**]  
Parameter  
Water conductivity.
- *l*: number (dimensionless)  
Parameter (default 2)  
tortuosity parameter.
- *n*: number (dimensionless)  
Parameter  
van Genuchten n.

## 41.6 Cosby\_et\_al

Modified Campbell retention curve model with Burdine theory. Parameters estimated from soil texture as specified by Cosby et al.

## 41.7 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]

```
< MACRO (alpha alpha)
 (cite cite ...) ; Has default value.
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (h_b h_b)
 (Theta_b Theta_b)
 (K_b K_b)
 (l 0.5 [])
 (n n)
 (n_ma n_ma)
 (enable_K_macro true)
 (enable_Theta_macro true)
 ;; Shared parameters are described in chapter 41.
 (description description) >
```

- *alpha*: number [ $\text{cm}^{-1}$ ]  
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 [ $\text{cm/h}$ ]  
Optional parameter  
Water conductivity of saturated soil.
- *h\_b*: number [ $\text{cm}$ ]  
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 [ $\text{cm/h}$ ]  
Optional parameter  
Water conductivity at boundary point.
- *l*: number (dimensionless)  
Parameter (default 0.5)  
tortuosity parameter.
- *n*: number (dimensionless)  
Parameter  
van Genuchten n.

- *n\_ma*: number (dimensionless)  
Parameter  
Macropore size distribution factor.
- *enable\_K\_macro*: boolean (see section 4.1.2)  
Parameter (default true)  
Include contribution from macropores in conductivity curve.
- *enable\_Theta\_macro*: boolean (see section 4.1.2)  
Parameter (default true)  
Include contribution from macropores in retention curve.

## 41.8 M\_BaC

Brooks and Corey retention curve model with Mualem theory.

```
< M_BaC (lambda lambda)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (h_b h_b)
 (K_at_h K_at_h)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- *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 4.1.7)  
Optional submodel  
Water conductivity at specified pressure.  

```
< h K >
```

  - *h*: number [**cm**]  
Parameter  
Soil water pressure.
  - *K*: number [**cm/h**]  
Parameter  
Water conductivity.

## 41.9 M\_BaC\_Bimodal

Brooks and Corey retention curve model with Mualem theory. Bimodal hydraulic conductivity curve.

```
< M_BaC_Bimodal (lambda lambda)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (h_b h_b)
 (Theta_b Theta_b)
 (K_b K_b)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- *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'.

## 41.10 M\_C

Campbell retention curve model with Mualem theory. See also [Campbell, 1974, Mualem, 1976]

```
< M_C (cite cite ...) ; Has default value.
 (Theta_sat Theta_sat)
 (K_sat K_sat)
 (h_b h_b)
 (K_at_h K_at_h)
 (b b)
 (l 0.5 [])
 ;; Shared parameters are described in chapter 41.
 (description description) >
```

- *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 4.1.7)  
Optional submodel  
Water conductivity at specified pressure.  
  

$\begin{matrix} < & h & K & > \end{matrix}$

  - *h*: number [**cm**]  
Parameter  
Soil water pressure.
  - *K*: number [**cm/h**]  
Parameter  
Water conductivity.
- *b*: number (dimensionless)  
Parameter  
Campbell parameter.
- *l*: number (dimensionless)  
Parameter (default 0.5)  
Mualem form parameter.

## 41.11 M\_vG

van Genuchten retention curve model with Mualem theory.

Used by horizon Ap\_JB1 hydraulic (see 40.16, page 211) , horizon Ap\_JB2 hydraulic (see 40.17, page 211) , horizon Ap\_JB3 hydraulic (see 40.18, page 211) , horizon Ap\_JB4 hydraulic (see 40.19, page 211) , horizon Ap\_JB5 hydraulic (see 40.20, page 211) , horizon Ap\_JB6 hydraulic (see 40.21, page 211) , horizon Ap\_JB7 hydraulic (see 40.22, page 211) , horizon B\_JB1 hydraulic (see 40.26, page 212) , horizon B\_JB2 hydraulic (see 40.27, page 212) , horizon B\_JB3 hydraulic (see 40.28, page 212) , horizon B\_JB4 hydraulic (see 40.29, page 212) , horizon B\_JB5 hydraulic (see 40.30, page 212) , horizon B\_JB6 hydraulic (see 40.31, page 212) , horizon B\_JB7 hydraulic (see 40.32, page 212) , horizon C\_JB1 hydraulic (see 40.33, page 212) , horizon C\_JB2 hydraulic (see 40.34, page 212) , horizon C\_JB3 hydraulic (see 40.35, page 213) , horizon C\_JB4 hydraulic (see 40.36, page 213) , horizon C\_JB5 hydraulic (see 40.37, page 213) , horizon C\_JB6 hydraulic (see 40.38, page 213) , horizon C\_JB7 hydraulic (see 40.39, page 213) , horizon Askov Ap hydraulic (see 40.23, page 211) , horizon Askov B hydraulic (see 40.24, page 212) , horizon Askov C hydraulic (see 40.25, page 212) , horizon Foulum AP hydraulic (see 40.40, page 213) , horizon Foulum B hydraulic (see 40.41, page 213) , and horizon Foulum C hydraulic (see 40.42, page 213) .

```

< M_vG (alpha alpha)
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (K_at_h K_at_h)
 (1 0.5 [])
 (n n)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >

```

- *alpha*: number [**cm**<sup>-1</sup>]  
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 4.1.7)  
Optional submodel  
Water conductivity at specified pressure.

< **h** *K* >

- *h*: number [**cm**]  
Parameter  
Soil water pressure.
- *K*: number [**cm/h**]  
Parameter  
Water conductivity.
- *l*: number (dimensionless)  
Parameter (default 0.5)  
tortuosity parameter.
- *n*: number (dimensionless)  
Parameter  
van Genuchten n.

## 41.12 M\_vG\_compact

van Genuchten retention curve model with Mualem theory and compaction.



```

< M_vG_compact (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (ref_alpha ref_alpha)
 (ref_n ref_n)
 (ref_K_sat ref_K_sat)
 (mod_alpha mod_alpha)
 (mod_n mod_n)
 (mod_K_sat mod_K_sat)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >

```

- *Theta\_sat*: number [<fraction>]  
State variable  
Saturation point.
- *Theta\_res*: number [<fraction>]  
Parameter (default 0)  
Soil residual water.
- *ref\_alpha*: number [ $\text{cm}^{-1}$ ]  
Parameter  
Reference van Genuchten alpha.
- *ref\_n*: number (dimensionless)  
Parameter  
Reference van Genuchten n.
- *ref\_K\_sat*: number [ $\text{cm/h}$ ]  
Parameter  
Reference water conductivity of saturated soil.
- *mod\_alpha*: plf [<fraction>  $\rightarrow$  <none>]  
Parameter  
Porosity modifier for van Genuchten alpha.
- *mod\_n*: plf [<fraction>  $\rightarrow$  <none>]  
Parameter  
Porosity modifier for van Genuchten n.
- *mod\_K\_sat*: plf [<fraction>  $\rightarrow$  <none>]  
Parameter  
Porosity modifier for water conductivity of saturated soil.

### 41.13 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 macropores.

$p_m = (1/(-h \cdot X + 1))^f$ ;  $h > h_m$   $p_m = (1/(-h_m \cdot X + 1))^f$ ;  $h \leq h_m$   $X = 1 \text{ cm}^{-1}$  See also [Børgesen et al., 2006]

```

< M_vGp (alpha alpha)
 (cite cite ...) ; Has default value.
 (Theta_sat Theta_sat)
 (Theta_res 0 [<fraction>])
 (K_sat K_sat)
 (l 0.5 [])
 (n n)
 (h_m h_m)
 (f f)
 ;; Shared parameters are described in chapter 41.
 (description description) >

```

- *alpha*: number [ $\text{cm}^{-1}$ ]  
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 [ $\text{cm}/\text{h}$ ]  
Optional parameter  
Water conductivity of saturated soil.
- *l*: number (dimensionless)  
Parameter (default 0.5)  
tortuosity parameter.
- *n*: number (dimensionless)  
Parameter  
van Genuchten n.
- *h\_m*: number [ $\text{cm}$ ]  
Parameter  
Pressure point of chance between matrix and macropores.
- *f*: number (dimensionless)  
Parameter  
Macropores conductivity curve shape parameter.

## 41.14 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 40, page 199) .

```

< hypres (cite cite ...) ; Has default value.
 (K_sat K_sat)
 (K_at_h K_at_h)
 (topsoil topsoil)
 ;; Shared parameters are described in chapter 41.
 (description description) >

```

- *K\_sat*: number [**cm/h**]  
Optional parameter  
Water conductivity of saturated soil.
- *K\_at\_h*: submodel (see section 4.1.7)  
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 4.1.2)  
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 horizon in the soil profile.

## 41.15 mod\_C

Modified Campbell retention curve model with Burdine theory.

```
< mod_C (Theta_sat Theta_sat)
 (K_sat K_sat)
 (h_b h_b)
 (K_at_h K_at_h)
 (b b)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- *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 4.1.7)  
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.

## 41.16 yolo

Yolo soil. Haverkamp et.al., 1977.

```
< yolo (M_intervals 500)
 ;; Shared parameters are described in chapter 41.
 (description description)
 (cite) >
```

- $M\_intervals$ : integer  
Parameter (default 500)  
Number of intervals for numeric integration of  $K$ .

# Chapter 42

## integer

Generic representation of integers.

### 42.1 \*

Use the product of its operands.

< "\*" *operands...* >

- *operands*: **integer** component (see chapter 42) sequence  
The operands for this function.

### 42.2 +

Use the sum of its operands.

< "+" *operands...* >

- *operands*: **integer** component (see chapter 42) sequence  
The operands for this function.

### 42.3 -

Negate integer or subtract integers. With one operand, negates it. With more than one operand, subtracts all but the first from the first.

< "-" *operands...* >

- *operands*: **integer** component (see chapter 42) sequence  
The operands for this function.

### 42.4 cond

Return the value of the first clause whose condition is true.

< cond *clauses...* >

- *clauses*: **IntegerCondClause** fixed component (see section 93.31) sequence  
List of clauses to match for.

## 42.5 **const**

Always give the specified value.

`< const value >`

- *value*: integer  
Parameter  
Fixed value for this integer.

## 42.6 **div**

Divide the first operand by the rest.

`< div operands... >`

- *operands*: **integer** component (see chapter 42) array of length 2  
The operands for this function.

## 42.7 **max**

Use the largest value of its operands.

`< max operands... >`

- *operands*: **integer** component (see chapter 42) sequence  
The operands for this function.

## 42.8 **min**

Use the smallest value of its operands.

`< min operands... >`

- *operands*: **integer** component (see chapter 42) sequence  
The operands for this function.

## 42.9 **mod**

Modulo the first operand by the rest.

`< mod operands... >`

- *operands*: **integer** component (see chapter 42) array of length 2  
The operands for this function.

## 42.10 **sqr**

Take the square of its argument.

`< sqr operand >`

- *operand*: **integer** component (see chapter 42)  
Operand for this function.

# Chapter 43

## litter

Litter, surface residuals, or mulch below canopy.

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

- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.

### 43.1 none

The effect of surface residuals is ignored by the model.

Used by column default Litter (see 24.1, page 138) .

### 43.2 permanent

A permanent litter layer cover the ground, as for example in a forest.

```
< permanent (albedo albedo)
 (interception_capacity interception_capacity)
 (vapor_flux_factor 1 [<fraction>])
 ;; Shared parameters are described in chapter 43.
 (description description)
 (cite) >
```

- *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.

### 43.3 residue

A dynamic litter layer based on applied fertilizer and crop residuals.

A 'mulch area index' is calculated from the surface organic materials, and from that a mulch cover is calculated based on Beer's law similarly to how the crop cover is calculated from the leaf area index. See also [Scopel et al., 2004]

```
< residue (cite cite ...) ; Has default value.
 (albedo albedo)
 (vapor_flux_factor 0 [<fraction>])
 (water_capacity water_capacity)
 (specific_AI specific_AI)
 (extinction_coefficient extinction_coefficient)
 ;; Shared parameters are described in chapter 43.
 (description description) >
```

- *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 [m<sup>2</sup>/kg DM]  
Parameter  
Area covered per litter mass.
- *extinction\_coefficient*: number (dimensionless)  
Parameter  
Beer's law extinction coefficient for litter.

### 43.4 Maize

A 'residue' model (see 43.3, page 232) build into Daisy.

Maize crop residues in La Tinaja. See also [Scopel et al., 1998]

### 43.5 Millet

A 'residue' model (see 43.3, page 232) build into Daisy.

Millet crop residues in Planaltina. See also [Macena et al., 2003]



# Chapter 44

## 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.

### 44.1 checkpoint

Create a checkpoint of the entire simulation state, suitable for later hot start.

```
< checkpoint (description description)
 (cite)
 (where checkpoint)
 (when when) ; Default finished value. >
```

- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.
- *where*: string (see section 4.1.5)  
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 26)  
Component (default 'finished')  
Make a checkpoint every time this condition is true.

### 44.2 harvest

Create a log of all harvests.

```
< harvest (where "harvest.dlf")
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (print_N true)
 (print_C false) >
```

- *where*: string (see section 4.1.5)  
Parameter (default 'harvest.dlf')  
Name of the log file to create.
- *print\_header*: string (see section 4.1.5)  
Parameter (default 'true')  
If this is set to 'false', no header is printed. If this is set to 'true', a full header is printed. If this is set to 'fixed', a small fixed size header is printed.
- *print\_tags*: boolean (see section 4.1.2)  
Parameter (default true)  
Print a tag line in the file.
- *print\_dimension*: boolean (see section 4.1.2)  
Parameter (default true)  
Print a line with units after the tag line.
- *print\_N*: boolean (see section 4.1.2)  
Parameter (default true)  
Print nitrogen content of harvest.
- *print\_C*: boolean (see section 4.1.2)  
Parameter (default false)  
Print carbon content of harvest.

### 44.3 select

Select variables to log.

```
< select (summary)
 (volume volume) ; Default box value.
 (description description)
 (cite)
 (from from)
 (to to)
 (when when)
 (parameter_names)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial) >
```

- *summary*: **summary** component (see chapter 73) sequence  
Component (default: an empty sequence)  
Summaries for this log file.
- *volume*: **volume** component (see chapter 88)  
Component (default 'box')  
Soil volume to log.
- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.

- *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 26)  
Add entries to the log file when this condition is true.
- *parameter\_names*: string (see section 4.1.5) 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)'.
- *entries*: **select** component (see chapter 67) sequence  
What to log in each column.
- *time\_columns*: boolean (see section 4.1.2)  
Optional parameter  
If 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.
- *print\_initial*: boolean (see section 4.1.2)  
Optional parameter  
Print a line with initial values when logging starts. By default, an initial line will be printed if any entry has 'handle' set to 'current'.

## 44.4 extern

A 'select' model (see 44.3, page 234) build into Daisy.

Log simulation state for extern use.

```
< extern (numbers numbers ...)
 (where where)
 (print_initial false)
 ;; Shared parameters are described in section 44.3.
 (summary)
 (volume volume) ; Default box value.
 (description description)
 (cite)
 (from from)
 (to to)
 (when when)
 (parameter_names)
 (entries entries ...)
 (time_columns time_columns) >
```

- *numbers*: submodel (see section 4.1.7) sequence  
Optional submodel  
Initial numeric values. By default, none.

- < (value *value*)  
     (name *name*) >
- *value*: number (dimension not specified)  
     State variable  
     Numeric value.
- *name*: string (see section 4.1.5)  
     State variable  
     Name to refer to number with.
- *where*: string (see section 4.1.5)  
     Optional parameter  
     Name of the extern log to use. By default, use the model name.

## 44.5 table

A ‘select’ model (see 44.3, page 234) build into Daisy.

Write results in a tabular Daisy log file.

```
< table (where where)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 ;; Shared parameters are described in section 44.3.
 (summary)
 (volume volume) ; Default box value.
 (description description)
 (cite)
 (from from)
 (to to)
 (when when)
 (parameter_names)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial) >
```

- *where*: string (see section 4.1.5)  
     Parameter  
     Name of the log file to create.
- *print\_header*: string (see section 4.1.5)  
     Parameter (default ‘true’)  
     If this is set to ‘false’, no header is printed. If this is set to ‘true’, a full header is printed. If this is set to ‘fixed’, a small fixed size header is printed.
- *print\_tags*: boolean (see section 4.1.2)  
     Parameter (default true)  
     Print a tag line in the file.
- *print\_dimension*: boolean (see section 4.1.2)  
     Parameter (default true)  
     Print a line with units after the tag line.

- *flush*: boolean (see section 4.1.2)  
Parameter (default false)  
Flush to disk after each entry (for debugging).
- *record\_separator*: string (see section 4.1.5)  
Parameter (default ' ')  
String to print between records (time steps).
- *field\_separator*: string (see section 4.1.5)  
Parameter (default ' ')  
String to print between fields.
- *missing\_value*: string (see section 4.1.5)  
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 4.1.5)  
Parameter (default ' ')  
String to print between array entries.

## 44.6 column

A 'table' model (see 44.5, page 236) defined in 'log-std.dai'.

A log table for a specific column.

```
< column (column "*")
 (parameter_names parameter_names ...) ; Has default value.
 (colfid colfid) ; Default cond value.
 ;; Shared parameters are described in section 44.5.
 (summary)
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (where where)
 (when when)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ") >
```

- *column*: string (see section 4.1.5)  
Parameter (default '\*')  
Name of column to log. Use "\*" to log all columns.
- *colfid*: **string** component (see chapter 72)  
Component (default 'cond')

```
(colfid cond ("string-equal" "${column}" "*")
 "")
 ((true)
 "${column}_"))
```

Parameter description:

File component name indicating column logged.

## 44.7 Field nitrogen

A ‘column’ model (see 44.6, page 237) 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 it 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.

```
< "Field nitrogen" (summary summary ...) ; Has default value.
 (unit "kg N/ha")
 (where where) ; Has default value.
 (when when) ; Default hourly value.
 (entries entries ...) ; Has default value.
 ;; Shared parameters are described in section 44.6.
 (column "*")
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (parameter_names parameter_names ...) ; Has default value.
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value. >
```

- *unit*: string (see section 4.1.5)  
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 capillary rise.
- *Soil-Drain*: [kg N/ha]  
Loss of nitrogen from the soil matrix to drain pipes.  
  
This includes 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]  
Mineral 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 with a shorter timestep. Or readjust the system to slow down those processes that remove mineral nitrogen.
- *Mineralization*: [kg N/ha]  
NH<sub>4</sub> produced in the soil matrix by mineralization of organic matter.
- *Immobilization*: [kg N/ha]  
NO<sub>3</sub> immobilized as a result of organic matter turnover.
- *Crop-Uptake*: [kg N/ha]  
Nitrogen uptake by the roots.
- *Volatilization*: [kg N/ha]  
NH<sub>4</sub>-fertilizer lost during the application.
- *N<sub>2</sub>O-Nitrification*: [kg N/ha]  
N<sub>2</sub>O production in the soil matrix due to nitrification.  
  
The N<sub>2</sub>O produced is assumed to escape to the atmosphere, and will not be traced by the model.

- *Denitrification*: [kg N/ha]  
Loss of NO<sub>3</sub> 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 exfoliation 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.

## 44.8 Field water

A ‘column’ model (see 44.6, page 237) 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 intercepted water on the canopy. The transpiration covers water removed from the soil through the root system and the stomata on the leaves.  
  
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 capillary rise.
- *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 slope. 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.

## 44.9 Soil nitrogen

A 'column' model (see 44.6, page 237) defined in 'log-std.dai'.

Nitrogen input, output, transformation and content for the soil.

```

< "Soil nitrogen" (summary summary ...) ; Has default value.
 (unit "kg N/ha")
 (where where) ; Has default value.
 (when when) ; Default hourly value.
 (entries entries ...) ; Has default value.
 ;; Shared parameters are described in section 44.6.
 (column "*")
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (parameter_names parameter_names ...) ; Has default value.
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value. >

```

- *unit*: string (see section 4.1.5)  
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 bioincorporation.  
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.

## 44.10 Soil water

A ‘column’ model (see 44.6, page 237) 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

## 44.11 biopore

A 'column' model (see 44.6, page 237) defined in 'log-std.dai'.

A log table for a specific biopore class.

```
< biopore (biopore "*")
 (parameter_names parameter_names ...) ; Has default value.
 (bioporefid bioporefid) ; Default cond value.
 ;; Shared parameters are described in section 44.6.
 (column "*")
 (summary)
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (where where)
 (when when)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value. >
```

- *biopore*: string (see section 4.1.5)  
Parameter (default ‘\*’)  
Name of biopore to log. Use “\*” to log all biopore classes.
- *bioporefid*: **string** component (see chapter 72)  
Component (default ‘cond’)

```
(bioporefid cond (("string-equal" "${biopore}" "*")
 biopores)
 ((true)
 "${biopore}"))
```

Parameter description:

File component name indicating biopore logged.

## 44.12 chemical

A ‘column’ model (see 44.6, page 237) defined in ‘log-std.dai’.

A log table for a specific chemical.

```
< chemical (chemical "*")
 (parameter_names parameter_names ...) ; Has default value.
 (chemfid chemfid) ; Default cond value.
 ;; Shared parameters are described in section 44.6.
 (column "*")
 (summary)
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (where where)
 (when when)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value. >
```

- *chemical*: string (see section 4.1.5)  
Parameter (default ‘\*’)  
Name of chemical to log.
- *chemfid*: **string** component (see chapter 72)  
Component (default ‘cond’)

```
(chemfid cond (("string-equal" "${chemical}" "*")
 chemicals)
 ((true)
 "${chemical}"))
```

Parameter description:

File name component indicating chemical logged.

## 44.13 Field chemical

A ‘chemical’ model (see 44.12, page 246) defined in ‘log-std.dai’.

Content, transport and transformation of chemicals in the field.

```
< "Field chemical" (summary summary ...) ; Has default value.
 (unit "g/ha")
 (where where) ; Has default value.
 (when when) ; Default hourly value.
 (entries entries ...) ; Has default value.
 ;; Shared parameters are described in section 44.12.
 (chemical "*")
 (column "*")
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (parameter_names parameter_names ...) ; Has default value.
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value.
 (chemfid chemfid) ; Default cond value. >
```

- *unit*: string (see section 4.1.5)  
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]  
Atmospheric deposition.  
  
This is usually zero when tracking pesticides, but can be relevant for atmospheric pollutants, and is also significant 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 significant 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 temporarily non-zero if some process remove the chemical fast compared to the timestep of the simulation.



## 44.14 Soil chemical

A ‘chemical’ model (see 44.12, page 246) defined in ‘log-std.dai’.

Content, transport and transformation of chemicals in the soil.

```
< "Soil chemical" (summary summary ...) ; Has default value.
 (unit "g/ha")
 (where where) ; Has default value.
 (when when) ; Default hourly value.
 (entries entries ...) ; Has default value.
 ;; Shared parameters are described in section 44.12.
 (chemical "*")
 (column "*")
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (parameter_names parameter_names ...) ; Has default value.
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value.
 (chemfid chemfid) ; Default cond value. >
```

- *unit*: string (see section 4.1.5)  
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 transformation.
- *Content*: [g/ha]  
Total amount in the specified soil interval.
- *Error*: [g/ha]  
Borrowed from the future.

This can be non-zero if some process remove the chemical too fast compared to the simulation timestep.

## 44.15 crop

A ‘column’ model (see 44.6, page 237) defined in ‘log-std.dai’.

A log table for a specific crop.

```
< crop (crop "*")
 (parameter_names parameter_names ...) ; Has default value.
 (cropfid cropfid) ; Default cond value.
 ;; Shared parameters are described in section 44.6.
 (column "*")
 (summary)
 (volume volume) ; Default box value.
 (description description) ; Has default value.
 (cite)
 (from from)
 (to to)
 (where where)
 (when when)
 (print_header true)
 (print_tags true)
 (print_dimension true)
 (entries entries ...)
 (time_columns time_columns)
 (print_initial print_initial)
 (flush false)
 (record_separator " ")
 (field_separator " ")
 (missing_value "00.00")
 (array_separator " ")
 (colfid colfid) ; Default cond value. >
```

- *crop*: string (see section 4.1.5)  
Parameter (default ‘\*’)  
Name of crop to log. Use ‘\*’ to log all crops.
- *cropfid*: **string** component (see chapter 72)  
Component (default ‘cond’)

```
(cropfid cond ("string-equal" "${crop}" "*")
 crops)
 (true)
 "${crop}"))
```

Parameter description:

File component name indicating crop logged.



# Chapter 45

## macro

Preferential flow in soil macro pores.

### 45.1 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.

```
< default (distribution distribution)
 (height_start height_start)
 (height_end height_end)
 (pressure_initiate -3 [cm])
 (pressure_end -30 [cm])
 (pond_max 0.5 [mm]) >
```

- *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<sub>p</sub>*: number [**h<sup>-1</sup>**]  
Macropore sink term.

## 45.2 none

No macropores.

## Chapter 46

# mactrans

Macropore transportation of solutes.

### 46.1 default

Solute follows water.

Used by tertiary old mactrans (see 75.3, page 397) .





# Chapter 47

## movement

This component handles the movement in the soil.

```
< component (Tertiary Tertiary) >
```

- *Tertiary*: **tertiary** component (see chapter 75)  
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 succeeded, and 0 if the first model failed but the second succeeded.
- *solute\_failure\_level*: integer  
The number of the last solute transport model to fail. It is -1 if the first model succeeded, and 0 if the first model failed but the second succeeded.

### 47.1 solute

Shared parameters for handling solutes.

```
< solute (matrix_solute matrix_solute ...)
 (matrix_solid matrix_solid) ; Default none value.
 (sink_sorbed true)
 ;; Shared parameters are described in chapter 47.
 (Tertiary Tertiary) >
```

- *matrix\_solute*: **transport** component (see chapter 78) 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 78)  
Component (default 'none')  
Matrix solute transport model used for fully sorbed constituents.
- *sink\_sorbed*: boolean (see section 4.1.2)  
Parameter (default true)  
Subtract sink term from sorbed matter.

### 47.2 rectangle

A 'solute' model (see 47.1, page 257) build into Daisy.

Two dimensional movement in a rectangular grid.

```
< rectangle (Tertiary Tertiary) ; Default none value.
 (matrix_solute matrix_solute ...) ; Has default value.
 (Geometry Geometry) ; Has partial value.
 (matrix_water matrix_water ...) ; Has default value.
 (drainpoints)
 (heat heat) ; Default Mollerup value.
 ;; Shared parameters are described in section 47.1.
 (matrix_solid matrix_solid) ; Default none value.
 (sink_sorbed true) >
```

- *Geometry*: **GeometryRect** fixed component (see section 93.26)  
Submodel (has partially specified default value)  
Discretization of the soil.
- *matrix\_water*: **uzrect** component (see chapter 85) 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 4.1.7) 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 39)  
Component (default 'Mollerup')  
Heat transport model.

## 47.3 vertical

A 'solute' model (see 47.1, page 257) build into Daisy.

One dimensional movement.

Used by column default Movement (see 24.1, page 138) .

```
< vertical (Tertiary Tertiary) ; Default old value.
 (matrix_solute matrix_solute ...) ; Has default value.
 (Geometry Geometry) ; Has default value.
 (matrix_water matrix_water ...) ; Has default value.
 ;; Shared parameters are described in section 47.1.
 (matrix_solid matrix_solid) ; Default none value.
 (sink_sorbed true) >
```

- *Geometry*: **Geometry1D** fixed component (see section 93.24)  
Submodel (has fully specified default value)  
Discretization of the soil.
- *matrix\_water*: **uzmodel** component (see chapter 84) 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 48

## 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 emmissivity (range 0-1).
- *black\_body\_radiation*: number [ $\mathbf{W/m^2}$ ]  
Radiation emitted by black bodies at current air temperature. Stefan-Boltzmann's law.

### 48.1 brunt

Brunt, 1932. Default parameterization by Jensen et.al., 1990. FAO recommendation.

Used by bioclimate default net\_radiation (see 18.1, page 107) .

```
< brunt (b 0.14 [1/sqrt(kPa)])
 (a 0.34 []) >
```

- *b*: number [ $\mathbf{1/sqrt(kPa)}$ ]  
Parameter (default 0.14)  
Brunt 'b' parameter (vapor pressure factor).
- *a*: number (dimensionless)  
Parameter (default 0.34)  
Brunt 'a' parameter (offset).

## **48.2 brutsaert**

Brutsaert, 1975

## **48.3 idso\_jackson**

Idso and Jackson, 1969

## **48.4 prata**

Prata, 1996

## **48.5 satterlund**

Satterlund, 1979

## **48.6 swinbank**

Swinbank, 1963

# Chapter 49

## nitrification

The nitrification process, transforming ammonium into nitrate and nitrous oxide.

Used by horizon component Nitrification (see 40, page 199) .

```
< component (N2O_fraction 0.02 [<fraction>]) >
```

- *N2O\_fraction*: number [**<fraction>**]  
Parameter (default 0.02)  
Fraction of ammonium lost as N2O.

### 49.1 soil

$k_{10} * M / (k + M)$ . Michaelis-Menten kinetics, with nitrification based on total ammonium content.

Used by horizon component Nitrification (see 40, page 199) .

```
< soil (k 5e-005 [g N/cm3])
 (k_10 2.08333e-007 [g N/cm3/h])
 (heat_factor heat_factor) ; Has default value.
 (water_factor water_factor) ; Has default value.
 ;; Shared parameters are described in chapter 49.
 (N2O_fraction 0.02 [<fraction>]) >
```

- *k*: number [**g N/cm<sup>3</sup>**]  
Parameter (default 5e-005)  
Half saturation constant.
- *k\_10*: number [**g N/cm<sup>3</sup>/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.

### 49.2 solute

$k_{10} * C / (k + C)$ . Michaelis-Menten kinetics, with nitrification based on ammonium solute.

```

< solute (k k)
 (k_10 k_10)
 (heat_factor heat_factor) ; Has default value.
 (water_factor water_factor) ; Has default value.
 ;; Shared parameters are described in chapter 49.
 (N2O_fraction 0.02 [<fraction>]) >

```

- $k$ : number [ $\text{g}/\text{cm}^3$ ]  
Parameter  
Half saturation constant.
- $k_{10}$ : number [ $\text{h}^{-1}$ ]  
Parameter  
Max rate.
- $heat\_factor$ : plf [ $\text{dg C} \rightarrow \text{<none>}$ ]  
Parameter (has default value with 0 points)  
Heat factor.
- $water\_factor$ : plf [ $\text{cm} \rightarrow \text{<none>}$ ]  
Parameter (has default value with 0 points)  
Water potential factor.



# Chapter 50

## number

Generic representation of numbers.

Used by select component `expr` (see 67, page 349) , and biopore component `density` (see 19, page 113) .

### 50.1 \*

Use the product of its operands.

`< "*" operands... >`

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

### 50.2 +

Use the sum of its operands.

`< "+" operands... >`

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

### 50.3 -

Negate number or subtract numbers. With one operand, negates it. With more than one operand, subtracts all but the first from the first.

`< "-" operands... >`

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

### 50.4 /

Divide the first operand by the rest.

`< "/" operands... >`

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

## 50.5 child

Numbers based on another number.

```
< child (value value) >
```

- *value*: **number** component (see chapter 50)  
Operand for this function.

## 50.6 convert

A ‘child’ model (see 50.5, page 266) build into Daisy.

Convert to specified dimension.

```
< convert value dimension >
```

- *value*: **number** component (see chapter 50)  
Operand for this function.
- *dimension*: string (see section 4.1.5)  
Parameter  
Dimension to convert to.

## 50.7 dim

A ‘child’ model (see 50.5, page 266) build into Daisy.

Specify dimension for number.

```
< dim value dimension
 (warn_known true) >
```

- *value*: **number** component (see chapter 50)  
Operand for this function.
- *dimension*: string (see section 4.1.5)  
Parameter  
Dimension to use.
- *warn\_known*: boolean (see section 4.1.2)  
Parameter (default true)  
Issue a warning if the dimensions is already known.

## 50.8 identity

A ‘child’ model (see 50.5, page 266) build into Daisy.

Pass value unchanged.

```
< identity (dimension dimension)
 (description description)
 (cite)
 ;; Shared parameters are described in section 50.5.
 (value value) >
```

- *dimension*: string (see section 4.1.5)  
Optional parameter  
Dimension of this value.

- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.

## 50.9 const

Always give the specified value.

Used by chemical default initial (see 22.1, page 121) .

< **const** *value* >

- *value*: number [<**user**>]  
Parameter  
Fixed value for this number.

## 50.10 initial\_zero

A ‘const’ model (see 50.9, page 267) build into Daisy.

Initial zero content.

## 50.11 zero\_gradient

A ‘const’ model (see 50.9, page 267) build into Daisy.

Assume same concentration in groundwater as in the bottom of the soil profile.

## 50.12 depth

Find soil value at specific depth.

< **depth** (*column column*)  
(*h h*)  
(*z z*) >

- *column*: **column** component (see chapter 24)  
The soil column whose properties we want to examine.
- *h*: **number** component (see chapter 50)  
The tension we want to compare with.
- *z*: **number** component (see chapter 50)  
The height we want to compare with.

## 50.13 depth\_K

A ‘depth’ model (see 50.12, page 267) build into Daisy.

Find water conductivity (K) for a given pressure (h).

## 50.14 depth\_Theta

A ‘depth’ model (see 50.12, page 267) build into Daisy.  
Find water content (Theta) for a given pressure (h).

## 50.15 exp

Take the exponential of its argument.

```
< exp operand >
```

- *operand*: **number** component (see chapter 50)  
Operand for this function.

## 50.16 fetch

Fetch the value and dimension in the current scope.

```
< fetch name >
```

- *name*: string (see section 4.1.5)  
Parameter  
Name of a the symbol.

## 50.17 get

Get the value of symbol in the current scope.

```
< get name dimension >
```

- *name*: string (see section 4.1.5)  
Parameter  
Name of a the symbol.
- *dimension*: string (see section 4.1.5)  
Parameter  
Expected dimension for the symbol.

## 50.18 horizon

Find soil value at specific horizon.

```
< horizon (horizon horizon)
 (h h)
 (top_soil top_soil) >
```

- *horizon*: **horizon** component (see chapter 40)  
The soil horizon whose properties we want to examine.
- *h*: **number** component (see chapter 50)  
The tension we want to compare with.
- *top\_soil*: boolean (see section 4.1.2)  
Parameter  
Set this to true for the A horizon.

## 50.19 soil\_K

A ‘horizon’ model (see 50.18, page 268) build into Daisy.

Find hydraulic conductivity (K) for a given pressure (h).

## 50.20 soil\_Theta

A ‘horizon’ model (see 50.18, page 268) build into Daisy.

Find water content (Theta) for a given pressure (h).

## 50.21 soil\_heat\_capacity

A ‘horizon’ model (see 50.18, page 268) build into Daisy.

Find heat capacity for a given pressure (h).

## 50.22 soil\_heat\_conductivity

A ‘horizon’ model (see 50.18, page 268) build into Daisy.

Find heat conductivity for a given pressure (h).

## 50.23 if

Select between two numbers depending on a boolean expression.

`< if if then else >`

- *if*: **boolean** component (see chapter 20)  
Select which number to use.
- *then*: **number** component (see chapter 50)  
Use this if true.
- *else*: **number** component (see chapter 50)  
Use this if false.

## 50.24 initial\_C

Find initial content from concentration.

Used by chemical NO3 initial (see 22.19, page 132) , and chemical NH4 initial (see 22.18, page 132) .

`< initial_C C >`

- *C*: number [g/cm<sup>3</sup>]  
Parameter  
Initial concentration in soil water with sorbtion ignored. That is  $M = C * \text{Theta}$ .

## 50.25 initial\_NH4

A ‘initial\_C’ model (see 50.24, page 269) build into Daisy.

Initial NH4 concentration in soil water.

## 50.26 initial\_NO3

A 'initial\_C' model (see 50.24, page 269) build into Daisy.  
Initial NO3 concentration in soil water.

## 50.27 let

Bind symbols in 'clauses' in a new scope, and evaluate 'expr' in that scope.

< let *clauses*... *expr* >

- *clauses*: submodel (see section 4.1.7) sequence  
List of identifiers and values to bind in this scope.
  - < *identifier* *expr* >
    - *identifier*: string (see section 4.1.5)  
Parameter  
Identifier to bind.
    - *expr*: **number** component (see chapter 50)  
Value to give it.
- *expr*: **number** component (see chapter 50)  
Expression to evaluate.

## 50.28 ln

Take the natural logarithm of its argument.

< ln *operand* >

- *operand*: **number** component (see chapter 50)  
Operand for this function.

## 50.29 log10

Take the base 10 logarithm of its argument.

< log10 *operand* >

- *operand*: **number** component (see chapter 50)  
Operand for this function.

## 50.30 max

Use the largest value of its operands.

< max *operands*... >

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

## 50.31 min

Use the smallest value of its operands.

< min *operands*... >

- *operands*: **number** component (see chapter 50) sequence  
The operands for this function.

## 50.32 plf

Look up argumen in a piecewise linear function.

```
< plf operand
 (domain "<unknown>")
 (range "<unknown>")
 (points points ...) >
```

- *operand*: **number** component (see chapter 50)  
Operand for this function.
- *domain*: string (see section 4.1.5)  
Parameter (default ‘<unknown>’)  
Unit for the operand of the function.
- *range*: string (see section 4.1.5)  
Parameter (default ‘<unknown>’)  
Unit for the operand of the function.
- *points*: submodel (see section 4.1.7) 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.

## 50.33 pow

Raise 'base' to the power of 'exponent'.

```
< pow base exponent >
```

- *base*: **number** component (see chapter 50)  
The base operand for this function.
- *exponent*: **number** component (see chapter 50)  
The exponent operand for this function.

## 50.34 soil\_h

Find pressure (h) for a given water content (Theta).

```
< soil_h (horizon horizon)
 (Theta Theta)
 (top_soil top_soil) >
```

- *horizon*: **horizon** component (see chapter 40)  
The soil horizon whose properties we want to examine.
- *Theta*: **number** component (see chapter 50)  
The water content we want to compare with.
- *top\_soil*: boolean (see section 4.1.2)  
Parameter  
Set this to true for the A horizon.

### 50.35 source

Extract information from a time series.

```
< source (source source)
 (begin begin)
 (end end) >
```

- *source*: **source** component (see chapter 70)  
The time series we want to extract a number from.
- *begin*: **Time** fixed component (see section 93.21)  
Optional submodel  
Ignore values before or at this date.
- *end*: **Time** fixed component (see section 93.21)  
Optional submodel  
Ignore values after this date.

### 50.36 source\_average

A ‘source’ model (see 50.35, page 272) build into Daisy.  
Find average number in time series.

### 50.37 source\_increase

A ‘source’ model (see 50.35, page 272) build into Daisy.  
Find increase in value during time series.

### 50.38 source\_sum

A ‘source’ model (see 50.35, page 272) build into Daisy.  
Calculate the sum of the values in a time series.

### 50.39 source\_unique

A ‘source’ model (see 50.35, page 272) build into Daisy.  
Find unique number in time series.

### 50.40 sqr

Take the square of its argument.

```
< sqr operand >
```

- *operand*: **number** component (see chapter 50)  
Operand for this function.

### 50.41 sqrt

Take the square root of its argument.

```
< sqrt operand >
```

- *operand*: **number** component (see chapter 50)  
Operand for this function.



**50.42    *x***

The value of the symbol '*x*' in the current scope.

Used by `xysource loop x` (see 91.5, page 469) .



# Chapter 51

## organic

Turnover of organic matter in the soil.

### 51.1 default

Mineralization and immobilization in soil. See also [Hansen et al., 1991, Bruun et al., 2003]

Used by column default OrganicMatter (see 24.1, page 138) , and column Foulum OrganicMatter (see 24.3, page 140) .

```
< default (ClayOM ClayOM) ; Default old value.
 (domsorp) ;
 (Bioincorporation Bioincorporation) ; Has default value.
 (am am ...) ; Has default value.
 (description description)
 (cite cite ...) ; Has default value.
 (heat_factor heat_factor) ; Has default value.
 (water_factor water_factor) ; Has default value.
 (active_underground false)
 (K_NH4 0.020833 [h-1])
 (K_NO3 0.020833 [h-1])
 (CO2_threshold 0.0001 [h-1])
 (buffer buffer) ; Has default value.
 (smb smb ...) ; Has default value.
 (som som ...) ; Has default value.
 (initial_SOM initial_SOM ...)
 (dom)
 (smb_tillage_factor)
 (som_tillage_factor)
 (min_AM_C 0.5 [g C/m2])
 (min_AM_N 0.05 [g N/m2])
 (init init) ; Has default value. >
```

- *ClayOM*: **ClayOM** component (see chapter 9)  
Component (default 'old')  
Clay effect model.
- *domsorp*: **domsorp** component (see chapter 30) sequence  
Component (default: an empty sequence)  
Interchange between DOM and SOM pools.
- *Bioincorporation*: **Bioincorporation** fixed component (see section 93.3)

Submodel (has fully specified default value)  
 Biological incorporation of litter.

- *am*: **am** component (see chapter 16) sequence  
 Component (has default value with length 1)

(**am root**)

Parameter description:  
 Added organic matter pools.

- *description*: string (see section 4.1.5)  
 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 4.1.5) 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 parameterization.

- *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 4.1.2)  
 Parameter (default false)  
 Set this flag to turn on mineralization below the root zone.
- *K\_NH4*: number [**h**<sup>-1</sup>]  
 Parameter (default 0.020833)  
 Maximal immobilization rate for ammonium.
- *K\_NO3*: number [**h**<sup>-1</sup>]  
 Parameter (default 0.020833)  
 Maximal immobilization rate for nitrate.
- *CO2\_threshold*: number [**h**<sup>-1</sup>]  
 Parameter (default 0.0001)  
 Turnover rate above which pools will contribute to 'CO2\_fast'.
- *buffer*: submodel (see section 4.1.7)  
 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/cm³] soil cells
State variable (default: an empty sequence)
Buffer nitrogen content.
 - *C*: number [g C/cm³] 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 [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. Overrides 'turnover_rate' if specified.

- *smb*: **SMB** component (see chapter 12) 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 13) 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 4.1.7) 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<sup>2</sup>**]  
Parameter  
Organic carbon content of this layer.
- *dom*: **DOM** fixed component (see section 93.20) 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<sup>2</sup>**]  
Parameter (default 0.5)  
Minimal amount of carbon in AOM ensuring it is not removed.
- *min\_AM\_N*: number [**g N/m<sup>2</sup>**]  
Parameter (default 0.05)  
Minimal amount of nitrogen in AOM ensuring it is not removed.
- *init*: submodel (see section 4.1.7)  
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 unspecified, the input rate from the initial 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 not 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 specified, 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 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 4.1.2)  
Parameter (default true)  
Print summari information for each row.
- *debug\_to\_screen*: boolean (see section 4.1.2)  
Parameter (default false)  
If true, print debug information to screen, else to the 'daisy.log' file.
- *top\_summary*: string (see section 4.1.5)  
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 [**g/cm<sup>3</sup>**] soil cells  
Total organic matter in the soil layer.
- *CO2*: number [**g CO2-C/cm<sup>3</sup>/h**] soil cells  
CO2 evolution in soil from all pools.
- *NO3\_source*: number [**g N/cm<sup>3</sup>/h**] soil cells  
Mineralization this time step (negative numbers mean immobilization).
- *NH4\_source*: number [**g N/cm<sup>3</sup>/h**] soil cells  
Mineralization this time step (negative numbers mean immobilization).
- *fertilized\_N*: number [**g N/cm<sup>2</sup>/h**]  
Amount of organic bound nitrogen applied. This includes nitrogen incorporated directly in the soil.
- *fertilized\_C*: number [**g C/cm<sup>2</sup>/h**]  
Amount of organic bound carbon applied. This includes carbon incorporated directly in the soil.
- *tillage\_N\_top*: number [**g N/m<sup>2</sup>/h**]  
Amount of nitrogen added to surface during tillage. This is a negative number.

- *tillage\_C\_top*: number [**g C/m<sup>2</sup>/h**]  
Amount of carbon added to surface during tillage. This is a negative number.
- *tillage\_N\_soil*: number [**g N/cm<sup>3</sup>/h**] soil cells  
Amount of nitrogen added to soil during tillage.
- *tillage\_C\_soil*: number [**g C/cm<sup>3</sup>/h**] soil cells  
Amount of carbon added to surface during tillage.
- *total\_C*: number [**g C/cm<sup>3</sup>**] soil cells  
Total organic C in the soil layer.
- *total\_N*: number [**g N/cm<sup>3</sup>**] soil cells  
Total organic N in the soil layer.
- *CO2\_fast*: number [**g CO<sub>2</sub>-C/cm<sup>3</sup>/h**] soil cells  
CO<sub>2</sub> evolution in soil from pools faster than 'CO<sub>2</sub>\_threshold'.
- *top\_CO2*: number [**g CO<sub>2</sub>-C/cm<sup>2</sup>/h**]  
CO<sub>2</sub> evolution at surface.
- *top\_DM*: number [**kg DM/m<sup>2</sup>**]  
Added organic dry matter on top of surface.
- *abiotic\_factor*: number (dimensionless) soil cells  
Product of current heat and water factors.

## 51.2 none

Ignore all soil organic matter dynamics.

# Chapter 52

## 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 93.27, page 516) .

### 52.1 file

Read a setup file containing lots of parentheses.

< file *where* >

- *where*: string (see section 4.1.5)  
Parameter  
File to read from.



# Chapter 53

## pet

The 'pet' component should calculate the potential evapotranspiration from meteorological data, as well as the crop and soil state.

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

- *description*: string (see section 4.1.5)  
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 4.1.5) sequence  
Parameter (default: an empty sequence)  
BibTeX keys that would be relevant for this model or parameterization.

### 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.

### 53.1 FAO\_PM

Potential evapotranspiration using Penman-Monteith.

```
< FAO_PM (rb 20 [s/m])
 (use_wet true)
 ;; Shared parameters are described in chapter 53.
 (description description)
 (cite) >
```

- *rb*: number [s/m]  
Parameter (default 20)  
Boundary layer resistance for wet surface.
- *use\_wet*: boolean (see section 4.1.2)  
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 [W/m<sup>2</sup>]  
Reference net radiation.
- *G*: number [W/m<sup>2</sup>]  
Soil heat flux.

## 53.2 Hargreaves

Potential evapotranspiration based on temperature. See also [Hargreaves and Samani, 1985]

## 53.3 PM

Potential evapotranspiration using Penman-Monteith.

- ```
< PM  (rb 20 [s/m])
      ;; Shared parameters are described in chapter 53.
      (description description)
      (cite) >
```
- *rb*: number [s/m]
Parameter (default 20)
Boundary layer resistance for wet surface. used for bare soil only.

53.4 makkink

Potential evapotranspiration using Makkink's Equation.

53.5 weather

Potential evapotranspiration using weather data.

Chapter 54

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.

54.1 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 tempeprature sum. Cut stress and leaf respiration does not affect this phenology model.

```
< TSum (EmrTSum EmrTSum)
    (EmrThrs 0 [dg C])
    (VegTSum VegTSum)
    (VegThrs 0 [dg C])
    (RepTSum RepTSum)
    (RepThrs 0 [dg C])
    ;; Shared parameters are described in chapter 54.
    (DS -1 [])
    (day_length 0 [h])
    (DAP 0 [d])
    (partial_day_length 0 [h]) >
```

- *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.

54.2 default

Default crop phenology model.

Used by crop Maize Devel (see 27.15, page 156) , crop Pioneer Maize Devel (see 27.22, page 157) , crop Ikuwala Maize Devel (see 27.16, page 156) , crop Silage Maize Devel (see 27.17, page 156) , crop Pea Devel (see 27.20, page 157) , crop Potato; Koege Devel (see 27.23, page 157) , crop Potato; SCRI Devel (see 27.24, page 157) , crop Potato; FertOrgaNic Devel (see 27.25, page 157) , crop Potato; Agria Devel (see 27.26, page 157) , crop Potato; Folva Devel (see 27.27, page 157) , crop Potato; Triada Devel (see 27.29, page 158) , crop Rye Devel (see 27.33, page 158) , crop Spring Barley Devel (see 27.36, page 158) , crop Spring Barley; Foulum Devel (see 27.37, page 159) , crop Sugar Beet Devel (see 27.40, page 159) , crop Fodder Beet Devel (see 27.10, page 156) , crop Spring Rape Devel (see 27.38, page 159) , crop Spring Wheat Devel (see 27.39, page 159) , crop Winter Barley Devel (see 27.49, page 160) , crop Winter Barley; Koge Devel (see 27.51, page 160) , crop Winter Rape Devel (see 27.52, page 160) , crop Winter Wheat Devel (see 27.53, page 160) , crop Winter Wheat; Eest Devel (see 27.54, page 160) , crop Winter Wheat; Foulum Devel (see 27.55, page 160) , crop Grass to grain Devel (see 27.13, page 156) , crop Grass Devel (see 27.14, page 156) , crop Ryegrass Devel (see 27.34, page 158) , crop Wclover Devel (see 27.45, page 159) , crop Beetroot Devel (see 27.3, page 155) , crop Broccoli Devel (see 27.4, page 155) , crop Broccoli - transplanted Devel (see 27.5, page 155) , crop Brussels sprouts Devel (see 27.6, page 155) , crop Brussels sprouts - transplanted Devel (see 27.7, page 155) , crop Celeriac Devel (see 27.8, page 155) , crop Celeriac - transplanted Devel (see 27.9, page 156) , crop Potato; SCRI - AArhus Devel (see 27.30, page 158) , crop Early potato Devel (see 27.31, page 158) , crop White cabbage Devel (see 27.46, page 160) , crop White cabbage - transplanted Devel (see 27.47, page 160) , crop Early white cabbage - transplanted Devel (see 27.48, page 160) , crop Onion Devel (see 27.18, page 156) , crop Onion -

planting of sets Devel (see 27.19, page 157) , crop Rug Devel (see 27.32, page 158) , crop Vaarbyg Devel (see 27.41, page 159) , crop Vinterbyg Devel (see 27.42, page 159) , crop Vinterhvede Devel (see 27.43, page 159) , crop Froegraes Devel (see 27.11, page 156) , crop Graes Devel (see 27.12, page 156) , crop Silomajs Devel (see 27.35, page 158) , crop Aert Devel (see 27.2, page 155) , and crop Vinterraps Devel (see 27.44, page 159) .

```
< default (EmrTSum EmrTSum)
           (EmrSMF EmrSMF)           ; Has default value.
           (DS_Emr 0.01 [DS])
           (DSRate1 DSRate1)
           (DSRate2 DSRate2)
           (TempEff1 TempEff1)
           (TempEff2 TempEff2)
           (PhotEff1 PhotEff1)
           (DSMature 2 [DS])
           (DSRepeat 4 [DS])
           (DSSetBack 1.7 [DS])
           (defined_until_ds 2 [DS])
           ;; Shared parameters are described in chapter 54.
           (DS -1 [])
           (day_length 0 [h])
           (DAP 0 [d])
           (partial_day_length 0 [h]) >
```

- *EmrTSum*: number [dg C d]
Parameter
Soil temperature sum at emergence.
- *EmrSMF*: plf [cm → <none>]
Parameter (has default value with 4 points)

```
(EmrSMF (-1000 1) (-150 1) (-50 1) (-30 1))
```

Parameter description:
Soil moisture (h-function) effect on emergence.

- *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.

Chapter 55

photosynthesis

Leaf photosynthesis.

```
< component (description description)  
            (cite)  
            (min_PAR 0.1 [W/m2]) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *min_PAR*: number [W/m²]
Parameter (default 0.1)
Minimum PAR at top of canopy for photosynthesis. If radiation is below this amount, photosynthesis will be disabled.

55.1 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.

```
< Farquhar (Xn 0.00116 [mol/mol/s])  
            (Gamma25 3.69 [Pa])  
            (Ea_Gamma 29000 [J/mol])  
            ("N-dist" N-dist) ; Default exp value.  
            (Stomatacon Stomatacon) ; Default Leuning value.  
            ;; Shared parameters are described in chapter 55.  
            (description description)  
            (cite)  
            (min_PAR 0.1 [W/m2]) >
```

- *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. $\Gamma_{25} = 3.69$ Pa for wheat (Collatz et al., 1991)

- *Ea_Gamma*: number [J/mol]
Parameter (default 29000)
Activation energy for Γ . $Ea_{\Gamma} = 29000$ (Jordan & Ogren, 1984)
- *N-dist*: **rubiscoNdist** component (see chapter 62)
Component (default 'exp')
Rubisco N-distribution in the canopy layer.
- *Stomatacon*: **stomatacon** component (see chapter 71)
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/m²/s] canopy intervals
Photosynthetic capacity in each layer.
- *Jm_vector*: number [mmol/m²/s] canopy intervals
Potential rate of electron transport in each layer.
- *pn_vector*: number [mol/m² 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/m² leaf/s] canopy intervals
Stomata conductance in each layer.
- *Nleaf_vector*: number [mol N/m²] canopy intervals
Distribution of photosynthetic N-leaf.
- *Ass_vector*: number [mol CH₂O/m²/h] canopy intervals
Brutto assimilate.
- *LAI_vector*: number [m² leaf/m² field] canopy intervals
LAI.
- *ci_middel*: number [Pa]
Stomata average CO₂ pressure.
- *Gamma*: number [Pa]
CO₂ compensation point of photosynthesis.

- *gbw*: number [**mol/m² leaf/s**]
Boundary lauer conductance.
- *gs*: number [**mol/m² field/s**]
Stomata conductance.
- *gs_ms*: number [**m/s**]
Stomata conductance.
- *Ass*: number [**g CH₂O/m²/h**]
'Net' leaf assimilate of CO₂ (brutto photosynthesis).
- *Res*: number [**g CH₂O/m²/h**]
Farquhar leaf respiration.
- *PAR_*: number [**mol/m²/h**]
PAR.
- *Vmax*: number [[**mmol/m²/s**]]
Photosynthetic Rubisco capacity.
- *jm*: number [[**mmol/m²/s**]]
Potential rate of electron transport.
- *leafPhotN*: number [[**mol N/m²**]]
Content of photosynthetic active leaf N.
- *fraction_total*: number []
Fraction of leaf contributing to the photosynthesis.

55.2 FC_C3

A 'Farquhar' model (see 55.1, page 291) build into Daisy.

Photosynthesis for C3 crops described by Faquhar et al. (1980).

- ```

< FC_C3 (S 710 [J/mol/K])
 (TempEff TempEff)
 (beta 0.95 [?])
 (theta 0.7 [?])
 (Kc25 40.4 [Pa])
 (Ko25 24800 [Pa])
 (H 220000 [J/mol])
 (c_Vm 26.35 [?])
 (Ea_Vm 65330 [J/mol])
 (Eda_Vm 202900 [J/mol])
 (Ea_Jm 37000 [J/mol])
 (Ea_ko 36000 [J/mol])
 (Ea_kc 59400 [J/mol])
 (Ea_rd 66400 [J/mol])
 (Sv 650 [J/mol/K])
 (alfa 0.08 [mol/mol])
 ;; Shared parameters are described in section 55.1.
 (description description)
 (cite)
 (min_PAR 0.1 [W/m2])
 (Xn 0.00116 [mol/mol/s])
 (Gamma25 3.69 [Pa])
 (Ea_Gamma 29000 [J/mol])
 ("N-dist" N-dist) ; Default exp value.
 (Stomatacon Stomatacon) ; Default Leuning value. >

```
- *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)  
Michaelis-Menten constant of Rubisco for CO<sub>2</sub>. Kc25 = 40.4 Pa for wheat (Collatz et al.,1991)
  - *Ko25*: number [**Pa**]  
Parameter (default 24800)  
Michaelis-Menten constant of Rubisco for O<sub>2</sub> 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)  
Deactivation energy for Vmax.  $Eda_{Vm} = 202900$  J/mol
- *Ea\_Jm*: number [J/mol]  
Parameter (default 37000)  
Activation energy for Jm.  $Ea_{Jm} = 37000$  J/mol (Farquhar et al., 1980).
- *Ea\_ko*: number [J/mol]  
Parameter (default 36000)  
Activation energy for ko.  $Ea_{ko} = 36000$  J/mol (Badger & Collatz, 1977).
- *Ea\_kc*: number [J/mol]  
Parameter (default 59400)  
Activation energy for kc.  $Ea_{kc} = 59400$  J/mol (Badger & Collatz, 1977)
- *Ea\_rd*: number [J/mol]  
Parameter (default 66400)  
Activation 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,

## 55.3 FC\_C4

A ‘Farquhar’ model (see 55.1, page 291) build into Daisy.

C4 photosynthesis and stomatal conductance model by Collatz et al., 1992.

- ```

< FC_C4  (alpha 0.04 [mol/mol])
          (beta 0.93 [?])
          (Q10k 1.8 [?])
          (Q10vm 2.4 [?])
          (Q10rd 2 [?])
          (kj 0.6 [?])
          (paab 0.86 [?])
          (theta 0.83 [?])
          ;; Shared parameters are described in section 55.1.
          (description description)
          (cite)
          (min_PAR 0.1 [W/m2])
          (Xn 0.00116 [mol/mol/s])
          (Gamma25 3.69 [Pa])
          (Ea_Gamma 29000 [J/mol])
          ("N-dist" N-dist)           ; Default exp value.
          (Stomatacon Stomatacon)    ; Default Leuning value. >

```
- *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)
Curvature parameter
 - *Q10k*: number (dimension not specified)
Parameter (default 1.8)
 $Q_{10k} = 1.8$ (Collatz et al., 1992)
 - *Q10vm*: number (dimension not specified)
Parameter (default 2.4)
 $Q_{10vm} = 2.4$ (Collatz et al., 1992)
 - *Q10rd*: number (dimension not specified)
Parameter (default 2)
 $Q_{10rd} = 2.0$ (Collatz et al., 1992)
 - *kj*: number (dimension not specified)
Parameter (default 0.6)
Initial slope of photosynthetic CO₂ response, $k_j = 0.6$ 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

55.4 GL

Goudriaan and Laar, 1978.

Used by crop default LeafPhot (see 27.1, page 153) , crop Maize LeafPhot (see 27.15, page 156) , crop Pioneer Maize LeafPhot (see 27.22, page 157) , crop Pea LeafPhot (see 27.20, page 157) , crop Potato; Koege LeafPhot (see 27.23, page 157) , crop Potato; SCRI LeafPhot (see 27.24, page 157) , crop Potato; FertOrgaNic LeafPhot (see 27.25, page 157) , crop Potato; Folva LeafPhot (see 27.27, page 157) , crop Potato; Triada LeafPhot (see 27.29, page 158) , crop Rye LeafPhot (see 27.33, page 158) , crop Spring Barley LeafPhot (see 27.36, page 158) , crop Sugar Beet LeafPhot (see 27.40, page 159) , crop Fodder Beet LeafPhot (see 27.10, page 156) , crop Spring Rape LeafPhot (see 27.38, page 159) , crop Spring Wheat LeafPhot (see 27.39, page 159) , crop Winter Barley LeafPhot (see 27.49, page 160) , crop Winter Barley; Koge LeafPhot (see 27.51, page 160) , crop Winter Rape LeafPhot (see 27.52, page 160) , crop Winter Wheat LeafPhot (see 27.53, page 160) , crop Winter Wheat; Eest LeafPhot (see 27.54, page 160) , crop Grass to grain LeafPhot (see 27.13, page 156) , crop Ryegrass LeafPhot (see 27.34, page 158) , crop Wclover LeafPhot (see 27.45, page 159) , crop Beetroot LeafPhot (see 27.3, page 155) , crop Broccoli LeafPhot (see 27.4, page 155) , crop Brussels sprouts LeafPhot (see 27.6, page 155) , crop Celeriac LeafPhot (see 27.8, page 155) , crop Potato; SCRI - AArhus LeafPhot (see 27.30, page 158) , crop Early potato LeafPhot (see 27.31, page 158) , crop White cabbage LeafPhot (see 27.46, page 160) , crop Early white cabbage - transplanted LeafPhot (see 27.48, page 160) , crop Onion LeafPhot (see 27.18, page 156) , crop Onion - planting of sets LeafPhot (see 27.19, page 157) , crop Rug LeafPhot (see 27.32, page 158) , crop Vaarbyg LeafPhot (see 27.41, page 159) , crop Vinterbyg LeafPhot (see 27.42, page 159) , crop Vinterhvede LeafPhot (see 27.43, page 159) , crop Froegraes LeafPhot (see 27.11, page 156) , crop Graes LeafPhot (see 27.12, page 156) , crop Silomajs LeafPhot (see 27.35, page 158) , crop Aert LeafPhot (see 27.2, page 155) , and crop Vinterraps LeafPhot (see 27.44, page 159) .

```
< GL  (Qeff Qeff)
      (Fm Fm)
      (TempEff TempEff)
      (DSEff DSEff)           ; Has default value.
      (DAPEff DAPEff)        ; Has default value.
      ;; Shared parameters are described in chapter 55.
      (description description)
      (cite)
      (min_PAR 0.1 [W/m²]) >
```

- *Qeff*: number [(g CO₂/m²/h)/(W/m²)]
Parameter
Quantum efficiency at low light.
- *Fm*: number [g CO₂/m²/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.

Chapter 56

ponddamp

Dampening affect of ponding on soil erosion from rain.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

56.1 EUROSEM

$KH = \exp(-b h)$ See also [Morgan et al., 1998]

```
< EUROSEM (cite cite ...) ; Has default value.  
          (b 2 [mm-1])  
          ;; Shared parameters are described in chapter 56.  
          (description description) >
```

- *b*: number [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.

56.2 Hairsine91

$KH = (h / dds)^{-0.8}$ See also [Hairsine and Rose, 1991]

56.3 Park82

$KH = 2.7183 * \exp(-h / dds)$ See also [Park et al., 1982]

56.4 none

$KH = 1.0$

Used by reaction colgen_Jarvis99 ponddamp (see 60.4, page 323) .

Chapter 57

program

Run a program.

Used by Toplevel @ run (see 93.27, page 516) .

57.1 AM_table

Generate a table of fertilizers.

57.2 Daisy

The Daisy crop/soil/atmosphere model. See also [Hansen et al., 1990, Hansen, 2002, Hansen et al., 1991]

```
< Daisy (weather weather)
      (harvest)
      (time time) ; Has partial value.
      (timestep timestep)
      (stop stop)
      (column column ...)
      (exchange)
      (scope scope) ; Default null value.
      (description description)
      (cite cite ...) ; Has default value.
      (output output ...)
      (activate_output activate_output) ; Default true value.
      (log_time_columns log_time_columns ...) ; Has default value.
      (log_prefix "")
      (print_time print_time) ; Default periodic value.
      (manager manager)
      (previous previous)
      (next_large next_large)
      (minimal_timestep minimal_timestep) >
```

- *weather*: **weather** component (see chapter 89)
Optional component
Weather model for providing climate information during the simulation. Can be overwritten by column specific weather.
- *harvest*: **Harvest** fixed component (see section 93.4) sequence
Submodel (default: an empty sequence)
Total list of all crop yields.

- *time*: **Time** fixed component (see section 93.21)
Submodel (has partially specified default value)
Current time in the simulation.
- *timestep*: **Timestep** fixed component (see section 93.1)
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 93.21)
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 24) sequence
List of columns to use in this simulation.
- *exchange*: **scope** component (see chapter 63) sequence
Component (default: an empty sequence)
List of exchange items for communicating with external models.
- *scope*: **scopesel** component (see chapter 64)
Component (default 'null')
Scope to evaluate expressions in.
- *description*: string (see section 4.1.5)
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 4.1.5) 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 parameterization.

- *output*: **log** component (see chapter 44) sequence
List of logs for output during the simulation.
- *activate_output*: **condition** component (see chapter 26)
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 4.1.5) 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 4.1.5)
Parameter (default ‘’)
Prefix for log file names. Set it to ‘log/’ to put all files in a subdir.
- *print_time*: **condition** component (see chapter 26)
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 14)
Specify the management operations to perform during the simulation.
- *previous*: **Time** fixed component (see section 93.21)
Optional submodel
Previous time in the simulation.
- *next_large*: **Time** fixed component (see section 93.21)
Optional submodel
End of next large timestep.
- *minimal_timestep*: **Timestep** fixed component (see section 93.1)
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.

57.3 GP2D

Write root density table using 2D extension to Gerwitz and Page

```
< GP2D (DS 2 [DS])
    (row_width row_width)
    (row_position 0 [cm])
    (WRoot WRoot)
    (Geometry Geometry) ; Has partial value.
    (soil_depth soil_depth)
    (crop_depth crop_depth)
    (crop_width crop_width) >
```

- *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/m²**]
Parameter
Total root dry matter.
- *Geometry*: **GeometryRect** fixed component (see section 93.26)
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.

57.4 Osvaldo

Find the modelling error. This is done between one set of measured data and multiple sets of simulated data.

```
< Osvaldo  (pars pars)
           (par_file_prefix par_file_prefix)
           (par_file_suffix par_file_suffix)
           (par_width par_width)
           (measured_file measured_file) >
```

- *pars*: integer
Parameter
Number of simulated dataset to compare.
- *par_file_prefix*: string (see section 4.1.5)
Parameter
Beginning of file name before simulation number.
- *par_file_suffix*: string (see section 4.1.5)
Parameter
End of file name after simulation number.
- *par_width*: integer
Parameter
Number of digits in simulated file name.
- *measured_file*: string (see section 4.1.5)
Parameter
Name of file containing measurements.

57.5 batch

Run a sequence of programs.

```
< batch  (directory ".")
         (run run ...) >
```


- *directory*: string (see section 4.1.5)
Parameter (default '.')
Directory in which to initialize, check and run the programs.
- *run*: **program** component (see chapter 57) sequence
List of programs to run. The programs will be run in the sequence listed.

57.6 cd

Change working directory.

```
< cd directory >
```

- *directory*: string (see section 4.1.5)
Parameter
Name of directory to change into.

57.7 cpedata

Manipulate data from Agrovand.

```
< cpedata (value value)
          (file file)
          (factor 1 [])
          (day day)
          (missing missing ...) ; Has default value.
          (filter)
          (original original ...)
          (dim_line dim_line)
          (begin begin)
          (weight weight)
          (debug 0)
          (handle handle)
          (origin origin) ; Has partial value.
          (every_hour false) >
```

- *value*: string (see section 4.1.5)
Parameter
Tag used for value.
- *file*: string (see section 4.1.5)
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 4.1.5)
Parameter
Tag used for day.
- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.
 < *tag allowed...* >
 - *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
 - *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
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 93.21)
Optional submodel
If specified, only print entries after this date.
- *weight*: string (see section 4.1.5)
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 4.1.5)
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 93.21)
Submodel (has partially specified default value)
Day 1.
- *every_hour*: boolean (see section 4.1.2)
Parameter (default false)
If true, print zeroes for hours with no data.

57.8 docmodel

Document specific models.

```
< docmodel (component component)
            (format format)           ; Default LaTeX value.
            (where where)
            (models models ...) >
```

- *component*: string (see section 4.1.5)
Parameter
Component to find the models in.
- *format*: **format** component (see chapter 36)
Component (default 'LaTeX')
Text format used for the document.
- *where*: string (see section 4.1.5)
Parameter
Name of file to store results in.
- *models*: string (see section 4.1.5) sequence
Parameter
Models to document.

57.9 document

Generate the components part of the reference manual.

```
< document (format format)           ; Default LaTeX value.
            (where "components.tex")
            (print_parameterizations false) >
```

- *format*: **format** component (see chapter 36)
Component (default 'LaTeX')
Text format used for the document.
- *where*: string (see section 4.1.5)
Parameter (default 'components.tex')
Name of file to store results in.
- *print_parameterizations*: boolean (see section 4.1.2)
Parameter (default false)
Include a copy of all loaded parameterizations in document.

57.10 gnuplot

Generate a gnuplot command file.

```
< gnuplot (cd true)
           (command_file "daisy.gnuplot")
           (extra)
           (graph graph ...) >
```

- *cd*: boolean (see section 4.1.2)
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 4.1.5)
Parameter (default 'daisy.gnuplot')
File name for gnuplot commands.
- *extra*: string (see section 4.1.5) 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 37) sequence
Graphs to plot.

57.11 hmovie

Manipulate data from Agrovand.

```
< hmovie (file file)
        (dim dim)
        (cd true)
        (description description)
        (cite)
        (missing missing ...) ; Has default value.
        (filter)
        (original original ...)
        (dim_line dim_line)
        (missing_value missing_value)
        (command_file "daisy.gnuplot")
        (x_min x_min)
        (x_max x_max)
        (y_min y_min)
        (y_max y_max)
        (z_min z_min)
        (z_max z_max)
        (x_ticks x_ticks ...)
        (x_tic_max x_tic_max)
        (output_base output_base)
        (output_width output_width)
        (tags tags ...) >
```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *dim*: string (see section 4.1.5)
Parameter
Dimension of all axes.
- *cd*: boolean (see section 4.1.2)
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 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

< tag allowed... >

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 missing values with this.
- *command_file*: string (see section 4.1.5)
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 4.1.7) sequence
List of tic markers for the x-axes.
 - < value name >**
 - *value*: number (dimension not specified)
Parameter
Tic position.
 - *name*: string (see section 4.1.5)
Parameter
Tic name.
- *x_tic_max*: number [**<user>**]
Parameter
Highest synthetic z value for at x tics.
- *output_base*: string (see section 4.1.5)
Parameter
Prefix used for all output files.
- *output_width*: integer
Parameter
Number of digits in output file name.
- *tags*: submodel (see section 4.1.7) 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 4.1.5)
Parameter
Column name.

57.12 hydraulic

Generate a table of the retention curve and hydraulic conductivity.

```
< hydraulic horizon
    (top_soil top_soil)
    (intervals 50) >
```

- *horizon*: **horizon** component (see chapter 40)
The hydraulic model to show in the table.
- *top_soil*: boolean (see section 4.1.2)
Parameter
Set to true for the plowing layer.
- *intervals*: integer
Parameter (default 50)
Number of intervals in the table.

57.13 minimize

Find local minimum for program.

The optimization will stop if the worst guess fails to improve more than 'epsilon' within 'min_iter' iterations, or if the total number of iterations exceeds 'max_iter'.

```
< minimize  (expr expr)
              (scope scope)
              (run run)
              (parameter parameter ...)
              (limit limit)
              (simplex simplex ...)
              (epsilon epsilon)
              (min_iter 10)
              (max_iter max_iter) >
```

- *expr*: **number** component (see chapter 50)
Expression to minimize.
- *scope*: **scopesel** component (see chapter 64)
Scope to evaluate expressions in.
- *run*: **program** component (see chapter 57)
Program to optimize.
- *parameter*: string (see section 4.1.5) sequence
Parameter
List of parameters to optimize.
- *limit*: **boolean** component (see chapter 20)
Limit parameter values so this expression is true.
- *simplex*: submodel (see section 4.1.7) sequence
List of points defining the initial simplex. You must define one more point than you have parameters.

```
<  value...  >
```

 - *value*: number (dimension not specified) sequence
Parameter
Value of each parameter.
- *epsilon*: number (dimension not specified)
Parameter
Minimal improvement of worst point to be considered for 'min_iter'.
- *min_iter*: integer
Parameter (default 10)
Maximal number of iterations with no improvement of worst point.

- *max_iter*: integer
Parameter
Stop after this number of iterations.

57.14 rootmatch

Match root data with GP2D model.

```
< rootmatch (file file)
    (gnuplot gnuplot)           ; Has default value.
    (description description)
    (cite)
    (row_position 0 [cm])
    (missing missing ...)       ; Has default value.
    (filter)
    (original original ...)
    (dim_line dim_line)
    (debug 0)
    (SpRtLength 100 [m/g])
    (DensRtTip 0.1 [cm/cm3])
    (row_distance row_distance)
    (pos_dim pos_dim)
    (dens_dim dens_dim)
    (tag_x tag_x)
    (tag_z_min tag_z_min)
    (tag_z_max tag_z_max)
    (tag_density tag_density)
    (show_data true)
    (show_match false)
    (x_offset 0 [cm])
    (tabular false)
    (min_dist -1 [cm])
    (SoilDepth SoilDepth) >
```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *gnuplot*: submodel (see section 4.1.7)
Submodel (has fully specified default value)

(gnuplot)

Parameter description:

Irrigation model for first season. If missing, don't irrigate.

```
< (x_offset 0 [cm])
    (x_start 0 [cm])
    (x_end x_end)
    (z_end z_end)
    (dx 1 [cm])
    (dz dz) >
```

- *x_offset*: number [cm]
Optional parameter (default 0)
Subtract this from printed x values.

- *x_start*: number [**cm**]
Parameter (default 0)
Start table here.
- *x_end*: number [**cm**]
Optional parameter
End table here. By default, this is ‘x_start’ plus ‘row_distance’.
- *z_end*: number [**cm**]
Optional parameter
End table here at this depth.
- *dx*: number [**cm**]
Parameter (default 1)
Horizontal interval size.
- *dz*: number [**cm**]
Optional parameter
Vertical interval size. By default identical to ‘dx’.
- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *row_position*: number [**cm**]
State variable (default 0)
Horizontal position of row crops.
- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:
List of strings indicating missing values.
- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.
 - < *tag allowed...* >
 - *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
 - *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
Optional parameter
If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.
- *debug*: integer
Parameter (default 0)
Show debug messages if larger than zero.
- *SpRtLength*: number [**m/g**]
Parameter (default 100)
Specific root length
- *DensRtTip*: number [**cm/cm³**]
Parameter (default 0.1)
Root density at (potential) penetration depth.
- *row_distance*: number [**cm**]
State variable
Distance between rows of crops.
- *pos_dim*: string (see section 4.1.5)
Parameter
Position dimension
- *dens_dim*: string (see section 4.1.5)
Parameter
Root density dimension.
- *tag_x*: string (see section 4.1.5)
Parameter
Name of column indicating horizontal distance from row.
- *tag_z_min*: string (see section 4.1.5)
Parameter
Name of column indicating minimal vertical distance from row.
- *tag_z_max*: string (see section 4.1.5)
Parameter
Name of column indicating maximal vertical distance from row.
- *tag_density*: string (see section 4.1.5)
Parameter
Name of column containing measured root density.
- *show_data*: boolean (see section 4.1.2)
Parameter (default true)
Show comparison of observed and modelled values.
- *show_match*: boolean (see section 4.1.2)
Parameter (default false)
Show matching observed and modelled values with errorbars.
- *x_offset*: number [**cm**]
State variable (default 0)
Subtract this from the x values shown with 'show_match'.

- *tabular*: boolean (see section 4.1.2)
Parameter (default false)
Show parameters in tabular form for easy import to spreadsheet.
- *min_dist*: number [**cm**]
State variable (default -1)
Ignore root data closer than this to the row in both dimensions.
- *SoilDepth*: number [**cm**]
Optional parameter
Specifies how to handle soil imposed maximum root depth. If positive, always use this depth. If negative, disable maximum root depth (making the root zone infinite). If not specified, use the value that gives the best fit.

Note that unless you specify a negative value, the root zone will be limited by DensRtTip, that is, depth where the corresponding 1D GP would give densities less than DensRtTip, will instead have a zero density.

57.15 write

Write string to file.

```
< write (where screen)
      (what what) >
```

- *where*: string (see section 4.1.5)
Parameter (default 'screen')
File to write it in. If the value is 'screen', write the string to the screen.
- *what*: string (see section 4.1.5)
Parameter
String to write.

57.16 wse

Generate a table of the water stress effect.

```
< wse use
      (intervals 10) >
```

- *use*: **wse** component (see chapter 90)
The water stress effect to show in the table.
- *intervals*: integer
Parameter (default 10)
Number of intervals in the table.

Chapter 58

raddist

The 'raddist' component calculates the radiation distribution in the canopy.

58.1 default

Default model of radiation distribution in the canopy.

Used by bioclimate default raddist (see 18.1, page 107) .

58.2 sun-shade

Sun-shade model of radiation distribution in the canopy.

```
< "sun-shade" (sigma_PAR 0.15 [])  
              (sigma_NIR 0.83 [])  
              (Ps_PAR 0.1 [])  
              (Ps_NIR 0.18 []) >
```

- *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 coefficient of beam PAR for horizontal leaves
- *Pcb_PAR*: number []
Canopy reflection coefficient of beam PAR for uniform leaf-angle distribution
- *Pscb_PAR*: number []
Canopy-soil reflection coefficient of beam PAR for uniform leaf-angle distribution
- *Pscd_PAR*: number []
Canopy-soil reflection coefficient of diffuse PAR for uniform leaf-angle distribution
- *Ph_NIR*: number []
Canopy reflection coefficient of beam NIR for horizontal leaves
- *Pcb_NIR*: number []
Canopy reflection coefficient of beam NIR for uniform leaf-angle distribution
- *Pscb_NIR*: number []
Canopy-soil reflection coefficient of beam NIR for uniform leaf-angle distribution
- *Pscd_NIR*: number []
Canopy-soil reflection coefficient of diffuse NIR for uniform leaf-angle distribution

Chapter 59

rainergy

Energy in rain.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

59.1 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 60.4, page 323) .

59.2 EUROSEM

Kinetic energy model taking vegetation into account. See also [Morgan et al., 1998]

Used by reaction colgen_Morgan98 rainergy (see 60.5, page 324) .

Chapter 60

reaction

Generic transformations between soil chemicals.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

60.1 adsorption

Maintain equilibrium between solute and sorbed form.

```
< adsorption (solute solute)  
            (equilibrium equilibrium)  
            (sorbed sorbed)  
            (adsorption_rate adsorption_rate)  
            (desorption_rate desorption_rate)  
            ;; Shared parameters are described in chapter 60.  
            (description description)  
            (cite) >
```

- *solute*: string (see section 4.1.5)
Parameter
Name of solute form of chemical.
- *equilibrium*: **adsorption** component (see chapter 15)
Function for calculating equilibrium between solute and sorbed form.
- *sorbed*: string (see section 4.1.5)
Parameter
Name of sorbed form of chemical.
- *adsorption_rate*: **number** component (see chapter 50)
Transformation rate from solute to sorbed form.
- *desorption_rate*: **number** component (see chapter 50)
Optional component

Transformation rate from sorbed to solute form. By default, this is identical to 'adsorption_rate'.

Log Variables

- *adsorption_source*: number [g/cm³/h] soil cells
Converted from solute to sorbed form this timestep (may be negative).

60.2 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_release (bound bound)
                (colloid colloid)
                (immobile immobile)
                ;; Shared parameters are described in chapter 60.
                (description description)
                (cite) >
```

- *bound*: string (see section 4.1.5)
Optional parameter
Chemical bound to colloids. If unspecified, the colloid bound form will not be traced.
- *colloid*: string (see section 4.1.5)
Optional parameter (default 'colloid')
Name of colloid whose release we mimic.
- *immobile*: string (see section 4.1.5)
Parameter
Immobile (or mixed form) chemical in the soil surface.

Log Variables

- *release*: number [g/cm²/h]
Release rate of immobile chemical as colloids.

60.3 colgen

Shared parameter and log variable for colloid generation models.

```
< colgen (ponddamp ponddamp)
         (colloid colloid)
         ;; Shared parameters are described in chapter 60.
         (description description)
         (cite) >
```

- *ponddamp*: **ponddamp** component (see chapter 56)
Model for calculating 'KH'.

- *colloid*: string (see section 4.1.5)
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 [**g/cm²/h**]
Depletion of detachable particles from top soil.

60.4 colgen_Jarvis99

A ‘colgen’ model (see 60.3, page 322) build into Daisy.

Colloid generation emulating the MACRO model. See also [Jarvis et al., 1999, Brubaker et al., 1992]

```
< colgen_Jarvis99  (ponddamp ponddamp)                ; Default none value.
                  (rainergy rainergy)                ; Default Brown87 value.
                  (cite cite ...)                    ; Has default value.
                  (Ms Ms)
                  (tillage_replenish_all false)
                  (Mmax Mmax)
                  (Mmax_tillage_factor Mmax_tillage_factor) ; Has default value.
                  (kd kd)
                  (kr kr)
                  (zi zi)
                  ;; Shared parameters are described in section 60.3.
                  (description description)
                  (colloid colloid) >
```

- *rainergy*: **rainergy** component (see chapter 59)
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 4.1.2)
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 function of days after tillage.

- *kd*: number [g/J]
Parameter
Detachment rate coefficient.
- *kr*: number [g/cm²/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/cm²]
Current amount of detachable particles in top soil.
- *P*: number [g/cm²/h]
Replenishment of detachable particles to top soil.
- *KE*: number [J/cm²/h]
Kinetic energy available for colloid generation.
- *E*: number [J/cm²/mm]
Kinetic energy in rain.

60.5 colgen_Morgan98

A 'colgen' model (see 60.3, page 322) build into Daisy.

Colloid generation using kinetic energy, emulating EUROSEM. See also [Morgan et al., 1998]

```
< colgen_Morgan98 (rainergy rainergy) ; Default EUROSEM value.
                  (cite cite ...) ; Has default value.
                  (kd kd)
                  ;; Shared parameters are described in section 60.3.
                  (ponddamp ponddamp)
                  (description description)
                  (colloid colloid) >
```

- *rainergy*: **rainergy** component (see chapter 59)
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 [**J/cm²/h**]
Kinertic energy available for colloid generation.
- *E*: number [**J/cm²/mm**]
Kinetic energy in rain.

60.6 colgen_Styczen88

A ‘colgen’ model (see 60.3, page 322) build into Daisy.

Colloid generation using rainfall momentum. See also [Styczen and Høeg-Schmidt, 1988]

```
< colgen_Styczen88 (cite cite ...) ; Has default value.
                    (Ae Ae)
                    (MA MA)
                    (droplet_diameter droplet_diameter)
                    ;; Shared parameters are described in section 60.3.
                    (ponddamp ponddamp)
                    (description description)
                    (colloid colloid) >
```

- *Ae*: number [**h²/g/cm²**]
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 [**kg²/m/s²**]
Squared vegetation droplet momentum.
- *CM*: number [**<fraction>**]
Vegetation factor.
- *MR*: number [**(N s)²/m²/s**]
Squared direct rainfall momentum.

60.7 default

Transformation between two soil chemicals.

```
< default (transform transform)
          (A A)
          (B B)
          ;; Shared parameters are described in chapter 60.
          (description description)
          (cite) >
```

- *transform*: **transform** component (see chapter 77)
Tranformation process between 'A' to 'B'.

- *A*: string (see section 4.1.5)
Parameter
Name of first soil component in equilibrium.
- *B*: string (see section 4.1.5)
Parameter
Name of second soil component in equilibrium.

Log Variables

- *S_{AB}*: number [g/cm³/h] soil cells
Converted from A to B this timestep (may be negative).

60.8 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 23.2, page 133) .

```
< denitrification (K 0.020833 [h-1])
                  (alpha 0.1 [(g NO3-N/h)/(g CO2-C/h)])
                  (heat_factor heat_factor)
                  (water_factor water_factor) ; Has default value.
                  (K_fast K_fast)
                  (alpha_fast alpha_fast)
                  (water_factor_fast water_factor_fast)
                  (redox_height redox_height)
                  ;; Shared parameters are described in chapter 60.
                  (description description)
                  (cite) >
```

- *K*: number [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 [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 $[(\text{g NO}_3\text{-N/h})/(\text{g CO}_2\text{-C/h})]$
Optional parameter
Anaerobic denitrification constant for fast pools. This applies to the CO_2 produced from turnover of fast OM pools. By default, this is identical to alpha.
- *water_factor_fast*: plf [**<fraction>** \rightarrow **<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) below which redox processes start. All NO_3 below this height will be denitrified immediately. By default no redox denitrification occurs.

Log Variables

- *potential*: number [$\text{g/cm}^3/\text{h}$] soil cells
Potential amount of denitrification at anaerobic conditions.
- *converted*: number [$\text{g/cm}^3/\text{h}$] sequence
Amount of denitrification.
- *converted_fast*: number [$\text{g/cm}^3/\text{h}$] soil cells
Additional denitrification due to turnover in fast pools.
- *converted_redox*: number [$\text{g/cm}^3/\text{h}$] soil cells
Additional denitrification due to chemical redox processes.
- *potential_fast*: number [$\text{g/cm}^3/\text{h}$] soil cells
Additional potential due to turnover in fast pools.

60.9 equilibrium

Equilibrium between two soil chemicals.

```
< equilibrium (equilibrium equilibrium)
    (A A)
    (secondary false)
    (primary primary)
    (surface false)
    (B B)
    (k_AB k_AB)
    (k_BA k_BA)
    (colloid colloid)
    ;; Shared parameters are described in chapter 60.
    (description description)
    (cite) >
```

- *equilibrium*: **equilibrium** component (see chapter 34)
Function for calculating equilibrium between A and B.

- *A*: string (see section 4.1.5)
Parameter
Name of first soil component in equilibrium.
- *secondary*: boolean (see section 4.1.2)
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 4.1.2)
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 4.1.2)
Parameter (default false)
Equilibrium should happen in the surface.
- *B*: string (see section 4.1.5)
Parameter
Name of second soil component in equilibrium.
- *k_{AB}*: **number** component (see chapter 50)
Transformation rate from soil component 'A' to 'B'.
- *k_{BA}*: **number** component (see chapter 50)
Optional component
Transformation rate from soil component 'B' to 'A'. By default, this is identical to 'k_{AB}'.
- *colloid*: string (see section 4.1.5)
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 [g/cm³/h] soil cells
Converted from A to B in soil this timestep (may be negative).
- *surface_AB*: number [g/cm²/h]
Converted from A to B on surface this timestep (may be negative).

60.10 filter_velocity

Filtration of soil colloids.


```

< filter_velocity  (immobile immobile)
                   (mobile mobile)
                   (fc_primary fc_primary)
                   (fc_secondary fc_secondary)
                   ;; Shared parameters are described in chapter 60.
                   (description description)
                   (cite) >

```

- *immobile*: string (see section 4.1.5)
Optional parameter
Immobile colloids in the soil. By default, filtered colloids are not tracked.
- *mobile*: string (see section 4.1.5)
Parameter
Mobile colloids dissolved in soil water.
- *fc_primary*: number [cm^{-1}]
Parameter
Filter coefficient in the primary domain
- *fc_secondary*: number [cm^{-1}]
Parameter
Filter coefficient in secondary domain

Log Variables

- *F_primary*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Filtration in the primary domain (intra-aggregate pores).
- *F_secondary*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Filtration in secondary domain (inter-aggregate pores).

60.11 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 23.2, page 133) .

```

< nitrification  ;; Shared parameters are described in chapter 60.
                 (description description)
                 (cite) >

```

Log Variables

- *NO3*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Amount of nitrate generated this hour.
- *NH4*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Amount of ammonium consumed this hour.
- *N2O*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Amount of nitrous oxide generated this hour.

60.12 sorption

Kinetic linear sorption equilibrium. Faster than the 'equilibrium' reaction model, more flexible than the 'adsorption' reaction model.

```
< sorption (solute solute)
           (K_clay K_clay)
           (K_OC K_OC)
           (sorbed sorbed)
           (colloid colloid)
           (K_d K_d)
           (k_sorption k_sorption)
           (k_desorption k_desorption)
           (soil_enrichment_factor 1 [])
           ;; Shared parameters are described in chapter 60.
           (description description)
           (cite) >
```

- *solute*: string (see section 4.1.5)
Parameter
Name of solute form of chemical.
- *K_clay*: number [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 [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 4.1.5)
Parameter
Name of sorbed form of chemical.
- *colloid*: string (see section 4.1.5)
Optional parameter
Sorp to this chemical instead of to the soil matrix.
- *K_d*: number [cm^3/g]
Optional parameter
Equilibrium parameter: $M = C (K_d \rho_b + \Theta)$ Here M is the total amount, C is solute concentration, K_d is this parameter, ρ_b is the dry bulk density, and Θ is the volumetric water content. By default, K_d is calculated from K_clay and K_OC.
- *k_sorption*: number [h^{-1}]
Parameter
Sorption rate.
- *k_desorption*: number [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 [$\text{g}/\text{cm}^2/\text{h}$]
Sorption on surface this timestep (may be negative).
- *S_sorption*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Sorption in soil this timestep (may be negative).
- *S_sorption_primary*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Sorption in primary domain this timestep (may be negative).
- *S_sorption_secondary*: number [$\text{g}/\text{cm}^3/\text{h}$] soil cells
Sorption in secondary domain this timestep (may be negative).

Chapter 61

rootdens

Root density calculations.

Used by RootSystem @ rootdens (see 93.13, page 494) .

```
< component (description description)  
            (cite)  
            (SpRtLength 100 [m/g]) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *SpRtLength*: number [m/g]
Parameter (default 100)
Specific root length

61.1 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]

```
< "Anders Pedersen" (cite cite ...) ; Has default value.  
                    (q q)  
                    (a_DS a_DS)  
                    ;; Shared parameters are described in chapter 61.  
                    (description description)  
                    (SpRtLength 100 [m/g]) >
```

- *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 \rightarrow cm⁻¹]
Parameter
Form parameter as a function of development stage.

Log Variables

- *a*: number [cm^{-1}]
Form parameter. Calculated from 'a_DS'.
- *L0*: number [cm/cm^3]
Root density at soil surface.

61.2 DS_Depth

Specify root density as a function of development stage.

```
< DS_Depth (entries entries ...)
;; Shared parameters are described in chapter 61.
(description description)
(cite)
(SpRtLength 100 [m/g]) >
```

- *entries*: submodel (see section 4.1.7) 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 .

61.3 DS_Rel

Specify root density as a function of development stage.

```
< DS_Rel (entries entries ...)
;; Shared parameters are described in chapter 61.
(description description)
(cite)
(SpRtLength 100 [m/g]) >
```

- *entries*: submodel (see section 4.1.7) 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 .

61.4 Depth_Depth

Specify root density as a function of development stage.

```
< Depth_Depth (entries entries ...)
                ;; Shared parameters are described in chapter 61.
                (description description)
                (cite)
                (SpRtLength 100 [m/g]) >
```

- *entries*: submodel (see section 4.1.7) 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 .

61.5 GP1D

Use exponential function for root density. See also [Gerwitz and Page, 1974]

```
< GP1D (cite cite ...) ; Has default value.
        (DensRtTip 0.1 [cm/cm3])
        (DensIgnore DensIgnore)
        ;; Shared parameters are described in chapter 61.
        (description description)
        (SpRtLength 100 [m/g]) >
```

- *DensRtTip*: number [cm/cm³]
Parameter (default 0.1)
Root density at (potential) penetration depth.
- *DensIgnore*: number [cm/cm³]
Optional parameter
Ignore cells with less than this root density. By default, this is the same as DensRtTip.

Log Variables

- *a*: number [**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.

61.6 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]

```
< GP2D (cite cite ...) ; Has default value.
      (row_position 0 [cm])
      (debug 0)
      (DensRtTip 0.1 [cm/cm3])
      (DensIgnore DensIgnore)
      (row_distance row_distance)
      ;; Shared parameters are described in chapter 61.
      (description description)
      (SpRtLength 100 [m/g]) >
```

- *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.

61.7 Gerwitz+Page74

Use exponential function for root density. See also [Gerwitz and Page, 1974]

```
< "Gerwitz+Page74" (cite cite ...) ; Has default value.
    (DensRtTip 0.1 [cm/cm3])
    (MinDens 0 [cm/cm3])
    ;; Shared parameters are described in chapter 61.
    (description description)
    (SpRtLength 100 [m/g]) >
```

- $DensRtTip$: number [cm/cm^3]
Parameter (default 0.1)
Root density at (potential) penetration depth.
- $MinDens$: number [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 [cm^{-1}]
Form parameter. Calculated from 'DensRtTip'.
- $L0$: number [cm/cm^3]
Root density at soil surface.

61.8 growth

Dynamic root growth model.

```

< growth (row_position 0 [cm])
          (Depth Depth)
          (WRoot WRoot)
          (DensRtTip 0.1 [cm/cm3])
          (row_distance row_distance)
          (Width Width)
          ;; Shared parameters are described in chapter 61.
          (description description)
          (cite)
          (SpRtLength 100 [m/g]) >

```

- *row_position*: number [**cm**]
State variable (default 0)
Horizontal position of row crops.
- *Depth*: number [**cm**]
Optional state variable
Expected depth of root zone (positive).
- *WRoot*: number [**g DM/m²**]
Optional state variable
Total root mass.
- *DensRtTip*: number [**cm/cm³**]
Parameter (default 0.1)
Root density at (potential) penetration depth.
- *row_distance*: number [**cm**]
Optional state variable
Distance between rows of crops.
- *Width*: number [**cm**]
Optional state variable
Expected width of root zone.

Chapter 62

rubiscoNdist

The 'rubiscoNdist' component calculates the rubisco N distribution for photosynthesis in the canopy.

62.1 exp

Boegh et al.(2002) rubisco N-distribution model in the canopy for photosynthesis.

Used by photosynthesis Farquhar N-dist (see 55.1, page 291) .

```
< exp (kn 0.713 [])  
      (f_photo 1 []) >
```

- *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 subtracted from leaf-N in the cropN_std module, therefore $f_{photo} = 1.0$ as default.

62.2 expr

expr rubisco N-distribution model in the canopy.

```
< expr value  
      (f_photo 1 []) >
```

- *value*: **number** component (see chapter 50)
Expression that evaluates to the relative rubisco N intensity 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 subtracted from leaf-N in the cropN_std module, therefore $f_{photo} = 1.0$ as default.

62.3 forced

Forced rubisco capacity distribution model in the canopy.

`< forced value >`

- *value*: **number** component (see chapter 50)
Expression that evaluates to the relative rubisco capacity where 1 is the value in top of the canopy.

62.4 uniform

Uniform rubisco N-distribution model in the canopy for photosynthesis.

`< uniform (f_photo 1 []) >`

- *f_photo*: number (dimensionless)
Parameter (default 1)
Fraction of photosynthetically active N in canopy, $f_{\text{photo}} = 0.75$ (Boegh et al., 2002). However, non-functional N is already subtracted from leaf-N in the cropN_std module, therefore $f_{\text{photo}} = 1.0$ as default.

Chapter 63

scope

A scope maps names to values.

63.1 exchange

Exchange values with an external model.

```
< exchange (description description)  
           (cite)  
           (entries entries ...) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *entries*: **exchange** component (see chapter 35) sequence
List of items to exchange.

Chapter 64

scopesel

A method to choose a scope in a Daisy simulation.

64.1 multi

Combine multiple named scopes.

The entries in the combined scope will have the form `<scope>.<key>`, where `<scope>` is the name of the scope containing the entry, and `<key>` is the name of the entry in `<scope>`.

```
< multi  name...  >
```

- *name*: string (see section 4.1.5) sequence
Parameter
Names of scope to select.

64.2 name

Select named scope.

```
< name  name  >
```

- *name*: string (see section 4.1.5)
Parameter
Name of scope to select.

64.3 null

Select the empty scope.

Used by action `extern_fertigation` scope (see 14.7, page 71) , column default scope (see 24.1, page 138) , and program Daisy scope (see 57.2, page 301) .

Chapter 65

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 40, page 199) .

65.1 alpha

Shared base class for non-empty secondary domains.

```
< alpha (alpha alpha) >
```

- *alpha*: number [h^{-1}]
Parameter
Exchange rate between primary and secondary water.

65.2 cracks

A ‘alpha’ model (see 65.1, page 345) build into Daisy.

Secondary domain specified by aperture and density of soil cracks.

```
< cracks (density density)  
          (aperture aperture)  
          (use_secondary true)  
          ;; Shared parameters are described in section 65.1.  
          (alpha alpha) >
```

- *density*: number [m^{-1}]
Parameter
Density of cracks.
- *aperture*: number [m]
Parameter
Average distance between walls in cracks.

- *use_secondary*: boolean (see section 4.1.2)
Parameter (default true)
Divide soil matrix into two domains for solute transport. Set this to false to make cracks affect only the conductivity curve.

65.3 pressure

A ‘alpha’ model (see 65.1, page 345) build into Daisy.

Horizon has secondary domain specified by pressure threshold.

The secondary domain consist of water in matrix pores larger than what corresponds to the specified pressure.

```
< pressure (K 0 [cm/h])
           (h_lim h_lim)
           ;; Shared parameters are described in section 65.1.
           (alpha alpha) >
```

- *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.

65.4 none

No secondary domain.

There is always full equilibrium between solute in different size matrix pores.

Used by horizon component `secondary_domain` (see 40, page 199) .

Chapter 66

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.

66.1 LAI

Initial crop growth is governed by a forced LAI function.

Used by crop default Seed (see 27.1, page 153) , crop Maize Seed (see 27.15, page 156) , crop Pioneer Maize Seed (see 27.22, page 157) , crop Ikuwala Maize Seed (see 27.16, page 156) , crop Potato; Koege Seed (see 27.23, page 157) , crop Potato; FertOrgaNic Seed (see 27.25, page 157) , crop Potato; Agria Seed (see 27.26, page 157) , crop Potato; Folva Seed (see 27.27, page 157) , crop Potato; Triada Seed (see 27.29, page 158) , crop Sugar Beet Seed (see 27.40, page 159) , crop Fodder Beet Seed (see 27.10, page 156) , crop Winter Rape Seed (see 27.52, page 160) , crop Grass to grain Seed (see 27.13, page 156) , crop Ryegrass Seed (see 27.34, page 158) , crop Wclover Seed (see 27.45, page 159) , crop Beetroot Seed (see 27.3, page 155) , crop Broccoli Seed (see 27.4, page 155) , crop Brussels sprouts Seed (see 27.6, page 155) , crop Celeriac Seed (see 27.8, page 155) , crop White cabbage Seed (see 27.46, page 160) , crop Early white cabbage - transplanted Seed (see 27.48, page 160) , crop Onion Seed (see 27.18, page 156) , crop Vaarbyg Seed (see 27.41, page 159) , crop Froegraes Seed (see 27.11, page 156) , crop Silomajs Seed (see 27.35, page 158) , and crop Vinterraps Seed (see 27.44, page 159) .

```
< LAI (DSLAI05 0.15 [])  
      (SpLAIfac SpLAIfac) ; Has default value.  
      (InitCAI true) >
```

- *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 4.1.2)
State variable (default true)
Initial CAI development phase.

66.2 release

Initial crop growth is governed by carbon released from seeds.

```
< release  (C C)
            (initial_weight initial_weight)
            (DM_fraction DM_fraction)
            (C_fraction C_fraction)
            (N_fraction N_fraction)
            (rate rate) >
```

- *C*: number [g C/m²]
Optional state variable
Unreleased carbon left in seeds.
- *initial_weight*: number [g w.w./m²]
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 67

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 interesting_content)
              (negate false)
              (accumulate false) >
```

- *multi*: string (see section 4.1.5)
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 4.1.2)
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 50)
Optional component
Expression for finding 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 4.1.5)
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 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *when*: **condition** component (see chapter 26)
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 4.1.5)
Optional parameter
Documentation for this entry.
- *tag*: string (see section 4.1.5)
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 4.1.5) 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 4.1.7)

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 4.1.5)

Parameter

Name of library where the attribute belong. Use `'fixed'` to denote a fixed component.

- *model*: string (see section 4.1.5)

Parameter

Name of model or fixed component where the attribute belongs.

- *submodels_and_attribute*: string (see section 4.1.5) sequence

Parameter

Name of submodels and attribute.

- *handle*: string (see section 4.1.5)

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 4.1.2)

Optional parameter

True if the content of this column is interesting enough to warrant an initial line in the log file. By default, this is true iff `'handle'` is `'current'`.

- *negate*: boolean (see section 4.1.2)

Parameter (default false)

Switch sign of value. I.e. upward fluxes become downward fluxes.

- *accumulate*: boolean (see section 4.1.2)

Parameter (default false)

Log accumulated values.

67.1 array

Log all members of an array.

67.2 value

Log a single numeric value.

67.3 content

A ‘value’ model (see 67.2, page 352) 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.

```
< content (x x)
          (height height)
          (z z)
          (y y)
          ;; Shared parameters are described in section 67.2.
          (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 interesting_content)
          (negate false)
          (accumulate false) >
```

- *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.

67.4 flow

A ‘value’ base model (see 67.2, page 352) build into Daisy.

Common base for logging flow through a specific plane.

```
< flow (volume volume) ; Default box value.
      (density false)
      ;; Shared parameters are described in section 67.2.
      (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 interesting_content)
      (negate false)
      (accumulate false) >
```

- *volume*: **volume** component (see chapter 88)
Component (default ‘box’)
Soil volume to log flow into.
- *density*: boolean (see section 4.1.2)
Parameter (default false)
If true, divide value with volume height.

67.5 flow_back

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from back of specified volume.

67.6 flow_bottom

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from bottom of specified volume.

```

< flow_bottom (to to)
                ;; Shared parameters are described in section 67.4.
                (multi sum)
                (volume volume)                                ; Default box value.
                (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 interesting_content)
                (negate false)
                (accumulate false)
                (density false) >

```

- *to*: number [cm]
Optional parameter
Specify height (negative) to measure interval. By default, measure to the bottom. OBSOLETE: Use (volume box (bottom TO)) instead.

67.7 flux_bottom

A ‘flow_bottom’ model (see 67.6, page 353) build into Daisy.

Flux entering bottom of specified volume. OBSOLETE: Use ‘(flow_bottom (density true))’ instead.

Used by log Soil water entries (see 44.10, page 244) , log Field water entries (see 44.8, page 240) , log Field nitrogen entries (see 44.7, page 238) , log Soil nitrogen entries (see 44.9, page 241) , log Field chemical entries (see 44.13, page 247) , and log Soil chemical entries (see 44.14, page 249) .

67.8 flow_front

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from front of specified volume.

67.9 flow_left

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from left of specified volume.

67.10 flow_right

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from right of specified volume.

67.11 flow_top

A ‘flow’ model (see 67.4, page 353) build into Daisy.

Extract flow from top of specified volume.

```
< flow_top (from from)
;; Shared parameters are described in section 67.4.
(multi sum)
(volume volume) ; Default box value.
(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 interesting_content)
(negate false)
(accumulate false)
(density false) >
```

- *from*: number [cm]
Optional parameter
Specify height (negative) to measure from. By default, measure from the top.
OBSOLETE: Use (volume box (top FROM)) instead.

67.12 flux_top

A ‘flow_top’ model (see 67.11, page 355) 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 44.10, page 244) , log Soil nitrogen entries (see 44.9, page 241) , and log Soil chemical entries (see 44.14, page 249) .

67.13 index

A ‘value’ model (see 67.2, page 352) build into Daisy.

Extract content at specified array index.

```

< index (index index)
    ;; Shared parameters are described in section 67.2.
    (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 interesting_content)
    (negate false)
    (accumulate false) >

```

- *index*: integer
Parameter
Specify array index to select.

67.14 number

A ‘value’ model (see 67.2, page 352) 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 44.8, page 240) , log Field nitrogen entries (see 44.7, page 238) , and log Field chemical entries (see 44.13, page 247) .

67.15 volume_base

A ‘value’ model (see 67.2, page 352) build into Daisy.

Shared parameters for volume based logs.

```

< volume_base (volume volume) ; Default box value.
  (density false)
  (density_z density_z)
  (density_x density_x)
  (density_y density_y)
  (min_root_density -1 [cm/cm3])
  (min_root_crop "*")
  ;; Shared parameters are described in section 67.2.
  (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 interesting_content)
  (negate false)
  (accumulate false) >

```

- *volume*: **volume** component (see chapter 88)
Component (default 'box')
Soil volume to log.
- *density*: boolean (see section 4.1.2)
Parameter (default false)
If true, divide total content with volume. Otherwise, obey 'density_z', 'density_x', and 'density_y'.
- *density_z*: boolean (see section 4.1.2)
Parameter
If true, divide total content with volume height. This parameter is ignored if 'density' is true.
- *density_x*: boolean (see section 4.1.2)
Parameter
If true, divide total content with volume width. This parameter is ignored if 'density' is true.
- *density_y*: boolean (see section 4.1.2)
Parameter
If true, divide total content with volume depth. This parameter is ignored if 'density' is true.
- *min_root_density*: number [cm/cm³]
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 4.1.5)
Parameter (default “*”)
Name of crop whose roots should be used for the root density requirements.
Set this to “*” to use all roots.

67.16 interval

A ‘volume_base’ model (see 67.15, page 356) build into Daisy.

Summarize specified interval. This is similar to ‘volume’, except for the default values of ‘density_x’ and ‘density_y’, and the unique ‘from’ and ‘to’ parameters.

Used by log Soil water entries (see 44.10, page 244) , log Field water entries (see 44.8, page 240) , log Field nitrogen entries (see 44.7, page 238) , log Soil nitrogen entries (see 44.9, page 241) , log Field chemical entries (see 44.13, page 247) , and log Soil chemical entries (see 44.14, page 249) .

```
< interval (from from)
           (to to)
           (density_z false)
           (density_x true)
           (density_y true)
           ;; Shared parameters are described in section 67.15.
           (multi sum)
           (volume volume) ; Default box value.
           (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 interesting_content)
           (negate false)
           (accumulate false)
           (density false)
           (min_root_density -1 [cm/cm3])
           (min_root_crop "*") >
```

- *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.

67.17 volume

A ‘volume_base’ model (see 67.15, page 356) build into Daisy.
Summarize specified volume.

67.18 water

A ‘volume_base’ model (see 67.15, page 356) build into Daisy.
Shared parameters for water limited volumn logging.

```
< water (h h)
  (h_ice 0 [cm])
  ;; Shared parameters are described in section 67.15.
  (multi sum)
  (volume volume) ; Default box value.
  (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 interesting_content)
  (negate false)
  (accumulate false)
  (density false)
  (density_z density_z)
  (density_x density_x)
  (density_y density_y)
  (min_root_density -1 [cm/cm3])
  (min_root_crop "*") >
```

- *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).

67.19 water_interval

A ‘water’ model (see 67.18, page 359) 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 unique 'from' and 'to' parameters.

```
< water_interval (from from)
                  (to to)
                  (density_z false)
                  (density_x true)
                  (density_y true)
                  ;; Shared parameters are described in section 67.18.
                  (multi sum)
                  (volume volume) ; Default box value.
                  (flux flux)
                  (expr expr)
                  (h h)
                  (dimension dimension)
                  (factor 1 [?])
                  (offset 0 [?])
                  (description description)
                  (cite)
                  (when when)
                  (h_ice 0 [cm])
                  (documentation documentation)
                  (tag tag)
                  (path path ...)
                  (spec spec)
                  (handle handle)
                  (interesting_content interesting_content)
                  (negate false)
                  (accumulate false)
                  (density false)
                  (min_root_density -1 [cm/cm3])
                  (min_root_crop "*") >
```

- *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.

67.20 water_volume

A 'water' model (see 67.18, page 359) build into Daisy.

Summarize water content in the specified volume.

Chapter 68

solute

Water composition.

Chapter 69

solver

A way to solve the matrix equation ' $A x = b$ '.

69.1 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 74.2, page 391) , uzrect Mollerup solver (see 85.1, page 435) , transport Mollerup solver (see 78.2, page 403) , and heatrect Mollerup solver (see 39.1, page 197) .

69.2 none

Don't solve the equation.

69.3 ublas

Solve equation using UBLAS lu functions.

Chapter 70

source

Time series, with possible error bars and formatting information.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

70.1 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.

```
< arithmetic (timestep h)
            (file file)
            (expr expr)
            (missing missing ...) ; Has default value.
            (filter)
            (original original ...)
            (dim_line dim_line)
            (handle normal)
            (accumulate false)
            (title title)
            (with with)
            (style style)
            (default_hour 8)
            (time_offset time_offset) ; Has default value.
            (valid valid) ; Default true value.
            ;; Shared parameters are described in chapter 70.
            (description description)
            (cite) >
```

- *timestep*: string (see section 4.1.5)
Parameter (default 'h')
Multiple with this dimension when accumulating.

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *expr*: **number** component (see chapter 50)
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 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:
List of strings indicating missing values.
- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.
 - < tag allowed... >
 - *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
 - *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
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 4.1.5)
Parameter (default 'normal')
Determine how to handle multiple simultaneously. Possible values are:

sum: use the sum of the values.

normal: use the arithmetic average of the values, and calculate the standard deviation.
- *accumulate*: boolean (see section 4.1.2)
Parameter (default false)
Accumulate values.
- *title*: string (see section 4.1.5)
Optional parameter
Name of data series for the legend on the graph.

By default the name of the 'expr' object.

- *with*: string (see section 4.1.5)
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 93.1)
Submodel (has fully specified default value)
Add this to time from sources. By default, use unmodified times.
- *valid*: **boolean** component (see chapter 20)
Component (default 'true')
Ignore entries if this boolean expression is false.

70.2 column

Read a a single column from a Daisy log, weather or data file.

```
< column (timestep h)
          (file file)
          (dimension dimension)
          (factor factor)
          (offset 0 [?])
          (missing missing ...) ; Has default value.
          (filter)
          (original original ...)
          (dim_line dim_line)
          (tag tag)
          (handle normal)
          (accumulate false)
          (title title)
          (with with)
          (style style)
          (default_hour 8)
          (time_offset time_offset) ; Has default value.
          (reset_offset false)
          ;; Shared parameters are described in chapter 70.
          (description description)
          (cite) >
```

- *timestep*: string (see section 4.1.5)
Parameter (default 'h')
Multiple with this dimension when accumulating.
- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *dimension*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (has default value with length 2)

(missing "" "00.00")

Parameter description:
List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

< tag allowed... >

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
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 4.1.5)
Parameter
Name of column in Daisy log file where data is found.
- *handle*: string (see section 4.1.5)
Parameter (default 'normal')
Determine how to handle multiple simultaneously. Possible values are:

sum: use the sum of the values.

normal: use the arithmetic average of the values, and calculate the standard deviation.
- *accumulate*: boolean (see section 4.1.2)
Parameter (default false)
Accumulate values.
- *title*: string (see section 4.1.5)
Optional parameter
Name of data series for the legend on the graph.

By default the same as 'tag'.
- *with*: string (see section 4.1.5)
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 93.1)
Submodel (has fully specified default value)
Add this to time from sources. By default, use unmodified times.
- *reset_offset*: boolean (see section 4.1.2)
Parameter (default false)
Set offset to first value read. Useful for plotting already accumulated data from a later date.

70.3 combine

Combine data from multiple sources with a single expression.

```
< combine  (source source ...)
           (expr expr)
           (accumulate false)
           (title title)
           (with with)
           (style style)
           ;; Shared parameters are described in chapter 70.
           (description description)
           (cite) >
```

- *source*: **source** component (see chapter 70) 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 50)
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 4.1.2)
Parameter (default false)
Accumulate values.
- *title*: string (see section 4.1.5)
Optional parameter
Name of data series for the legend on the graph.
By default the name of the 'expr' object.
- *with*: string (see section 4.1.5)
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'.

70.4 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.

```

< merge  (source source ...)
          (dimension dimension)
          (accumulate false)
          (title title)
          (with with)
          (style style)
          ;; Shared parameters are described in chapter 70.
          (description description)
          (cite) >

```

- *source*: **source** component (see chapter 70) sequence
List of timeseries to merge.
- *dimension*: string (see section 4.1.5)
Optional parameter
Dimension of data to plot. By default use the first source with a known dimension.
- *accumulate*: boolean (see section 4.1.2)
Parameter (default false)
Accumulate values.
- *title*: string (see section 4.1.5)
Parameter
Name of data series for the legend on the graph.
- *with*: string (see section 4.1.5)
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 71

stomatacon

The 'Stomatacon' component calculates the stomata conductance of water vapour.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

71.1 WSF

Common water stress effect parameters.

```
< WSF (beta 0 [cm3/g])  
      (ABA_min 0 [g/cm3])  
      (delta 0 [MPa-1])  
      ;; Shared parameters are described in chapter 71.  
      (description description)  
      (cite) >
```

- *beta*: number [cm³/g]
Parameter (default 0)
Effect of ABA concentration. The effect is $\exp(-\text{beta}(\text{—ABA—} - \text{ABA_min}))$, where —ABA— is the ABA concentration in the xylem.
- *ABA_min*: number [g/cm³]
Parameter (default 0)
Level of ABA with unstressed production.
- *delta*: number [MPa⁻¹]
Parameter (default 0)
Effect of crown water potential. The effect is $\exp(-\text{delta} - \text{psi.c})$, where psi.c is the crown potential.

71.2 BB_base

A 'WSF' base model (see 71.1, page 373) build into Daisy.

Common parameters for Ball&Berry derived models.

```
< BB_base (m m)
          (b b)
          ;; Shared parameters are described in section 71.1.
          (description description)
          (cite)
          (beta 0 [cm3/g])
          (ABA_min 0 [g/cm3])
          (delta 0 [MPa-1]) >
```

- *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/m²/s]
Parameter
Stomatal intercept. Ball and Berry (1982) & Wang and Leuning(1998): (0.01 mol/m²/s)

71.3 BB

A ‘BB_base’ model (see 71.2, page 373) build into Daisy.

Stomata conductance calculated by the Ball & Berry model. See also [Ball et al., 1987]

71.4 Leuning

A ‘BB_base’ model (see 71.2, page 373) build into Daisy.

Stomata conductance calculated by the Leuning model. See also [Leuning, 1995]

Used by photosynthesis Farquhar Stomatacon (see 55.1, page 291) .

```
< Leuning (cite cite ...) ; Has default value.
          (Do 1500 [[Pa]])
          ;; Shared parameters are described in section 71.2.
          (m m)
          (description description)
          (b b)
          (beta 0 [cm3/g])
          (ABA_min 0 [g/cm3])
          (delta 0 [MPa-1]) >
```

- *Do*: number [[Pa]]
Parameter (default 1500)
Empirical coefficient.

71.5 MNA

A ‘WSF’ model (see 71.1, page 373) build into Daisy.

Stomata conductance calculated by the model given by Eq. 14.

```

< MNA  (max max)
        (m m)
        (alpha alpha)
        (lambda lambda)
        (b b)
        ;; Shared parameters are described in section 71.1.
        (description description)
        (cite)
        (beta 0 [cm3/g])
        (ABA_min 0 [g/cm3])
        (delta 0 [MPa-1]) >

```

- *max*: number [mol H₂O/m² leaf/s]
Optional parameter
Maximal conductivity. By default, there is no maximum.
- *m*: number [mol H₂O/m² leaf/s]
Parameter
Conductivity factor.
- *alpha*: number (dimensionless)
Parameter
Humidity effect
- *lambda*: number [umol CO₂/m² leaf/s]
Parameter
Net photosynthesis effect
- *b*: number [mol/m²/s]
Parameter
Stomatal intercept. Ball and Berry (1982) & Wang and Leuning(1998): (0.01 mol/m²/s)

71.6 SHA12

A ‘WSF’ model (see 71.1, page 373) build into Daisy.

Stomata conductance calculated by the model given by Eq. 12. See also [Ahmadi et al., 2009]

```

< SHA12  (min min)
          (m m)
          (alpha 1 [])
          (cite cite ...) ; Has default value.
          (lambda 1 [])
          ;; Shared parameters are described in section 71.1.
          (description description)
          (beta 0 [cm3/g])
          (ABA_min 0 [g/cm3])
          (delta 0 [MPa-1]) >

```

- *min*: number [mol H₂O/m² 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 photosynthesis effect

71.7 SHA14

A ‘WSF’ model (see 71.1, page 373) build into Daisy.

Stomata conductance calculated by the model given by Eq. 14. See also [Ahmadi et al., 2009]

```
< SHA14 (max max)
      (m m)
      (alpha alpha)
      (cite cite ...) ; Has default value.
      (lambda lambda)
      ;; Shared parameters are described in section 71.1.
      (description description)
      (beta 0 [cm3/g])
      (ABA_min 0 [g/cm3])
      (delta 0 [MPa-1]) >
```

- *max*: number [mol H₂O/m² leaf/s]
Optional parameter
Maximal conductivity. By default, there is no maximum.
- *m*: number [mol H₂O/m² leaf/s]
Parameter
Conductivity factor.
- *alpha*: number (dimensionless)
Parameter
Humidity effect
- *lambda*: number [m² leaf s/umol CO₂]
Parameter
Net photosynthesis effect

Chapter 72

string

Generic representation of strings.

72.1 cond

Return the value of the first clause whose condition is true.

Used by log chemical chemfid (see 44.12, page 246) , log crop cropfid (see 44.15, page 250) , log column colfid (see 44.6, page 237) , and log biopore bioporefid (see 44.11, page 245) .

`< cond clauses... >`

- *clauses*: **StringerCondClause** fixed component (see section 93.30) sequence
List of clauses to match for.

72.2 identity

Return the specified value.

`< identity (value value) >`

- *value*: string (see section 4.1.5)
Parameter
Constant value.

72.3 number

Extract the value of a number.

`< number (number number) >`

- *number*: **number** component (see chapter 50)
Number to manipulate.

72.4 dimension

A ‘number’ model (see 72.3, page 377) build into Daisy.

Extract the dimension of a number as a string.

72.5 value

A ‘number’ model (see 72.3, page 377) build into Daisy.

Extract the value of a number as a string.

```
< value (precision precision)  
      ;; Shared parameters are described in section 72.3.  
      (number number) >
```

- *precision*: integer
Optional parameter
Number of decimals after point. By default, use a floating format.

Chapter 73

summary

Summary reports for log parameterizations.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *title*: string (see section 4.1.5)
Optional parameter
Title of this summary. By default, use the name of the parameterization.

73.1 Rsqr

Calculate coefficient of determination.

```
< Rsqr (measure measure ...)  
      (print_data false)  
      ;; Shared parameters are described in chapter 73.  
      (description description)  
      (cite)  
      (title title) >
```

- *measure*: submodel (see section 4.1.7) sequence
Measured data.

```
< (time time)  
  (data data ...) >
```
- *time*: **Time** fixed component (see section 93.21)
Optional submodel
Time of measurement. Measured data will be compared to the first simulated value at or after this time. By default, compare with the first available simulated data.
- *data*: submodel (see section 4.1.7) sequence
Data measured at this time.

```
< tag value... >
```

- * *tag*: string (see section 4.1.5)
Parameter
Name of simulated data to compare with.
- * *value*: number (dimension not specified) sequence
Parameter
Measured data.
- *print_data*: boolean (see section 4.1.2)
Parameter (default false)
Print a table with all data in the summary.

73.2 RsqrW

Calculate coefficient of determination.

```
< RsqrW (measure measure ...)
  (print_data false)
  ;; Shared parameters are described in chapter 73.
  (description description)
  (cite)
  (title title) >
```

- *measure*: submodel (see section 4.1.7) sequence
Measured data.

```
< (time time)
  (data data ...) >
```

 - *time*: **Time** fixed component (see section 93.21)
Optional submodel
Time of measurement. Measured data will be compared to the first simulated value at or after this time. By default, compare with the first available simulated data.
 - *data*: submodel (see section 4.1.7) sequence
Data measured at this time.

```
< weight tag value... >
```

 - * *weight*: number (dimensionless)
Parameter
Weight given this measurement.
 - * *tag*: string (see section 4.1.5)
Parameter
Name of simulated data to compare with.
 - * *value*: number (dimension not specified) sequence
Parameter
Measured data.
- *print_data*: boolean (see section 4.1.2)
Parameter (default false)
Print a table with all data in the summary.

73.3 balance

A summary model providing a balance for a log parameterization.

Used by log Field chemical summary (see 44.13, page 247) , and log Soil chemical summary (see 44.14, page 249) .

```

< balance (content content ...)
          (precision 2)
          (where where)
          (input input ...)
          (require_top false)
          (output output ...)
          ;; Shared parameters are described in chapter 73.
          (description description)
          (cite)
          (title title) >

```

- *content*: string (see section 4.1.5) 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 4.1.5)
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 4.1.5) sequence
Parameter
Tags of columns in log file representing inputs.
- *require_top*: boolean (see section 4.1.2)
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 4.1.5) sequence
Parameter
Tags of columns in log file representing outputs.

73.4 Field crop nitrogen balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Crop nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 44.7, page 238) .

73.5 Field mineral nitrogen balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Mineral balance for the complete system.
Used by log Field nitrogen summary (see 44.7, page 238) .

73.6 Field nitrogen balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 44.7, page 238) .

73.7 Field organic nitrogen balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Organic nitrogen balance for the complete system.
Used by log Field nitrogen summary (see 44.7, page 238) .

73.8 Field water balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Water balance for the complete system.
Used by log Field water summary (see 44.8, page 240) .

73.9 N_balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Unit is kg, precion is g.

73.10 Soil NH4+ balance

A ‘N_balance’ model (see 73.9, page 382) defined in ‘log-std.dai’.
NH4+ balance for a soil interval.
Used by log Soil nitrogen summary (see 44.9, page 241) .

73.11 Soil NO3- balance

A ‘N_balance’ model (see 73.9, page 382) defined in ‘log-std.dai’.
NO3- balance for a soil interval.
Used by log Soil nitrogen summary (see 44.9, page 241) .

73.12 Soil mineral-N balance

A ‘N_balance’ model (see 73.9, page 382) defined in ‘log-std.dai’.
Mineral N balance for a soil interval.
Used by log Soil nitrogen summary (see 44.9, page 241) .

73.13 Soil organic-N balance

A ‘N_balance’ model (see 73.9, page 382) defined in ‘log-std.dai’.
Organic nitrogen balance for a soil interval.
Used by log Soil nitrogen summary (see 44.9, page 241) .

73.14 Soil N balance

A ‘balance’ model (see 73.3, page 380) defined in ‘log-std.dai’.
Nitrogen balance for a soil interval.
Used by log Soil nitrogen summary (see 44.9, page 241) .

73.15 Soil water balance

A ‘balance’ model (see 73.3, page 380) 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 44.10, page 244) .

73.16 simple

A simple log file summary model.

```
< simple  (fetch fetch ...)
          (precision 2)
          (period period)
          (where where)
          (print_sum true)
          (sum_name Sum)
          ;; Shared parameters are described in chapter 73.
          (description description)
          (cite)
          (title title) >
```

- *fetch*: **FetchPretty** fixed component (see section 93.28) 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 4.1.5)
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 4.1.5)
Optional parameter
File name to store the summary. By default, the summary will be stored in daisy.log and the screen.
- *print_sum*: boolean (see section 4.1.2)
Parameter (default true)
Print sum of all the summary lines.
- *sum_name*: string (see section 4.1.5)
Parameter (default ‘Sum’)
Name of the sum of all the entries.

Chapter 74

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.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

74.1 PMSW

Peter van der Keur's SVAT model. See also [van der Keur et al., 2001]

```

< PMSW (cite cite ...) ; Has default value.
  (albedo 0.2 [])
  (b1 0.53 [])
  (b2 0.0065 [])
  (b3 0.1 [])
  (b4 0.9 [])
  (ndif 2.5 [])
  (c_d 0.05 [])
  (z_0s 0.01 [m])
  (z0_def 0.005 [m])
  (w 0.0025 [m])
  (alpha_u 3 [])
  (arac 0.00662 [])
  (alpha_k 2 [])
  (alpha_r 0.5 [])
  (theta_w 0.05 [cm3/cm3])
  (theta_c 0.25 [cm3/cm3])
  (rcmin_const 30 [s/m])
  (rcmax 1000 [])
  (tref 298 [])
  (zeta 0.0002 [])
  (f3const 0.0016 [])
  (spar 100 [])
  (tmin 0 [dg C])
  (tmax 55 [dg C])
  (nu_1 26.5 [])
  (nu_2 0.57 [])
  (nu_3 0.008 [])
  (dt1 -5 [dg C])
  (dt2 5 [dg C])
  (acc 0.01 [])
  ;; Shared parameters are described in chapter 74.
  (description description) >

```

- *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)
- *z_0def*: 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 [**cm**³/**cm**³]
Parameter (default 0.05)
Soil water content at 'wilting point'
- *theta_c*: number [**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 [**W/m²**]
Sensible heat flux from source- to screen height
- *netrad_brunt*: number [**W/m²**]
Net radiation by Brunt
- *netlong_brunt*: number [**W/m²**]
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 [**W/m²**]
Sensible heat flux from leaf to mean source
- *hs*: number [**W/m²**]
Sensible heat flux from soil to mean source
- *lea*: number [**W/m²**]
Latent heat flux from source- to screen height
- *lel*: number [**W/m²**]
Latent heat flux from leaf to mean source
- *gflux*: number [**W/m²**]
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
- *dsttc_star*: number [**dg C**]
corrected temperature gradient between soil and mean source
- *theta_0_20*: number [**cm³/cm³**]
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/m³**]
absolute vapor pressure
- *t_{air}*: number [**degr.C**]
air temperature
- *srad*: number [**W/m²**]
global radiation
- *u_{ref}*: number [**m/s**]
friction velocity from ??
- *prec*: number [**mm**]
precipitation

74.2 SSOC

Sun-Shade Open Canopy. See also [Plauborg et al., 2010]

```
< SSOC (solver solver) ; Default cxsparse value.
      (cite cite ...) ; Has default value.
      (hypostomatous true)
      (max_iteration 1500)
      (z_0b 0.0006 [m])
      ;; Shared parameters are described in chapter 74.
      (description description) >
```

- *solver*: **solver** component (see chapter 69)
Component (default 'cxsparse')
Model used for solving the energy balance equation system.
- *hypostomatous*: boolean (see section 4.1.2)
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 [**m² m⁻²**]
Leaf area index.
- *s*: number [**Pa K⁻¹**]
Slope of water vapour pressure curve.
- *lambda*: number [**J kg⁻¹**]
Latent heat of vaporization in atmosphere.

- *rho_a*: number [**kg m⁻³**]
Air density.
- *gamma*: number [**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 [**m s⁻¹**]
Heat conductance in the atmosphere - from canopy point to reference height (screen height).
- *g_H_s_c*: number [**m s⁻¹**]
Heat conductance from soil surface to canopy point.
- *g_H_sun_c*: number [**m s⁻¹**]
Heat conductance from sunlit leaves to canopy point.
- *gb_W_sun*: number [**m s⁻¹**]
Water conductance for sunlit leaves boundary layer.
- *g_W_sun_c*: number [**m s⁻¹**]
Water conductance from sunlit leaves to canopy point.
- *G_W_sun_c*: number [**W m⁻² K⁻¹**]
Scaled water conductance from sunlit leaves to canopy point.
- *g_H_shadow_c*: number [**m s⁻¹**]
Heat conductance from shadow leaves to canopy point.
- *gb_W_shadow*: number [**m s⁻¹**]
Water conductance for shadow leaves boundary layer.
- *g_W_shadow_c*: number [**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 [**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 [$\mathbf{W\ m^{-2}}$]
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 [$\mathbf{W\ m^{-2}}$]
Sensible heat flux from the soil.
- *H_sun*: number [$\mathbf{W\ m^{-2}}$]
Sensible heat flux from the sunlit leaves to the canopy point.
- *H_shadow*: number [$\mathbf{W\ m^{-2}}$]
Sensible heat flux from the shadow leaves to canopy point.
- *H_c_a*: number [$\mathbf{W\ m^{-2}}$]
Sensible heat flux from the canopy point to free atmosphere.
- *LE_sun*: number [$\mathbf{W\ m^{-2}}$]
Latent heat flux from the sunlit leaves to the canopy point.
- *LE_shadow*: number [$\mathbf{W\ m^{-2}}$]
Latent heat flux from the shadow leaves to the canopy point.
- *LE_atm*: number [$\mathbf{W\ m^{-2}}$]
Latent heat flux from the canopy point to the free atmosphere.
- *E_trans*: number [$\mathbf{mm/h}$]
Leaf transpiration.

74.3 none

No SVAT in effect.

Used by bioclimate default svat (see 18.1, page 107) .

Chapter 75

tertiary

Transport of water and solute outside the matrix.

Used by movement component Tertiary (see 47, page 257) .

75.1 biopores

Tertiary domain divided into biopore classes.

```
< biopores (pressure_initiate -3 [cm])
            (pressure_end -30 [cm])
            (pond_max 0.05 [cm])
            (classes classes ...)
            (pressure_limit pressure_limit)
            (pressure_barrier 5 [cm])
            (active_msg range)
            (active active ...)
            (deactivate_steps 3) >
```

- *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 19) 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 transfer between the domains. If you specify a too small value for this parameter, the solution may be unstable.
- *active_msg*: string (see section 4.1.5)
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 4.1.2) 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 [**cm**³]
Water volume.
- *water_height*: number [**cm**]
Water volume multiplied with surface area.
- *solute_mass*: submodel (see section 4.1.7) sequence
Total amount of solutes in biopores.

<i>< name value ></i>	Log Variables
– <i>value</i> : number [g]	Value for chemical.
– <i>name</i> : string (see section 4.1.5)	Name of chemical.
- *solute_storage*: submodel (see section 4.1.7) sequence
Total amount of solutes in biopores divided by surface area.

<i>< name value ></i>	Log Variables
– <i>value</i> : number [g/cm ²]	Value for chemical.
– <i>name</i> : string (see section 4.1.5)	Name of chemical.
- *ddt*: number [**h**]
Emulated timestep. Timestep scaled for available water.

75.2 none

No tertiary transport.

Used by movement rectangle Tertiary (see 47.2, page 257) .

75.3 old

Tertiary water and solute movement based on the obsolete 'macro' and 'mactrans' components. Provided for backward compatibility.

Used by movement vertical Tertiary (see 47.3, page 258) .

```
< old  (macro macro)
      (mactrans mactrans)  ; Default default value. >
```

- *macro*: **macro** component (see chapter 45)
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 46)
Component (default 'default')
Solute transport model in macropores.

Chapter 76

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 40, page 199) .

76.1 M_Q

Millington-Quirk. $\Theta^{7/3} / \Theta_{sat}^2$.

Used by horizon component tortuosity (see 40, page 199) .

76.2 linear

Linear Impedance factor. $a + b \Theta$.

< linear (b 2 [])
(a a) >

- *b*: number (dimensionless)
Parameter (default 2)
Theta factor.
- *a*: number [cm^3/cm^3]
Optional parameter
Theta offset. By default, this corresponds to the wilting point.

Chapter 77

transform

Generic transformations between soil components.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

77.1 equilibrium

Two soil components reaching for equilibrium.

```
< equilibrium (equilibrium equilibrium)  
          (k_AB k_AB)  
          (k_BA k_BA)  
          ;; Shared parameters are described in chapter 77.  
          (description description)  
          (cite) >
```

- *equilibrium*: **equilibrium** component (see chapter 34)
Function for calculating equilibrium between A and B.
- *k_AB*: **number** component (see chapter 50)
Transformation rate from soil component 'A' to 'B' [h⁻¹].
- *k_BA*: **number** component (see chapter 50)
Optional component
Transformation rate from soil component 'B' to 'A' [h⁻¹]. By default, this is identical to 'k_AB'.

Chapter 78

transport

Solute transport in primary domain.

78.1 Hansen

Solute transport using convection-dispersion.

Used by movement vertical matrix_solute (see 47.3, page 258) .

78.2 Mollerup

Coupled vertical and horizontal transport. See Mollerup 2007 for details.

Used by movement rectangle matrix_solute (see 47.2, page 257) .

```
< Mollerup (solver solver) ; Default cxsparse value.  
          (debug 0)  
          (stabilizing_method Timestep_reduction)  
          (enable_boundary_diffusion true)  
          (upstream_weight 1 [<fraction>]) >
```

- *solver*: **solver** component (see chapter 69)
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.
- *stabilizing_method*: string (see section 4.1.5)
Optional parameter (default 'Timestep_reduction')
Method used for stabilizing results. Must be one of these:

None: No stabilizing method applied

Timestep_reduction: Reduce timesteps.

Streamline_diffusion: Increase diffusion.
- *enable_boundary_diffusion*: boolean (see section 4.1.2)
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.

78.3 convection

Pure forward calculation of flow except through upper boundary. $J[\text{edge}] = q[\text{edge}] * C_{\text{old}}[\text{upstream}]$

Used by movement rectangle *matrix_solute* (see 47.2, page 257) , and movement vertical *matrix_solute* (see 47.3, page 258) .

78.4 none

Disable all transport except through boundaries.

Used by movement solute *matrix_solid* (see 47.1, page 257) , movement rectangle *matrix_solute* (see 47.2, page 257) , and movement vertical *matrix_solute* (see 47.3, page 258) .

Chapter 79

ui

Top level user interface.

Used by Toplevel @ ui (see 93.27, page 516) .

79.1 none

No user unterface.

This is useful when running from a batch program, or as a component in a larger system.

79.2 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 80

uifilter

Presentation of data in the user interface.

80.1 base

Base filter for the user interface.

80.2 raw

A ‘base’ model (see 80.1, page 407) build into Daisy.
Raw filter for the user interface.

80.3 simple

A ‘base’ model (see 80.1, page 407) build into Daisy.
Simple filter for the user interface.

```
< simple (default default)  
      (all all ...) >
```

- *default*: string (see section 4.1.5)
Parameter
Name of default component.
- *all*: submodel (see section 4.1.7) sequence
List of components to support.

```
< name all... >
```

- *name*: string (see section 4.1.5)
Parameter
Name of this component.
- *all*: submodel (see section 4.1.7) sequence
List of models to support.

```
< name all... >
```

- * *name*: string (see section 4.1.5)
Parameter
Name of this model.
- * *all*: string (see section 4.1.5) sequence
Parameter
List of parameters to support.

Chapter 81

uiitem

User interface information about a item.

< component (name *name*) >

- *name*: string (see section 4.1.5)
Parameter
Name of user interface unit.

81.1 raw

Raw item for the user interface.

Chapter 82

unit

Specify units of physical quantities.

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 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

82.1 SI

Base parameterization for all SI based units.

```
< SI (time 0)  
      (length 0)  
      (mass 0)  
      (electric_current 0)  
      (thermodynamic_temperature 0)  
      (amount_of_substance 0)  
      (luminous_intensity 0)  
      ;; Shared parameters are described in chapter 82.  
      (description description)  
      (cite) >
```

- *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].

82.2 SIfactor

A ‘SI’ model (see 82.1, page 411) build into Daisy.

Convert to SI base units by multiplying with a factor.

```
< SIfactor  (factor factor)
              ;; Shared parameters are described in section 82.1.
              (time 0)
              (length 0)
              (mass 0)
              (electric_current 0)
              (thermodynamic_temperature 0)
              (amount_of_substance 0)
              (luminous_intensity 0)
              (description description)
              (cite) >
```

- *factor*: number (dimensionless)
Parameter
Factor to multiply with to get base unit.

82.3

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.

Unitless.

82.4 %

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.

Percent.

82.5 <fraction>

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.

Unitless.

82.6 <none>

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Unitless.

82.7 A

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Ampere.

82.8 J

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Joule.

82.9 J/cm²/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Joule per square centimeter per hour.

82.10 K

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kelvin.

82.11 L/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Volume per mass.

82.12 MJ/d/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Megajoule per day per square meter.

82.13 MJ/h/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Megajoule per hour per square meter.

82.14 MJ/m²/d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Megajoule per square meter per day.

82.15 MJ/m²/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Megajoule per square meter per hour.

82.16 MPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Megapascal.

82.17 MPa⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Inverse pressure.

82.18 Mg DM/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Ton dry matter per hectar.

82.19 Mg DM/ha/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Harvest and fertilizing.

82.20 Mg w.w./ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Wet weight per area.

82.21 Mg/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Ton per hectar.

82.22 Pa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Pascal.

82.23 Pa⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base inverse pressure.

82.24 T w.w./ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Wet weight per area.

82.25 W

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Watt.

82.26 $W m^{-2}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Watt per square meter.

82.27 W/cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Watt per square centimeter.

82.28 W/m^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Watt per square meter.

82.29 ag/L

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.30 cd

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Candela.

82.31 cm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Centimeter.

82.32 cm/d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil water movement.

82.33 cm/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil water movement.

82.34 cm^{-1}

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Per centimeter.

82.35 cm^{-2}

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Per square centimeter.

82.36 cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Square centimeter.

82.37 cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Cube centimeter.

82.38 $\text{cm}^3 \text{ H}_2\text{O}/\text{cm}^3$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil water fraction.

82.39 cm^3/cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil water fraction.

82.40 cm^3/g

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Volume per mass.

82.41 cm^3/ng

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Volume per mass.

82.42 d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Day.

82.43 d^{-1}

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Day.

82.44 fg/L

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.45 g

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Gram.

82.46 g C/cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil scale carbon per area.

82.47 g C/cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Carbon concentration.

82.48 g C/m^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Crop scale carbon per area.

82.49 g DM/cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Gram dry matter per square centimeter.

82.50 g DM/m^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Crop scale dry matter per area.

82.51 g N/cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil scale nitrogen per area.

82.52 $\text{g N/cm}^2/\text{h}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen per aquare centimeter per hour.

82.53 g N/cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen concentration.

82.54 g N/l

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen concentration.

82.55 g N/m^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Crop scale nitrogen per area.

82.56 $\text{g N/m}^2/\text{h}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen per aquare meter per hour.

82.57 $\text{g w.w.}/\text{m}^2$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Wet weight per area.

82.58 $\text{g}/\text{cm}/\text{h}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per length flux.

82.59 g/cm^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Soil scale mass per area.

82.60 $\text{g}/\text{cm}^2/\text{h}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mass per area flux.

82.61 $\text{g}/\text{cm}^2/\text{mm}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Irrigation and percolation concentration.

82.62 g/cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Solute concentration.

82.63 g/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale pesticide mass per area.

82.64 $\text{g}/\text{ha}/\text{d}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.65 $\text{g}/\text{ha}/\text{h}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.66 g/ha/mm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
gram per hectare per millimeter.

82.67 g/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Gram per kilogram.

82.68 g/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Crop scale mass per area.

82.69 g/m²/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Gram per square meter per hour.

82.70 g/mol

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Gram per mole.

82.71 h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Hour.

82.72 hPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Hectopascal.

82.73 h⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Hour.

82.74 ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Hectare.

82.75 hours

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Hour.

82.76 kPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kilopascal.

82.77 kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kilogram.

82.78 kg C/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale carbon per area.

82.79 kg C/ha/d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale carbon.

82.80 kg C/ha/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale application and removal of carbon.

82.81 kg DM/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kilogram dry matter per hectare.

82.82 kg DM/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per area.

82.83 kg N/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale nitrogen per area.

82.84 kg N/ha/d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale nitrogen.

82.85 kg N/ha/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale application and removal of nitrogen.

82.86 kg N/ha/mm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Irrigation and percolation concentration.

82.87 kg N/ha/y

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Deposition.

82.88 kg m⁻² s⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per area flux.

82.89 kg w.w./ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Wet weight per area.

82.90 kg/Pa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per pressure.

82.91 kg/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Field scale mass per area.

82.92 kg/ha/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Harvest and fertilizing.

82.93 kg/ha/mm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Irrigation and percolation concentration.

82.94 kg/ha/y

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Deposition.

82.95 kg/m

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per length.

82.96 kg/m/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per length flux.

82.97 kg/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per area.

82.98 kg/m²/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per area flux.

82.99 kg/m³

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per volume.

82.100 kg/mol

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kilogram per mole.

82.101 kgN/ha/year

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Deposition.

82.102 l/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Liter per hectar.

82.103 l/ha/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Liter per hectar per hour.

82.104 l/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Volume per mass.

82.105 m

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Meter.

82.106 $m s^{-1}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base speed.

82.107 m/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base speed.

82.108 m^{-1}

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Per meter.

82.109 m^{-2}

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Per aquare meter.

82.110 $m^{-2} kg s^{-1}$

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base mass per area flux.

82.111 m^2

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Square meter.

82.112 m^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Cube meter.

82.113 m^3/cm^3

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Kilo.

82.114 m^3/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Base volume per mass.

82.115 mg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Miligram.

82.116 mg N/kg dry soil

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen concentration in dry soil.

82.117 mg N/l

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Nitrogen concentration.

82.118 mg/L

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.119 mg/g

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Milligram per gram.

82.120 mg/ha

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Miligram per hectare

82.121 mg/ha/d

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.122 mg/ha/h

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.123 mg/l

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.124 mg/m²

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Crop scale pesticide per area.

82.125 minutes

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Minute.

82.126 ml/g

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Volume per mass.

82.127 mm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Millimeter.

82.128 mm/d

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Percolation intensity.

82.129 mm/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Percolation intensity.

82.130 mm/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Percolation intensity.

82.131 mm⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Per millimeter.

82.132 mmol/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Millimole per kilogram.

82.133 mmol/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Millimole per square meter.

82.134 mmol/m²/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Millimole per square meter per second.

82.135 mol

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mole.

82.136 mol/kg

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mole per kilogram.

82.137 mol/m²

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mole per square meter.

82.138 mol/m²/s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mole per square meter per second.

82.139 ng/L

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.140 ng/MPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nanogram per megapascal.

82.141 ng/cm³

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Low solute concentration.

82.142 ng/kPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Nanogram per kilopascal.

82.143 ng/l

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Lower solute concentration.

82.144 ng/mm/h

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Mass per length flux.

82.145 none

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Unitless.

82.146 pg/L

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.147 ppm

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Part per million.

82.148 s

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Second.

82.149 s⁻¹

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Second.

82.150 seconds

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Second.

82.151 t/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Ton per hectar.

82.152 ug

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Microgram.

82.153 ug/L

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Concentration.

82.154 ug/L/MPa

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Microgram per liter per megapascal.

82.155 ug/ha

A ‘Sifactor’ model (see 82.2, page 412) build into Daisy.
Microgram per hectare

82.156 ug/ha/d

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.157 ug/ha/h

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Pesticide application.

82.158 ug/l

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Low solute concentration.

82.159 ug/m²

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Microgram per square meter.

82.160 ug/m²/h

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Microgram per square meter per hour.

82.161 ug/m³

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Mass per volume.

82.162 um

A ‘SIfactor’ model (see 82.2, page 412) build into Daisy.
Micrometer.

82.163 base

A base unit.

82.164 <error>

A ‘base’ model (see 82.163, page 428) build into Daisy.
Bogus unit.

82.165 <unknown>

A ‘base’ model (see 82.163, page 428) build into Daisy.
Nothing is known about the dimension of this unit.

82.166 dgEast

A ‘base’ model (see 82.163, page 428) build into Daisy.
Degrees East of Greenwich.

82.167 dgNorth

A ‘base’ model (see 82.163, page 428) build into Daisy.
Degrees North of Equator.

82.168 dry soil fraction

A ‘base’ model (see 82.163, page 428) build into Daisy.
Fraction of dry soil.

82.169 rad

A ‘base’ model (see 82.163, page 428) build into Daisy.
Radians

82.170 factor

Convert to base units by multiplying with a factor.

```
< factor  (base base)
          (factor factor)
          ;; Shared parameters are described in chapter 82.
          (description description)
          (cite) >
```

- *base*: string (see section 4.1.5)
Parameter
Base unit to convert to and from.
- *factor*: number (dimensionless)
Parameter
Factor to multiply with to get base unit.

82.171 dg

A ‘factor’ model (see 82.170, page 429) build into Daisy.
Degrees

82.172 dgSouth

A ‘factor’ model (see 82.170, page 429) build into Daisy.
Degrees North of Equator.

82.173 dgWest

A ‘factor’ model (see 82.170, page 429) build into Daisy.
Degrees West of Greenwich.

82.174 new dg

A ‘factor’ model (see 82.170, page 429) build into Daisy.
New degrees

82.175 ppm dry soil

A ‘factor’ model (see 82.170, page 429) build into Daisy.
Part per million in dry soil.

82.176 offset

Convert to base units by multiplying factor, then subtracting offset.

```
< offset (base base)
      (factor 1 [])
      (offset 0 [])
      ;; Shared parameters are described in chapter 82.
      (description description)
      (cite) >
```

- *base*: string (see section 4.1.5)
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.

82.177 dg C

A ‘offset’ model (see 82.176, page 430) build into Daisy.
degree Celcius.

82.178 dg F

A ‘offset’ model (see 82.176, page 430) build into Daisy.
degree Fahrenheit.

82.179 pF

log10 (- cmH2O).

Chapter 83

uz1d

The 'uz1d' component handles the horizontal water movement in the unsaturated zone soil matrix.

83.1 none

Disable transport

Used by uzrect v+h horizontal (see 85.3, page 437) .

83.2 richards

A numerical solution to Richard's Equation.

```
< richards (max_time_step_reductions 4)
           (time_step_reduction 4)
           (max_iterations 25)
           (max_absolute_difference 0.02 [cm])
           (max_relative_difference 0.001 [])
           (K_average K_average) ; Default arithmetic value. >
```

- *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 17)
Optional component (default 'arithmetic')
Model for calculating average K between cells.

Chapter 84

uzmodel

The 'uzmodel' component handles the vertical water movement in the unsaturated zone soil matrix.

84.1 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 85.3, page 437) , and movement vertical matrix_water (see 47.3, page 258) .

```
< lr (overflow_warn true)
    (h_fc -100 [cm])
    (z_top -10 [cm]) >
```

- *overflow_warn*: boolean (see section 4.1.2)
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.

84.2 none

No water movement, and no sink.

84.3 richards

A numerical solution to Richard's Equation.

Used by uzrect v+h vertical (see 85.3, page 437) , and movement vertical matrix_water (see 47.3, page 258) .

```

< richards (debug 0)
          (max_time_step_reductions 16)
          (time_step_reduction 4)
          (max_iterations 25)
          (max_number_of_small_time_steps 200000)
          (msg_number_of_small_time_steps 5000)
          (max_absolute_difference 0.02 [cm])
          (max_relative_difference 0.001 [])
          (K_average K_average) >

```

- *debug*: integer
Parameter (default 0)
Print additional debug messages, higher numbers means more messages.
- *max_time_step_reductions*: integer
Parameter (default 16)
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_number_of_small_time_steps*: integer
Parameter (default 200000)
Maximum number of small time steps in a large time step.
- *msg_number_of_small_time_steps*: integer
Parameter (default 5000)
Number of small time steps in a large time step between message.
- *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 17)
Optional component
Model for calculating average K between cells. The default model is 'geometric' if there are macropores, and 'arithmetic' otherwise.

Chapter 85

uzrect

2D water movement in a rectangular grid.

85.1 Mollerup

A finite volume solution to matrix water transport. See Mollerup 2007 for details.

Used by movement rectangle matrix_water (see 47.2, page 257) .

```
< Mollerup (solver solver) ; Default cxsparse value.
  (debug 0)
  (max_time_step_reductions 16)
  (time_step_reduction 4)
  (max_iterations 12)
  (max_number_of_small_time_steps 20000)
  (msg_number_of_small_time_steps 100)
  (max_absolute_difference 0.02 [cm])
  (max_relative_difference 0.001 [])
  (K_average K_average) ; Default arithmetic value.
  (max_iterations_timestep_reduction_factor 0)
  (max_pressure_potential 1e+009 [])
  (min_pressure_potential -1e+009 [])
  (forced_T forced_T) >
```

- *solver*: **solver** component (see chapter 69)
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 16)
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_number_of_small_time_steps*: integer
Parameter (default 20000)
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_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 25)
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_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.

85.2 const

Steady-state water flow.

Used by movement rectangle matrix_water (see 47.2, page 257) .

```
< const (q_x 0 [cm/h])
      (q_z 0 [cm/h]) >
```

- q_x : number [cm/h]
Parameter (default 0)
Horizontal flow.
- q_z : number [cm/h]
Parameter (default 0)
Vertical flow upwards.

85.3 $v+h$

Transport water in the matrix in two phases, first vertical, then horizontal.

Used by movement rectangle `matrix_water` (see 47.2, page 257) .

```
< "v+h" (vertical vertical ...) ; Has default value.
      (horizontal horizontal ...) ; Has default value. >
```

- *vertical*: **uzmodel** component (see chapter 84) 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 83) 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.

Chapter 86

vegetation

That green stuff.

```
< component (description description)
              (cite)
              (EpInterchange 0.6 [<fraction>]) >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *EpInterchange*: number [<fraction>]
Parameter (default 0.6)
Canopy adsorption fraction of unreachd potential soil evaporation.

Log Variables

- *LAI*: number [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 [m^2/m^2]
Fraction of soil covered by crops
- *LAIvsH*: plf [$\text{m}^2/\text{m}^2 \rightarrow \text{cm}$]
Total canopy LAI below given height
- *HvsLAI*: plf [$\text{cm} \rightarrow \text{m}^2/\text{m}^2$]
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.

86.1 crops

Keep track of all crops on the field.

Used by column default Vegetation (see 24.1, page 138) .

```
< crops (crops)
    (ForcedLAI)
    ;; Shared parameters are described in chapter 86.
    (description description)
    (cite)
    (EpInterchange 0.6 [<fraction>]) >
```

- *crops*: **crop** component (see chapter 27) sequence
Component (default: an empty sequence)
List of crops growing in the field
- *ForcedLAI*: submodel (see section 4.1.7) 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 [$\text{m}^2/\text{m}^2 \rightarrow \text{yday}$]
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.

86.2 permanent

Permanent (non-crop) vegetation.

```
< permanent (Height 80 [cm])
              (Root Root) ; Has default value.
              (Canopy Canopy) ; Has default value.
              (root_DM 2 [Mg DM/ha])
              (N_actual N_actual)
              (LAIvsDAY LAIvsDAY)
              (YearlyLAI)
              (LAIfactor 1 [])
              (N_per_LAI 10 [kg N/ha/LAI])
              (DM_per_LAI 0.5 [Mg DM/ha/LAI])
              (litter_am litter-am ...) ; Has default value.
              (Albedo 0.2 [])
              ;; Shared parameters are described in chapter 86.
              (description description)
              (cite)
              (EpInterchange 0.6 [<fraction>]) >
```

- *Height*: number [cm]
Parameter (default 80)
Permanent height of vegetation.
- *Root*: **RootSystem** fixed component (see section 93.13)
Submodel (has fully specified default value)
Root system.
- *Canopy*: **CanopySimple** fixed component (see section 93.15)
Submodel (has fully specified default value)
Canopy.
- *root_DM*: number [Mg DM/ha]
Parameter (default 2)
Permanent root drymatter.
- *N_actual*: number [g N/m²]
Optional state variable
N uptake until now (default: 'N_demand').
- *LAIvsDAY*: plf [$\text{m}^2/\text{m}^2 \rightarrow \text{yday}$]
Parameter
LAI as a function of Julian day. These numbers are used when there are no yearly numbers (YearlyLAI).

- *YearlyLAI*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Yearly LAI measurements.

```
< year LAIvsDAY >
```

 - *year*: integer
Parameter
Year for which to use yearly LAI measurements.
 - *LAIvsDAY*: plf [$\text{m}^2/\text{m}^2 \rightarrow \text{yday}$]
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 8) 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 [$\text{g N/m}^2/\text{h}$]
Nitrogen uptake this hour.
- *N_litter*: number [$\text{g N/m}^2/\text{h}$]
Nitrogen in litter this hour.

86.3 Bush

A 'permanent' model (see 86.2, page 441) defined in 'vegetation.dai'.

86.4 Coniferous

A ‘permanent’ model (see 86.2, page 441) defined in ‘vegetation.dai’.

86.5 Grass

A ‘permanent’ model (see 86.2, page 441) defined in ‘vegetation.dai’.

86.6 Hardwood

A ‘permanent’ model (see 86.2, page 441) defined in ‘vegetation.dai’.

Chapter 87

vernalization

Requirement for a cold period before flowering.

87.1 default

Temperature sum dependent vernalization.

Used by crop Rye Vernal (see 27.33, page 158) , crop Winter Barley Vernal (see 27.49, page 160) , crop Winter Barley; Koge Vernal (see 27.51, page 160) , crop Winter Rape Vernal (see 27.52, page 160) , crop Winter Wheat Vernal (see 27.53, page 160) , crop Grass to grain Vernal (see 27.13, page 156) , crop Ryegrass Vernal (see 27.34, page 158) , crop Wclover Vernal (see 27.45, page 159) , crop Rug Vernal (see 27.32, page 158) , crop Vinterbyg Vernal (see 27.42, page 159) , crop Vinterhvede Vernal (see 27.43, page 159) , crop Froegraes Vernal (see 27.11, page 156) , and crop Vinterraps Vernal (see 27.44, page 159) .

```
< default (DSLIm DSLIm)  
          (TaLim TaLim)  
          (TaSum TaSum) >
```

- *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.

87.2 none

No vernalization.

Used by crop default Vernal (see 27.1, page 153) .

Chapter 88

volume

A subset of 3D space.

Used by Irrigation @ event volume (see 93.29, page 517) .

88.1 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 14.18, page 76) , action table volume (see 14.45, page 83) , action extern_fertigation volume (see 14.7, page 71) , action extern_subsoil volume (see 14.8, page 72) , log select volume (see 44.3, page 234) , select volume_base volume (see 67.15, page 356) , and select flow volume (see 67.4, page 353) .

```
< box (description description)
      (cite)
      (bottom bottom)           ; Default state value.
      (top top)                 ; Default state value.
      (left left)               ; Default state value.
      (right right)            ; Default state value.
      (front front)            ; Default state value.
      (back back)              ; Default state value. >
```

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.
- *bottom*: **bound** component (see chapter 21)
Component (default 'state')

```
(bottom state (type none))
```

Parameter description:
Lower boundary on the z-axis.

- *top*: **bound** component (see chapter 21)
Component (default 'state')

`(top state (type none))`

Parameter description:
Upper boundary on the z-axis.

- *left*: **bound** component (see chapter 21)
Component (default 'state')

`(left state (type none))`

Parameter description:
Lower boundary on the x-axis.

- *right*: **bound** component (see chapter 21)
Component (default 'state')

`(right state (type none))`

Parameter description:
Upper boundary on the x-axis.

- *front*: **bound** component (see chapter 21)
Component (default 'state')

`(front state (type none))`

Parameter description:
Lower boundary on the y-axis.

- *back*: **bound** component (see chapter 21)
Component (default 'state')

`(back state (type none))`

Parameter description:
Upper boundary on the y-axis.

Chapter 89

weather

A 'wsource' is a source of raw weatherdata.

89.1 weather

Weather interface implementation.

```
< weather (snow_fraction snow_fraction) ; Has default value.  
          (max_rain max_rain) >
```

- *snow_fraction*: plf [**dg C** \rightarrow **<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.

- *max_rain*: number [**mm**]
Optional parameter
Largest amount of rain in one timestep. By default, no limit on rain.

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 [**W/m²**]
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 [**W/m²**]
Average radiation this day.
- *diffuse_radiation*: number [**W/m²**]
Diffuse radiation this hour.
- *daily_extraterrastial_radiation*: number [**W/m²**]
Extraterrastial 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.
- *relative_humidity*: number [**<fraction>**]
Relative 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 4.1.7) sequence
Total atmospheric deposition of nitrogen.

< name value > Log Variables

- *value*: number [**g/cm²/h**]
Value for chemical.
- *name*: string (see section 4.1.5)
Name of chemical.

89.2 base

A ‘weather’ base model (see 89.1, page 449) build into Daisy.

Weather that does not change during the simulation.

```

< base  (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 ...)
        (PrecipScale PrecipScale ...)
        (TempOffset TempOffset ...)
        (Begin Begin)
        (End End)
        (begin begin)
        (end end)
        ;; Shared parameters are described in section 89.1.
        (snow_fraction snow_fraction)           ; Has default value.
        (max_rain max_rain) >

```

- *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/m²**]
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 4.1.5)
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 4.1.5)
Optional parameter
Name of weather station.
- *Note*: string (see section 4.1.5)
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.

- *PrecipScale*: number (dimensionless) sequence
Optional parameter
Scale factors for precipitation. Can contain one or twelve numbers, in the later case the numbers corresponds to months.
- *TempOffset*: number [**dg C**] sequence
Optional parameter
Temperature offset. Can contain one or twelve numbers, in the later case the numbers corresponds to months.
- *Begin*: **Time** fixed component (see section 93.21)
Optional submodel
Beginning of weather data.
- *End*: **Time** fixed component (see section 93.21)
Optional submodel
End of weather data.
- *begin*: **Time** fixed component (see section 93.21)
Optional submodel
Only use data after this date.
- *end*: **Time** fixed component (see section 93.21)
Optional submodel
Only use data before this date.

89.3 **const**

A ‘base’ model (see 89.2, page 450) build into Daisy.

Used by weather combine reserve (see 89.7, page 458) .

89.4 **none**

A ‘const’ model (see 89.3, page 454) build into Daisy.

No weather. No precipitation or global radiation, temperature is 10 dg Celcius. The static information is taken from a climate station operated by Copenhagen University, and located in Taastrup, Denmark. You can overwrite any parameters to simulate an experiment under controled, constant weather conditions.

89.5 **table**

A ‘base’ model (see 89.2, page 450) build into Daisy.

Read weather data from a file.

```

< table (file file)
      (missing missing ...) ; Has default value.
      (filter)
      (original original ...)
      (dim_line dim_line)
      ;; Shared parameters are described in section 89.2.
      (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)
      (snow_fraction snow_fraction) ; Has default value.
      (max_rain max_rain)
      (Station Station)
      (Note Note)
      (PrecipCorrect PrecipCorrect ...)
      (PrecipScale PrecipScale ...)
      (TempOffset TempOffset ...)
      (Begin Begin)
      (End End)
      (begin begin)
      (end end) >

```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

`< tag allowed... >`

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
 - *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
Optional parameter
If true, assume the line after the tags contain dimensions. By default this will be true iff 'original' is not specified.

89.6 default

A 'table' model (see 89.5, page 454) build into Daisy.

Read weather data from specific file.


```

< default  file
            (missing_years)
            ;; Shared parameters are described in section 89.5.
            (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)
            (missing missing ...)           ; Has default value.
            (filter)
            (original original ...)
            (dim_line dim_line)
            (snow_fraction snow_fraction)   ; Has default value.
            (max_rain max_rain)
            (Station Station)
            (Note Note)
            (PrecipCorrect PrecipCorrect ...)
            (PrecipScale PrecipScale ...)
            (TempOffset TempOffset ...)
            (Begin Begin)
            (End End)
            (begin begin)
            (end end) >

```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *missing-years*: submodel (see section 4.1.7) 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 4.1.7)
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 4.1.7)
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.

89.7 combine

A ‘weather’ model (see 89.1, page 449) build into Daisy.

Combine multiple weather sources.

```
< combine (entry entry ...)
          (reserve reserve) ; Default const value.
          ;; Shared parameters are described in section 89.1.
          (snow_fraction snow_fraction) ; Has default value.
          (max_rain max_rain) >
```

- *entry*: submodel (see section 4.1.7) sequence
List of weather sources.

```

< (source source)
  (begin begin)
  (end end)
  (use use ...) ; Has default value. >

```

- *source*: **weather** component (see chapter 89)
Source of weather data.
- *begin*: **Time** fixed component (see section 93.21)
Optional submodel
Use weather data from source after this time. By default, use data from start of source.
- *end*: **Time** fixed component (see section 93.21)
Optional submodel
Use weather data from source until this time. By default, use data until end of source.
- *use*: string (see section 4.1.5) 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 89)
Component (default 'const')

```

(reserve const (GlobRad 0 [W/m^2])
  (AirTemp 10 [dgC])
  (Precip 0 [mm/h])
  (Latitude 56 [dgNorth])
  (Longitude 12 [dgEast])
  (Elevation 30 [m])
  (TimeZone 15 [dgEast])
  (ScreenHeight 2 [m])
  (TAverage 7.8 [dgC])
  (TAmplitude 8.5 [dgC])
  (MaxTDay 209 [yday])
  (Surface reference)
  (Station Taastrup))

```

Parameter description:
Reserve weather model to use when no source match.

89.8 indirect

A 'weather' model (see 89.1, page 449) build into Daisy.

Delegate to another weather source.

```

< indirect (source source)
  ;; Shared parameters are described in section 89.1.
  (snow_fraction snow_fraction) ; Has default value.
  (max_rain max_rain) >

```

- *source*: **weather** component (see chapter 89)
Use this weather source.

89.9 time

A ‘indirect’ model (see 89.8, page 459) 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.

```
< time (offset offset)
      (from from)
      (to to)
      ;; Shared parameters are described in section 89.8.
      (source source)
      (snow_fraction snow_fraction) ; Has default value.
      (max_rain max_rain) >
```

- *offset*: **Timestep** fixed component (see section 93.1)
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.

Chapter 90

wse

The water stress effect on crop growth.

90.1 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.

90.2 none

Water stress has no effect on plant growth.

90.3 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 27.29, page 158) .

< **partial** (`y_half` *y_half*) >

- *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 91

xysource

XY data series.

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

91.1 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.

```
< arithmetic (file file)
(x x)
(missing missing ...) ; Has default value.
(filter)
(original original ...)
(dim_line dim_line)
(y y)
(title title)
(with with)
(style style)
(valid valid) ; Default true value.
(xbar xbar)
(ybar ybar)
;; Shared parameters are described in chapter 91.
(description description)
(cite) >
```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *x*: **number** component (see chapter 50)
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 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

```
< tag allowed... >
```

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 50)
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 4.1.5)
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 4.1.5)
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'.

- *valid*: **boolean** component (see chapter 20)
Component (default 'true')
Ignore entries if this boolean expression is false.
- *xbar*: **number** component (see chapter 50)
Optional component
Expression for calculating x errorbar for this source for each row. The expression can refer to the value in a specific column by the tag for that column.
- *ybar*: **number** component (see chapter 50)
Optional component
Expression for calculating y errorbar for this source for each row. The expression can refer to the value in a specific column by the tag for that column.

91.2 combine

Combine data from multiple time series with a single expression. Data from times series are matched by date.

```
< combine (x x)
          (source source ...)
          (y y)
          (title title)
          (with with)
          (style style)
          ;; Shared parameters are described in chapter 91.
          (description description)
          (cite) >
```

- *x*: **number** component (see chapter 50)
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 70) 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 50)
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 4.1.5)
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 4.1.5)
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'.

91.3 flux

Read a daisy 2d log file, extract flux through line. Values between 'begin' and 'end' will be accumulated, unless 'when' is set.

```
< flux (file file)
      (x x)
      (dimension dimension)
      (z z)
      (missing missing ...) ; Has default value.
      (filter)
      (original original ...)
      (dim_line dim_line)
      (begin begin)
      (end end)
      (when when)
      (title title)
      (pos_dim cm)
      (with with)
      (style style)
      ;; Shared parameters are described in chapter 91.
      (description description)
      (cite) >
```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *x*: number [**cm**]
Optional parameter
Plot flux through this position.
- *dimension*: string (see section 4.1.5)
Optional parameter
Dimension for data. By default, use dimension from file.
- *z*: number [**cm**]
Optional parameter
Plot flux through this depth.

- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

```
< tag allowed... >
```

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.
- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.

- *original*: string (see section 4.1.5) 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 4.1.2)
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 93.21)
Optional submodel
Ignore values before this time.
- *end*: **Time** fixed component (see section 93.21)
Optional submodel
Ignore values after this time.
- *when*: **Time** fixed component (see section 93.21)
Optional submodel
Use value closest to this time.
- *title*: string (see section 4.1.5)
Optional parameter
Name of data series for the legend on the graph.
By default the specified 'z' or 'x' value, time, and tag.
- *pos_dim*: string (see section 4.1.5)
Parameter (default 'cm')
Dimension for soil position.

- *with*: string (see section 4.1.5)
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'.

91.4 inline

A list of x, y pairs.

```
< inline (points points)
          (title title)
          (with with)
          (style style)
          (x_dimension x_dimension)
          (y_dimension y_dimension)
          ;; Shared parameters are described in chapter 91.
          (description description)
          (cite) >
```

- *points*: plf [<unknown> → <unknown>]
Parameter
List of (x y) pairs.
- *title*: string (see section 4.1.5)
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 4.1.5)
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 4.1.5)
Parameter
Dimension for x points.
- *y_dimension*: string (see section 4.1.5)
Parameter
Dimension for y points.

91.5 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.

```
< loop (x x) ; Default x value.
      (begin begin)
      (end end)
      (tag x)
      (y y)
      (title title)
      (with lines)
      (style style)
      (step step)
      ;; Shared parameters are described in chapter 91.
      (description description)
      (cite) >
```

- *x*: **number** component (see chapter 50)
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 4.1.5)
Parameter (default 'x')
Name of free variable to calculate the 'x' and 'y' expressions from.
- *y*: **number** component (see chapter 50)
Expression for calculating the y value.
- *title*: string (see section 4.1.5)
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 4.1.5)
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.

91.6 merge

Merge multiple xy data series into one.

```
< merge  (source source ...)
          (title title)
          (with with)
          (style style)
          (x_dimension x_dimension)
          (y_dimension y_dimension)
          ;; Shared parameters are described in chapter 91.
          (description description)
          (cite) >
```

- *source*: **xysource** component (see chapter 91) sequence
XY data series to merge.
- *title*: string (see section 4.1.5)
Parameter
Name of data series for the legend on the graph.
- *with*: string (see section 4.1.5)
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 4.1.5)
Parameter
Dimension for x points.
- *y_dimension*: string (see section 4.1.5)
Parameter
Dimension for y points.

91.7 profile

Plot soil profile.

```
< profile (file file)
          (dimension dimension)
          (to to)
          (missing missing ...) ; Has default value.
          (filter)
          (original original ...)
          (dim_line dim_line)
          (when when) ; Has partial value.
          (title title)
          (pos_dim cm)
          (with with)
          (style style)
          ;; Shared parameters are described in chapter 91.
          (description description)
          (cite) >
```

- *file*: string (see section 4.1.5)
Parameter
Name of Daisy log file where data is found.
- *dimension*: string (see section 4.1.5)
Optional parameter
Dimension for data. By default, use dimension from file.
- *to*: number [**cm**]
Optional parameter
Plot profile to this depth. By default, plot full profile.
- *missing*: string (see section 4.1.5) sequence
Parameter (has default value with length 2)

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

Parameter description:

List of strings indicating missing values.

- *filter*: submodel (see section 4.1.7) sequence
Submodel (default: an empty sequence)
Only include data from rows that passes all these filters.

```
< tag allowed... >
```

- *tag*: string (see section 4.1.5)
Parameter
Name of column in Daisy log file to filter for.

- *allowed*: string (see section 4.1.5) sequence
Parameter
List of allowable values in filter.
- *original*: string (see section 4.1.5) 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 4.1.2)
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 93.21)
Submodel (has partially specified default value)
Use value closest to this time.
- *title*: string (see section 4.1.5)
Optional parameter
Name of data series for the legend on the graph.

By default the specified 'z' value, time, and tag.
- *pos_dim*: string (see section 4.1.5)
Parameter (default 'cm')
Dimension for soil position.
- *with*: string (see section 4.1.5)
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'.

91.8 xycombine

Combine data from multiple time series with a single expression. Data from times series are matched by date.


```

< xycombine (source source ...)
            (expr expr)
            (title title)
            (with with)
            (style style)
            ;; Shared parameters are described in chapter 91.
            (description description)
            (cite) >

```

- *source*: **xysource** component (see chapter 91) 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 sources must be identical.
- *expr*: **number** component (see chapter 50)
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 4.1.5)
Optional parameter
Name of data series for the legend on the graph.
By default use the expression.
- *with*: string (see section 4.1.5)
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 92

zone

A subset of 3D space.

Used by SoilRegion @ volume (see 93.10, page 492) .

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

- *description*: string (see section 4.1.5)
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 4.1.5) sequence
Parameter (default: an empty sequence)
BibTeX keys that would be relevant for this model or parameterization.

92.1 box

A zone defined by intervals on each axis. By default, the intervals fill the entire axis. You can modify this by setting the parameters.

```
< box (bottom bottom) ; Default state value.  
      (top top) ; Default state value.  
      (left left) ; Default state value.  
      (right right) ; Default state value.  
      (front front) ; Default state value.  
      (back back) ; Default state value.  
      ;; Shared parameters are described in chapter 92.  
      (description description)  
      (cite) >
```

- *bottom*: **bound** component (see chapter 21)
Component (default 'state')

```
(bottom state (type none))
```

Parameter description:
Lower boundary on the z-axis.

- *top*: **bound** component (see chapter 21)
Component (default 'state')

```
(top state (type none))
```

Parameter description:
Upper boundary on the z-axis.

- *left*: **bound** component (see chapter 21)
Component (default 'state')

(left state (type none))

Parameter description:
Lower boundary on the x-axis.

- *right*: **bound** component (see chapter 21)
Component (default 'state')

(right state (type none))

Parameter description:
Upper boundary on the x-axis.

- *front*: **bound** component (see chapter 21)
Component (default 'state')

(front state (type none))

Parameter description:
Lower boundary on the y-axis.

- *back*: **bound** component (see chapter 21)
Component (default 'state')

(back state (type none))

Parameter description:
Upper boundary on the y-axis.

92.2 polygon

A 2D region defined by points connected by lines. The object is assumed to be infinite in the third dimension.

```
< polygon (outer outer ...)
          ;; Shared parameters are described in chapter 92.
          (description description)
          (cite) >
```

- *outer*: submodel (see section 4.1.7) sequence
Points surrounding the polygon in clockwise order.

```
< z x >
```

– *z*: number [**cm**]
Parameter
Vertical position.

– *x*: number [**cm**]
Parameter
Horizontal position.

Chapter 93

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.

93.1 Timestep

Relative time.

```
< Timestep (seconds 0)
           (minutes 0)
           (hours 0)
           (days 0)
           (microseconds 0) >
```

- *seconds*: integer
State variable (default 0)
Number of seconds.
- *minutes*: integer
State variable (default 0)
Number of minutes.
- *hours*: integer
State variable (default 0)
Number of hours.
- *days*: integer
State variable (default 0)
Number of days.
- *microseconds*: integer
State variable (default 0)
Number of microseconds.

93.2 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 ...)
               (PrecipScale PrecipScale ...)
               (TempOffset TempOffset ...)
               (Begin Begin)
               (End End) >

```

- *GlobRad*: number [**W**/m²]
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/m²**]
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.
- *TAplitude*: 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 4.1.5)
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 4.1.5)
Optional parameter
Name of weather station.
- *Note*: string (see section 4.1.5)
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.
- *PrecipScale*: number (dimensionless) sequence
Optional parameter
Scale factors for precipitation. Can contain one or twelve numbers, in the later case the numbers corresponds to months.

- *TempOffset*: number [**dg C**] sequence
Optional parameter
Temperature offset. Can contain one or twelve numbers, in the later case the numbers corresponds to months.
- *Begin*: **Time** fixed component (see section 93.21)
Optional submodel
Beginning of weather data.
- *End*: **Time** fixed component (see section 93.21)
Optional submodel
End of weather data.

93.3 Bioincorporation

Biological incorporation of organic matter in soil.

```
< Bioincorporation (AOM AOM ...) ; Has default value.
(R_max 0.5 [g DM/m2/h])
(k_half 1 [g DM/m2])
(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 8) sequence
Component (has default value with length 2)

```
(AOM "AOM-SLOW-BIOINCORPORATION"
      "AOM-FAST")
```

Parameter description:
Incorporated AM parameters.

- *R_max*: number [**g DM/m²/h**]
Parameter (default 0.5)
Maximal speed of incorporation.
- *k_half*: number [**g DM/m²**]
Parameter (default 1)
Halflife constant.
- *C_per_N_factor*: plf [**(g C/cm²)/(g N/cm²)** → **<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/m²/h**]
Fraction of litter incorporated this hour. The formula is $\text{speed} = (\text{R_max} * \text{litter}) / (\text{k_half} + \text{litter})$.
- *DM*: number [**g DM/m²/h**]
DM removed from surface.
- *C_removed*: number [**g C/m²/h**]
C removed from surface.
- *N_removed*: number [**g N/m²/h**]
N removed from surface.
- *CO2*: number [**g C/m²/h**]
C respired.
- *C_added*: number [**g C/cm³/h**] soil cells
C added to soil.
- *N_added*: number [**g N/cm³/h**] soil cells
N added to soil.

93.4 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 93.21)
Submodel (has partially specified default value)
Time of the harvest operation.
- *crop*: string (see section 4.1.5)
State variable
Name of crop that was harvested.
- *column*: string (see section 4.1.5)
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 [g/m²]
State variable
Total stem dry matter in harvest.
- *stem_N*: number [g/m²]
State variable
Total stem nitrogen in harvest.
- *stem_C*: number [g/m²]
State variable
Total stem carbon in harvest.
- *dead_DM*: number [g/m²]
State variable
Total dead leaf dry matter in harvest.
- *dead_N*: number [g/m²]
State variable
Total dead leaf nitrogen in harvest.

- *dead_C*: number [g/m²]
State variable
Total dead leaf carbon in harvest.
- *leaf_DM*: number [g/m²]
State variable
Total leaf dry matter in harvest.
- *leaf_N*: number [g/m²]
State variable
Total leaf nitrogen in harvest.
- *leaf_C*: number [g/m²]
State variable
Total leaf carbon in harvest.
- *sorg_DM*: number [g/m²]
State variable
Total storage organ dry matter in harvest.
- *sorg_N*: number [g/m²]
State variable
Total storage organ nitrogen in harvest.
- *sorg_C*: number [g/m²]
State variable
Total storage organ carbon in harvest.
- *water_productivity*: number [kg DM/m³ H₂O]
State variable
Storage organ harvested per evapotranspiration.

93.5 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/m2/h C])
        (mrprime 1.5e-007 [kg/J])
        (m1 2 [kg/J])
        (m2 0.00416667 [h-1])
        (rho_s 100 [kg/m3])
        (f_c 0.07 [])
        (rho_l 200 [kg/m3])
        (rho_2 0.5 [m-1])
        (Psa 0.208333 [mm])
        (fsa 0.9 [])
        (K_snow_factor 2.86e-006 [W m5/kg2/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 [**m**⁻¹]
Parameter (default 10)
Snow pack depth melting factor.
- *mtprime*: number [**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 [**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.

93.6 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/cm³/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 4.1.7) 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 [cm⁻¹]
Parameter (default -15000)
Pressure below which no more water will freeze.
- *enable_ice*: boolean (see section 4.1.2)
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 [**erg/cm²/h**] soil edges
Heat flux.
- *conductivity*: number [**erg/cm/dg C/h**]
Heat conductivity.
- *capacity*: number [**erg/cm³/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.

93.7 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 [**cm³/cm³/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 4.1.7) 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 4.1.7) 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 during freezing.

Log Variables

- *K*: number [**cm/h**] soil cells
Hydraulic conductivity.
- *tillage*: number [**cm³/cm³/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 [**cm³/cm³/h**] soil cells
Water sink due to soil drainage.

- *S_{root}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Water sink due to root uptake.
- *sink_{cell}*: integer
Cell with largest forward sink compared to available water.
- *Theta_{primary}*: number [cm^3/cm^3] soil cells
Water content in primary matrix system. Conventionally, this is the intra-aggregate pores.
- *Theta_{secondary}*: number [cm^3/cm^3] soil cells
Water content in secondary matrix system. Conventionally, this is the inter-aggregate pores.
- *S_{sum}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Total water sink (due to root uptake and macropores).
- *S_{incorp}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Incorporated water sink, typically from subsoil irrigation.
- *S_p*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Water sink (due to macropores).
- *S_{ice}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Ice sink due to thawing or freezing.
- *S_{ice_{water}}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Water sink due to thawing or freezing.
- *S_{forward_{total}}*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] 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 [$\text{cm}^3/\text{cm}^3/\text{h}$] 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 presumed to occupy the large pores, so it is h ($\text{Theta}_{\text{sat}} - X_{\text{ice}}$).
- *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 pressure 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 pressure in highest saturated cell. If there are multiple saturated cells in the same depth, the one with the highest pressure will be used.

93.8 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 93.12)
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 mulching. With a crop factor of 1.2 a combined Kc of 1.15 is reached at LAI=5. See also [Kjaersgaard et al., 2008]
Conversion 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 [**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.

93.9 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 93.11, page 492) .

```
< SoilLayer  end horizon  >
```

- *end*: number [**cm**]
Parameter
End point of this layer (a negative number).
- *horizon*: **horizon** component (see chapter 40)
Soil properties of this layer.

93.10 SoilRegion

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 93.11, page 492) .

```
< SoilRegion  volume horizon  >
```

- *volume*: **zone** component (see chapter 92)
Volume covered by this zone.
- *horizon*: **horizon** component (see chapter 40)
Soil properties of this zone.

93.11 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 93.9) 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*: **SoilRegion** fixed component (see section 93.10) sequence
Submodel (default: an empty sequence)
Regions 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.

93.12 Ridge

Surface model after ridging.

Used by Surface @ ridge (see 93.8, page 490) .

```
< 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 [**cm³/cm³**]
Soil water content.
- *Theta_pre*: number [**cm³/cm³**]
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.

93.13 RootSystem

Standard root system model.

```
< RootSystem (rootdens rootdens)
  (ABAprd ABAprd) ; 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)
  (Density Density ...)
  (ABAConc 0 [g/cm3])
  (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 61)
Optional component
Root density model.
- *ABAprd*: **ABAprd** component (see chapter 7)
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 resistance in xyleme.
- *PotRtDpt*: number [**cm**]
Optional state variable
Potential root penetration depth.
- *Depth*: number [**cm**]
Optional state variable
Rooting Depth.
- *Density*: number [**cm/cm³**] soil cells
Optional state variable
Root density in soil layers.
- *ABAConc*: number [**g/cm³**]
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 radiation of that day that was received that hour.

Log Variables

- *production_stress*: number (dimensionless)
SVAT induced stress, or -1 if not applicable.
- *H2OExtraction*: number [$\text{cm}^3/\text{cm}^3/\text{h}$] soil cells
Extraction of H2O in soil layers.
- *NH4Extraction*: number [$\text{g N}/\text{cm}^3/\text{h}$] soil cells
Extraction of NH4-N in soil layers.
- *NO3Extraction*: number [$\text{g N}/\text{cm}^3/\text{h}$] soil cells
Extraction of NO3-N in soil layers.
- *ABAExtraction*: number [$\text{g}/\text{cm}^3/\text{h}$] 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 [$\text{g N}/\text{m}^2/\text{h}$]
NH4-N uptake.
- *NO3Upt*: number [$\text{g N}/\text{m}^2/\text{h}$]
NO3-N uptake.

93.14 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 [(m2/m2)/(g DM/m2)])
                  (SOrgAImod SOrgAImod) ; Has default value.
                  (SOrgPhotEff 1 [])
                  (SpStemAI 0 [(m2/m2)/(g DM/m2)])
                  (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 [m2/m2])
                  (StemAI 0 [m2/m2])
                  (SOrgAI 0 [m2/m2])
                  (LADm -9999.99 [cm2/cm3]) >

```

- *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 [**(m²/m²)/(g DM/m²)**]
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 [**(m²/m²)/(g DM/m²)**]
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 [$(\text{m}^2/\text{m}^2)/(\text{g DM}/\text{m}^2)$]
Parameter (default 0)
Specific stem weight. Used only after the intital phase.
- *StemAIMod*: plf [**DS** \rightarrow **<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>** \rightarrow **cm**]
Parameter
Crop height as function of DS.
- *HvsWStem*: plf [**g DM/m²** \rightarrow **<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/m² to reach full height.

- *LAI Dist0*: number (dimensionless) array of length 3
Parameter
Relative CAI distribution at DS=0.
- *LAI Dist1*: 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 [**m²/m²**]
State variable (default 0)
Leaf Area Index.

- *StemAI*: number [$\mathbf{m}^2/\mathbf{m}^2$]
State variable (default 0)
Stem Area Index.
- *SOrgAI*: number [$\mathbf{m}^2/\mathbf{m}^2$]
State variable (default 0)
Storage Organ Area Index.
- *LADm*: number [$\mathbf{cm}^2/\mathbf{cm}^3$]
State variable (default -9999.99)
Maximal Leaf Area Density.

Log Variables

- *CAI*: number [$\mathbf{m}^2/\mathbf{m}^2$]
Crop Area Index.
- *LAIvsH*: plf [$\mathbf{cm} \rightarrow \mathbf{m}^2/\mathbf{m}^2$]
Accumulated Leaf Area Index at Height.
- *ForcedCAI*: number [$\mathbf{m}^2/\mathbf{m}^2$]
CAI forced upon us by vegetation module.
- *SimCAI*: number [$\mathbf{m}^2/\mathbf{m}^2$]
CAI simulated by crop model.
- *CAImRat*: number (dimensionless)
(CAIm - CAI) / CAIm.

93.15 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 [\mathbf{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. $\text{NIRref} = 0.51$ (Ross, 1975)
- *NIRext*: number (dimensionless)
Parameter (default 0.18)
NIR extinction coefficient. $\text{NIRext} = 0.18$ (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** \rightarrow **<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** \rightarrow **cm**]
Parameter (has default value with 2 points)

(leaf_width (0 3) (2 3))

Parameter description:
Leaf width.

Log Variables

- *CAI*: number [**m²/m²**]
Crop Area Index.
- *LAIvsH*: plf [**cm** \rightarrow **m²/m²**]
Accumulated Leaf Area Index at Height.

93.16 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 8) sequence
Component (has default value with length 2)

```
(Root "AOM-SLOW"
      "AOM-FAST")
```

Parameter description:
Root AM parameters.

- *Stem*: **AOM** component (see chapter 8) sequence
Component (has default value with length 2)

```
(Stem "AOM-SLOW"
      "AOM-FAST")
```

Parameter description:
Stem AM parameters.

- *Leaf*: **AOM** component (see chapter 8) sequence
Component (has default value with length 2)

```
(Leaf "AOM-SLOW"
      "AOM-FAST")
```

Parameter description:
Leaf AM parameters.

- *Dead*: **AOM** component (see chapter 8) sequence
Component (has default value with length 2)

```
(Dead "AOM-SLOW"
      "AOM-FAST")
```

Parameter description:
Dead leaves AM parameters.

- *SOrg*: **AOM** component (see chapter 8) 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 93.21)
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 guarantee 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.

93.17 Production

Crop production in the default crop model.

```

< Production  (ShldResC 0 [<fraction>])
               (ReMobilDS 1.2 [])
               (ReMobilRt 0.1 [d-1])
               (StemRes 0 [g DM/m2])
               (CH20ReleaseRate 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)
               (LargeRtDR 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 [])
               (CH20Pool 0.001 [g CH20/m2])
               (WLeaf 0.001 [g DM/m2])
               (WStem 0 [g DM/m2])
               (WRoot 0.001 [g DM/m2])
               (WSOrg 0 [g DM/m2])
               (WDead 0 [g DM/m2])
               (NLeaf 0 [g N/m2])
               (NStem 0 [g N/m2])
               (NRoot 0 [g N/m2])
               (NSOrg 0 [g N/m2])
               (NDead 0 [g N/m2])
               (NCrop NCrop)
               (C_AM 0 [g C/m2])
               (N_AM 0 [g N/m2])
               (DailyNetRoot 0 [g DM/m2])
               (DailyNetShoot 0 [g DM/m2]) >

```

- *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 [d⁻¹]
Parameter (default 0.1)
Remobilization, release rate.
- *StemRes*: number [g DM/m²]

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 [$\mathbf{dg} \mathbf{C} \rightarrow \langle \mathbf{none} \rangle$]
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 [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 [$\text{g CH}_2\text{O}/\text{m}^2$]
State variable (default 0.001)
CH₂O Pool.
- *WLeaf*: number [$\text{g DM}/\text{m}^2$]
State variable (default 0.001)
Leaf dry matter weight.
- *WStem*: number [$\text{g DM}/\text{m}^2$]
State variable (default 0)
Stem dry matter weight.
- *WRoot*: number [$\text{g DM}/\text{m}^2$]
State variable (default 0.001)
Root dry matter weight.
- *WSOrg*: number [$\text{g DM}/\text{m}^2$]
State variable (default 0)
Storage organ dry matter weight.
- *WDead*: number [$\text{g DM}/\text{m}^2$]
State variable (default 0)
Dead leaves dry matter weight.
- *NLeaf*: number [$\text{g N}/\text{m}^2$]
State variable (default 0)
Nitrogen stored in the leaves.
- *NStem*: number [$\text{g N}/\text{m}^2$]
State variable (default 0)
Nitrogen stored in the stem.
- *NRoot*: number [$\text{g N}/\text{m}^2$]
State variable (default 0)
Nitrogen stored in the roots.
- *NSOrg*: number [$\text{g N}/\text{m}^2$]
State variable (default 0)
Nitrogen stored in the storage organ.

- *NDead*: number [g N/m²]
State variable (default 0)
Nitrogen stored in dead leaves.
- *NCrop*: number [g N/m²]
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/m²]
State variable (default 0)
Added C in plant material.
- *N_{AM}*: number [g N/m²]
State variable (default 0)
Added N in plant material.
- *DailyNetRoot*: number [g DM/m²]
State variable (default 0)
Root growth minus root respiration so far this day.
- *DailyNetShoot*: number [g DM/m²]
State variable (default 0)
Leaf growth minus leaf respiration so far this day.

Log Variables

- *CLeaf*: number [g C/m²]
Leaf C weight.
- *CStem*: number [g C/m²]
Stem C weight.
- *CRoot*: number [g C/m²]
Root C weight.
- *CSOrg*: number [g C/m²]
Storage organ C weight.
- *CDead*: number [g C/m²]
Dead leaves C weight.
- *CCrop*: number [g C/m²]
Crop C weight.
- *PotCanopyAss*: number [g CH₂O/m²/h]
Potential canopy assimilation, i.e. stressfree production.
- *CanopyAss*: number [g CH₂O/m²/h]
Canopy assimilation.
- *NetPhotosynthesis*: number [g CO₂/m²/h]
Net Photosynthesis.
- *AccNetPhotosynthesis*: number [g CO₂/m²]
Accumulated Net Photosynthesis.
- *Respiration*: number [g CH₂O/m²/h]
Crop Respiration.

- *MaintRespiration*: number [g CH₂O/m²/h]
Maintenance Respiration.
- *GrowthRespiration*: number [g CH₂O/m²/h]
Growth Respiration.
- *LeafRespiration*: number [g CO₂/m²/h]
Total Leaf Respiration.
- *StemRespiration*: number [g CO₂/m²/h]
Total Stem Respiration.
- *SOrgRespiration*: number [g CO₂/m²/h]
Total SOrg Respiration.
- *RootRespiration*: number [g CO₂/m²/h]
Total Root Respiration.
- *LeafMaintRespiration*: number [g CO₂/m²/h]
Leaf Maintenance Respiration.
- *StemMaintRespiration*: number [g CO₂/m²/h]
Stem Maintenance Respiration.
- *SOrgMaintRespiration*: number [g CO₂/m²/h]
SOrg Maintenance Respiration.
- *RootMaintRespiration*: number [g CO₂/m²/h]
Root Maintenance Respiration.
- *LeafGrowthRespiration*: number [g CO₂/m²/h]
Leaf Growth Respiration.
- *StemGrowthRespiration*: number [g CO₂/m²/h]
Stem Growth Respiration.
- *SOrgGrowthRespiration*: number [g CO₂/m²/h]
SOrg Growth Respiration.
- *RootGrowthRespiration*: number [g CO₂/m²/h]
Root Growth Respiration.
- *IncWLeaf*: number [g DM/m²/h]
Leaf growth.
- *IncWStem*: number [g DM/m²/h]
Stem growth.
- *IncWSOrg*: number [g DM/m²/h]
Storage organ growth.
- *IncWRoot*: number [g DM/m²/h]
Root growth.
- *DeadWLeaf*: number [g DM/m²/h]
Leaf DM removed.
- *DeadNLeaf*: number [g N/m²/h]
Leaf N removed.

- *DeadWRoot*: number [g DM/m²/h]
Root DM removed.
- *DeadNRoot*: number [g N/m²/h]
Root N removed.
- *C_Loss*: number [g C/m²/h]
C lost from the crop

93.18 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 leaves.
- *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.

93.19 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/cm3])
          (NH4_root_min 0 [g N/cm3])
          (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 efficiency, Leaf.

- *TLRootEff*: plf [**DS** → **<fraction>**]
Parameter (has default value with 2 points)

(TLRootEff (0 0.1) (2 0.1))

Parameter description:
Translocation efficiency, Root.

- *NO3_root_min*: number [**g N/cm³**]
Parameter (default 0)
Minimum nitrate concentration near roots for uptake.
- *NH4_root_min*: number [**g N/cm³**]
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 radiation 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 [g/m²]
Potential nitrogen content in crop.
- *CrNCnt*: number [g/m²]
Critical nitrogen content in crop.
- *NfNCnt*: number [g/m²]
Non-functional nitrogen content in crop.
- *nitrogen_stress*: number (dimensionless)
Nitrogen stress factor.
- *Fixated*: number [g N/m²/h]
N fixation from air.
- *AccFixated*: number [g N/m²]
Accumuated N fixation from air.

93.20 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 93.25)
Submodel (has partially specified default value)
Nitrogen content of DOM pool.
- *C*: **DOM-Element** fixed component (see section 93.25)
Submodel (has partially specified default value)
Carbon content of DOM pool.
- *diffusion_coefficient*: number [cm²/s]
Parameter
Diffusion coefficient.
- *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'.
- *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
Fraction of this pool that ends up in each SMB pools

93.21 Time

Year, month, day and hour, minute, second and microsecond.

Used by Weatherdata @ Begin (see 93.2, page 477) , Harvest @ time (see 93.4, page 482) , and Harvesting @ last_cut (see 93.16, page 502) .

```
< 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 4.1.5)
Current weekday. Monday is 1, Sunday is 7.

93.22 HorHeat

Heat capacity and conductivity per horizon.

Used by horizon component HorHeat (see 40, page 199) .

```
< 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/cm³/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.

93.23 Fetch

A summary file line.

```
< Fetch  tag  >
```

- *tag*: string (see section 4.1.5)
Parameter
The tag of a column in the log file to summarize in this line.

93.24 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.

93.25 DOM-Element

A single element in a Dissolved Organic Matter pool.

Used by DOM @ N (see 93.20, page 512) .

< "DOM-Element" (M *M* ...) >

- *M*: number [**g/cm³**] soil cells
State variable
Mass in water and soil.

Log Variables

- *S*: number [**g/cm³/h**] soil cells
Combined source term.
- *C*: number [**g/cm³**] soil cells
Concentration in water.
- *S_{drain}*: number [**g/cm³/h**] soil cells
Source term (soil drainage only).
- *J_{matrix}*: number [**g/cm²/h**] soil edges
Transportation in matrix (positive up).
- *J_{tertiary}*: number [**g/cm²/h**] soil edges
Transportation outside matrix (positive up).
- *S_p*: number [**g/cm³/h**] soil cells
Source term (macropore transport only).

93.26 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.

93.27 Toplevel

The top level syntax for a Daisy setup file.

```
< 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 79)
Optional component
Top level user interface.
- *input*: **parser** component (see chapter 52)
Optional component
Command to add more information about the simulation.
- *path*: string (see section 4.1.5) 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, separated 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 4.1.5)
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 57)
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 4.1.2)
Parameter (default true)
OBSOLETE: Set this to true to enable the old system of build-in unit conversation.
- *install_directory*: string (see section 4.1.5)
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.

93.28 FetchPretty

A summary file line.

```
< FetchPretty  tag
                (name name)
                (factor 1 []) >
```

- *tag*: string (see section 4.1.5)
Parameter
The tag of a column in the log file to summarize in this line.
- *name*: string (see section 4.1.5)
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.

93.29 Irrigation

Keep track of active irrigation events. Usually not set explicitly, but may be found in a checkpoint.

```
< Irrigation  (event) >
```

- *event*: submodel (see section 4.1.7) 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 4.1.7) sequence
Solutes in irrigation water.

```
<  name value >
```

 - * *name*: string (see section 4.1.5)
State variable
Name of chemical.
 - * *value*: number [g/cm²/mm]
State variable
Value for chemical.

- *volume*: **volume** component (see chapter 88)
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 4.1.5)
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 4.1.2)
State variable
True if event should not declare when it is over.

93.30 StringerCondClause

If condition is true, return value.

< **StringerCondClause** *condition value* >

- *condition*: **boolean** component (see chapter 20)
Condition to test for.
- *value*: string (see section 4.1.5)
Parameter
Value to return.

93.31 IntegerCondClause

If condition is true, return value.

< **IntegerCondClause** *condition value* >

- *condition*: **boolean** component (see chapter 20)
Condition to test for.
- *value*: integer
Parameter
Value to return.

Version

Daisy version 5.14. LaTeX manual generated: Tue Dec 11 12:39:59 2012

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, separated by spaces, the first number representing the factor for January, and the last for December. So if the first number is 1.20 and the weather data for a given day in January 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 ways 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, 2011] Abrahamsen, P. (2011). *Daisy Tutorial*. Distributed with Daisy.
- [Abrahamsen and Hansen, 2000] Abrahamsen, P. and Hansen, S. (2000). Daisy: An open soil-crop-atmosphere model. *Environmental Modelling & 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.
- [FOCUS, 2002] FOCUS (2002). Generic guidance for focus groundwater scenarios. Version 1.1.
- [Gerwitz and Page, 1974] Gerwitz, A. and Page, E. R. (1974). An empirical mathematical model to describe plant root systems. *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. Web page. <http://code.google.com/p/daisy-model/wiki/DaisyDescription>.
- [Hansen et al., 1990] Hansen, S., Jensen, H. E., Nielsen, N. E., and Svendsen, H. (1990). DAISY — soil plant atmosphere 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.
- [Hargreaves and Samani, 1985] Hargreaves, G. and Samani, Z. (1985). Reference crop evapotranspiration from temperature. *Applied Engineering in Agriculture*, 1(2):96–99.
- [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., Møllerup, 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.
- [Plauborg et al., 2010] Plauborg, F., Abrahamsen, P., Gjettermann, B., Møllerup, M., Iversen, B. V., Fulai Liu, M. N. A., and Hansen, S. (2010). Modelling of root ABA synthesis, stomatal conductance, transpiration and potato production under water saving irrigation regimes. *Agricultural Water Management*, 98(3):425–439.
- [Scopel et al., 1998] Scopel, E., Müller, 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.
- [van der Keur et al., 2001] van der Keur, P., Hansen, S., Schelde, K., and Thomsen, A. (2001). Modification of DAISY SVAT model for potential use of remotely sensed data. *Agricultural and Forest Meteorology*, 106(3):215–231.
- [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

- <, 117
- <=, 117
- <error>, 428
- <fraction>, 412
- <none>, 413
- <unknown>, 428
- >, 118
- >=, 118
- *, 229, 265
- +, 229, 265
- , 229, 265
- /, 265
- %, 412
- 2,4-D, 130

- A, 326, 328, 413
- a, 53, 165, 261, 334, 336, 337, 399
- a_DS, 333
- A_F, 55
- A_solute, 176
- a_x, 337
- a_z, 337
- ABA_effect, 292
- ABA_min, 373
- ABAConc, 495
- ABAExtraction, 496
- ABAprd, 494
- ABAprdproduction, 47
- abiotic_factor, 282
- absorbed_shadow_Long_canopy, 112
- absorbed_shadow_NIR_canopy, 112
- absorbed_shadow_PAR_canopy, 112
- absorbed_sun_Long_canopy, 112
- absorbed_sun_NIR_canopy, 112
- absorbed_sun_PAR_canopy, 112
- absorbed_total_Long_canopy, 111
- absorbed_total_Long_soil, 112
- absorbed_total_NIR_canopy, 111
- absorbed_total_NIR_soil, 112
- absorbed_total_PAR_canopy, 111
- absorbed_total_PAR_soil, 111
- acc, 388
- AccFixated, 512
- AccNetPhotosynthesis, 507
- accumulate, 351, 366, 369–371

- ACExt_NIR, 440
- ACExt_PAR, 439
- ACRef_NIR, 440
- ACRef_PAR, 440
- action, 63, 71
- actions, 63, 80, 87
- activate_output, 302
- active, 396
- active_msg, 396
- active_underground, 276
- activity, 81
- adsorption, 89, 123, 321
- adsorption_rate, 321
- adsorption_source, 322
- Ae, 325
- aender_gaes_dybstroelse, 95
- Aert, 155
- after, 150, 189
- after_mm_dd, 148
- ag/L, 415
- age, 485
- air_pressure, 450
- air_temperature, 449
- AirTemp, 451, 478
- Albedo, 442
- albedo, 109, 231, 232, 386, 439
- albedo_dry, 490
- albedo_wet, 490
- alfa, 295
- all, 407
- allow_old_units, 516
- allow_sideways, 141
- allow_upward_flow, 115
- allowed, 85, 184, 187, 306, 309, 313, 366, 368, 456, 464, 467, 472
- alpha, 219, 220, 224, 226, 296, 326, 345, 375, 376
- alpha_fast, 327
- alpha_k, 387
- alpha_r, 387
- alpha_u, 387
- am, 73, 74, 93, 276
- AM_table, 301
- Ammonia, 102
- AmmoniumNitrate, 102

- amount, 64, 68, 69, 83, 149
- amount_of_substance, 412
- and, 118, 144
- Anders Pedersen, 333
- anisotropy, 200
- AOM, 49, 481
- AOM-DIRECT, 50
- AOM-FAST, 50
- AOM-SLOW, 51
- AOM-SLOW-BIOINCORPORATION, 51
- AOM-SLOW-OLD, 51
- Ap_JB1, 211
- Ap_JB2, 211
- Ap_JB3, 211
- Ap_JB4, 211
- Ap_JB5, 211
- Ap_JB6, 211
- Ap_JB7, 211
- aperture, 345
- applied_DM, 140
- aquitard, 191, 214
- arac, 387
- area, 137
- ARExt, 440
- arithmetic, 105, 141, 365, 463
- array, 352
- array_separator, 237
- As, 324
- Askov, 140
- Askov Ap, 211
- Askov B, 212
- Askov C, 212
- Ass, 293
- Ass_vector, 292
- assert, 63
- at, 146–148, 150, 151
- atom, 173
- Atrazine, 130
- attributes, 200
- average, 105
- B, 326, 328
- b, 217, 223, 228, 261, 299, 374, 375, 399
- b1, 386
- b2, 386
- B2M, 113
- b3, 386
- b4, 386
- B_BaC, 215
- B_BaC_Bimodal, 216
- B_C, 217
- B_C_inverse, 218
- B_JB1, 212
- B_JB2, 212
- B_JB3, 212
- B_JB4, 212
- B_JB5, 212
- B_JB6, 212
- B_JB7, 212
- B_solute, 176
- B_vG, 218
- back, 448, 476
- background_mineralization, 280
- balance, 380
- base, 93, 271, 407, 428–430, 450
- batch, 304
- BB, 374
- BB_base, 373
- Beetroot, 155
- before, 151, 189
- before_mm_dd, 148
- Begin, 454, 481
- begin, 185, 187, 272, 306, 454, 459, 467, 469
- Bentazon, 131
- beta, 294, 296, 373
- Bioclimate, 139
- bioclimate, 107
- bioinc, 280
- Bioincorporation, 275, 481
- biomod, 53
- biopore, 113, 245, 246
- bioporefid, 246
- biopores, 395
- black_body_radiation, 261
- boolean, 117
- border, 492
- bottom, 183, 186, 447, 475
- bound, 119, 322
- bound_release, 322
- box, 447, 475
- Broccoli, 155
- Broccoli - transplanted, 155
- Brown87, 319
- brunt, 261
- Brussels sprouts, 155
- Brussels sprouts - transplanted, 155
- brutsaert, 262
- BSI3, 201
- BSI7, 202
- buffer, 276
- Bush, 442
- C, 49, 59, 61, 123, 269, 277, 348, 512, 515
- C_added, 482
- C_AM, 507
- C_below, 125

- c_d, 387
- C_e, 58
- c_e, 58
- C_fraction, 348
- C_JB1, 212
- C_JB2, 212
- C_JB3, 213
- C_JB4, 213
- C_JB5, 213
- C_JB6, 213
- C_JB7, 213
- C_Loss, 509
- C_per_N, 50, 60, 61, 200
- C_per_N_factor, 481
- C_primary, 129
- c_r, 56
- C_removed, 482
- C_secondary, 125
- C_soil, 514
- c_T, 58
- C_u, 78
- c_Vm, 295
- CAI, 500, 501
- CAImRat, 500
- CalciumNitrate, 102
- Canopy, 154, 161, 441
- canopy_dissipate, 128
- canopy_dissipation_halftime, 123
- canopy_dissipation_rate, 123
- canopy_dissipation_rate_coefficient, 123
- canopy_ea, 110
- canopy_ep, 110
- canopy_harvest, 128
- canopy_in, 128
- canopy_out, 128
- canopy_storage, 125
- canopy_washoff_coefficient, 124
- canopy_water_below, 110
- canopy_water_bypass, 110
- canopy_water_capacity, 110
- canopy_water_in, 110
- canopy_water_out, 110
- canopy_water_storage, 108
- canopy_water_temperature, 110
- CanopyAss, 507
- CanopySimple, 500
- CanopyStandard, 496
- CanopyTemperature, 111
- canvas, 182
- capacity, 487
- cattle_manure, 98
- cattle_slurry, 96
- CCrop, 507
- cd, 305, 307, 308, 415
- CDead, 507
- Celeriac, 155
- Celeriac - transplanted, 156
- CH2OPool, 506
- CH2OReleaseRate, 505
- check, 144
- check_hour, 143
- checkpoint, 233
- chemfid, 246
- chemical, 83, 121, 125, 246
- Chemistry, 139
- chemistry, 133
- child, 266
- ci_middel, 292
- ci_vector, 292
- cite, 47, 49, 59, 61, 71, 72, 94, 102, 123, 133, 137, 153, 165, 167, 169, 171, 175, 177, 181, 199, 215, 231, 233, 234, 267, 276, 285, 291, 299, 302, 309, 313, 319, 321, 333, 341, 350, 365, 373, 379, 385, 401, 411, 439, 447, 463, 475
- classes, 395
- clauses, 63, 229, 270, 377
- clay, 201, 202, 204–210
- ClayOM, 53, 275
- CLeaf, 507
- cloudiness, 450
- CM, 325
- cm, 415
- cm⁻¹, 415
- cm⁻², 415
- cm², 416
- cm³, 416
- cm³ H₂O/cm³, 416
- cm³/cm³, 416
- cm³/g, 416
- cm³/ng, 416
- cm/d, 415
- cm/h, 415
- CO₂, 281, 482
- CO₂_fast, 282
- CO₂_threshold, 276
- coarse_sand, 202, 204, 206, 208, 210, 211
- coarse_silt, 203
- colfid, 237
- colgen, 322
- colgen_Jarvis99, 323
- colgen_Morgan98, 324
- colgen_Styczen88, 325
- colloid, 322, 323, 328, 330
- column, 87, 137, 183, 237, 267, 302, 367, 483

- combine, 75, 134, 370, 458, 465
- command_file, 308, 309
- common, 130, 181, 191
- component, 307
- cond, 63, 229, 377
- condedge, 141
- condition, 63, 85, 143, 518
- conductivity, 487
- Coniferous, 443
- const, 163, 230, 267, 436, 454
- constituents, 73
- content, 352, 381
- convection, 404
- convert, 266
- converted, 327
- converted_fast, 327
- converted_redox, 327
- Cosby_et_al, 219
- cover, 393, 439
- cpedata, 305
- cracks, 345
- creation, 93
- CrLeafCnc, 510
- CrNCnt, 512
- CRoot, 507
- crop, 64, 65, 70, 75, 80, 82, 93, 144, 145, 153, 250, 483
- CROP-FAST, 51
- CROP-SLOW, 51
- crop_depth, 304
- crop_dm_over, 144
- crop_dm_sorg_over, 144
- crop_ds_after, 145
- crop_ea, 111
- crop_ea_soil, 111
- crop_ea_svat, 111
- crop_ep, 110
- crop_uptake_reflection_factor, 123
- crop_width, 304
- cropfid, 250
- crops, 440
- CrpN, 154, 509
- CrRootCnc, 511
- CrSOrgCnc, 510
- CrStemCnc, 510
- cs_vector, 292
- CSOrg, 507
- CStem, 507
- cut, 75
- cut_delay, 503
- cut_stress, 503
- cxspare, 363
- D, 323
- d, 416
- d^{-1} , 416
- daily, 145
- daily_air_temperature, 449
- daily_air_temperature_above, 145
- daily_cloudiness, 450
- daily_extraterrastial_radiation, 450
- daily_global_radiation, 450
- daily_max_air_temperature, 449
- daily_min_air_temperature, 449
- daily_precipitation_above, 145
- DailyNetRoot, 507
- DailyNetShoot, 507
- Daisy, 78, 301
- DAP, 287
- DAPEff, 298
- data, 379, 380
- date, 64, 65
- day, 64–70, 86, 147, 161, 305
- day_length, 287, 450
- days, 76, 86, 146, 477
- dds, 323
- DDT, 131
- ddt, 396
- deactivate_steps, 396
- Dead, 502
- dead_C, 484
- dead_DM, 483
- dead_N, 483
- DeadNLeaf, 508
- DeadNRoot, 509
- DeadWLeaf, 508
- DeadWRoot, 509
- debug, 114, 197, 306, 314, 336, 403, 434, 435
- debug_cell, 176
- debug_equations, 281
- debug_rows, 281
- debug_to_screen, 281
- decompose_CO2_factor, 124
- decompose_conc_factor, 124
- decompose_depth_factor, 124
- decompose_halftime, 124
- decompose_heat_factor, 124
- decompose_lag_increment, 124
- decompose_products, 125
- decompose_rate, 124
- decompose_water_factor, 124
- deep, 192
- DeepPercolation, 195
- default, 55, 57, 107, 121, 133, 138, 153, 167, 210, 253, 255, 275, 288, 317, 325, 407, 445, 456
- default_hour, 367, 369

- defined_until_ds, 290
- delta, 373
- denitrification, 326
- dens_dim, 314
- DensIgnore, 335, 336
- Density, 495
- density, 113, 334, 335, 345, 353, 357
- density_x, 357
- density_y, 357
- density_z, 357
- DensRtTip, 314, 335–338
- DepDry, 453, 480
- DepDryNH4, 453, 480
- deposit, 128, 450
- Deposition, 453, 480
- Depth, 338, 495
- depth, 79, 81, 83, 101, 163, 267
- Depth_Depth, 335
- depth_K, 267
- depth_Theta, 268
- DepWetNH4, 453, 480
- description, 47, 49, 59, 61, 71, 72, 94, 102, 123, 133, 137, 153, 165, 167, 169, 171, 175, 177, 181, 199, 215, 231, 233, 234, 267, 276, 285, 291, 299, 302, 308, 313, 319, 321, 333, 341, 350, 365, 373, 379, 385, 401, 411, 439, 447, 463, 475
- desorption_rate, 321
- DetentionCapacity, 491
- Devel, 154
- device, 182
- dg, 429
- dg C, 430
- dg F, 430
- dgEast, 429
- dgNorth, 429
- dgSouth, 429
- dgWest, 429
- DH, 325
- diameter, 113
- DiffRad, 452, 479
- diffuse_radiation, 450
- diffusion_coefficient, 124, 512
- difrad, 108, 165
- difrad0, 111
- dim, 266, 308
- dim_line, 85, 184, 187, 306, 309, 314, 366, 368, 456, 464, 467, 472
- dimension, 178, 183, 186, 266, 268, 349, 368, 371, 377, 466, 471
- DIN3, 202
- DIN5, 203
- direct_rain, 109
- directory, 305, 516
- disk_harrowing, 79
- dispersivity, 492
- dispersivity_transversal, 492
- dist, 101, 280
- distribution, 253, 482
- div, 230
- DM, 67, 482
- DM_fraction, 348
- DM_per_LAI, 442
- Do, 374
- do, 80, 81
- docmodel, 307
- document, 307
- documentation, 350
- DOM, 512
- dom, 278
- DOM-Element, 515
- dom_pool, 167
- domain, 271
- domsorp, 167, 275
- done, 65, 66
- DPF, 165
- DptEmr, 494
- Drain, 139
- drain, 114, 169
- drain_secondary, 125
- draineqd, 171
- DrainFlow, 170, 195
- drainpoints, 258
- droplet_diameter, 325
- dry, 285
- dry soil fraction, 429
- dry_bulk_density, 200
- dry_matter_fraction, 95
- DS, 66, 287, 303
- ds, 145
- DS_cut_fixate, 511
- DS_Depth, 334
- DS_Emr, 289
- DS_fixate, 511
- DS_Rel, 334
- DS_start_fixate, 511
- DSEff, 297
- DSLAI05, 347
- DSLIm, 445
- DSMature, 290
- DSmax, 503
- DSnew, 503
- DSRate1, 289
- DSRate2, 289
- DSRepeat, 290
- DSSetBack, 290

- dt, 78, 127, 303, 488
- dt1, 388
- dt2, 388
- dtcta, 389
- dtcta_star, 389
- dtltc, 389
- dtltc_star, 390
- dtstc, 389
- dtstc_star, 390
- dx, 313
- dz, 313
- dZs, 485
- E, 324, 325
- e.a, 392
- e.abs, 391
- e.c, 392
- e.c.abs, 389
- E_Leaf, 505
- e.pa, 391
- E_Root, 505
- e.sat.air, 392
- e.slabs, 389
- E_SMB, 53
- E_SOrg, 505
- E_Stem, 505
- E_trans, 393
- Ea_Gamma, 292
- Ea_Jm, 295
- Ea_kc, 295
- Ea_ko, 295
- Ea_rd, 295
- Ea_Vm, 295
- Early potato, 158
- Early white cabbage - transplanted, 160
- EconomicYield_N, 503
- EconomicYield_W, 503
- Eda_Vm, 295
- efficiency, 50, 60, 62, 279, 513
- electric_current, 412
- element, 173
- Elevation, 452, 479
- else, 75, 147, 269
- emerge, 70, 84
- empty, 120
- EmrSMF, 289
- EmrThrs, 288
- EmrTSum, 288, 289
- enable_boundary_diffusion, 403
- enable_fertilization, 85
- enable_ice, 486
- enable_irrigation, 85
- enable_K_macro, 221
- enable_N_stress, 154
- enable_Theta_macro, 221
- End, 454, 481
- end, 66, 94, 122, 126, 127, 185, 187, 272, 277, 279, 454, 459, 467, 469, 486, 488, 491
- end_time, 86
- EndDSRelRtRes, 506
- entries, 235, 334, 335, 341
- entry, 458
- env_lai_factor, 390
- EPext, 497, 501
- EpFac, 498, 501
- EpFacDS, 498, 501
- EpFactor, 490
- EpFactorDry, 440
- EpFactorWet, 440
- EpFacWet, 498, 501
- EpInterchange, 439
- Eps, 491
- epsilon, 311
- epsilon_0, 261
- Ept, 496
- eq_depth, 170
- EqDrnFlow, 170, 195
- equilibrium, 175, 321, 327, 401
- equivalent_weight, 73
- error, 70
- EUROSEM, 299, 319
- EvapSnowPack, 486
- EvapSoilSurface, 491
- event, 517
- every, 145
- every_hour, 306
- exchange, 177, 302, 341
- ExfoliationFac, 505
- exp, 268, 339
- exponent, 271
- expr, 47, 48, 144, 146, 270, 311, 339, 349, 366, 370, 473
- extern, 71, 146, 163, 192, 235
- extern_fertigation, 71
- extern_subsoil, 72
- extinction_coefficient, 232
- extra, 182, 308
- f, 226
- f1_dolman, 390
- f3const, 388
- f_1, 390
- f_2, 390
- f_3, 390
- f_4, 390
- f_c, 485
- f_def, 390

- f_letop, 390
- f_photo, 339, 340
- F_primary, 329
- F_secondary, 329
- f_SMB1, 53
- f_temp, 390
- f_theta, 390
- fa, 165
- faar_geder_dybstroelse, 96
- factor, 53, 54, 186, 305, 350, 368, 412, 429, 430, 517
- false, 117, 146
- FAO3, 207
- FAO7, 208
- FAO_PM, 285
- Farquhar, 291
- FC_C3, 293
- FC_C4, 295
- fc_primary, 329
- fc_secondary, 329
- fertilize, 66, 74
- fertilize_at, 67
- fertilize_at_index, 68
- fertilize_base, 73
- fertilize_incorporate, 68
- fertilize_index, 67
- fertilize_rest, 67
- fertilize_rest_index, 67
- fertilize_year, 67
- fertilized_C, 281
- fertilized_N, 281
- fertilizer, 85
- Fetch, 514
- fetch, 268, 383
- FetchPretty, 517
- fg/L, 416
- Field chemical, 247
- Field crop nitrogen balance, 381
- Field mineral nitrogen balance, 381
- Field nitrogen, 238
- Field nitrogen balance, 381
- Field organic nitrogen balance, 382
- Field water, 240
- Field water balance, 382
- field_separator, 237
- file, 84, 164, 183, 186, 192, 283, 305, 308, 312, 366, 368, 455, 457, 463, 466, 471
- filter, 85, 184, 187, 306, 309, 313, 366, 368, 456, 464, 467, 471
- filter_velocity, 328
- fine_sand, 202, 204, 205, 207, 209, 210
- fine_silt, 203
- finished, 146
- finite, 119
- first_year_utilization, 95, 140
- fixate_factor, 511
- Fixated, 512
- fixed, 192
- flow, 353
- flow_back, 353
- flow_bottom, 353
- flow_front, 354
- flow_left, 354
- flow_right, 354
- flow_top, 355
- flush, 237
- flux, 68, 69, 72, 76, 77, 84, 193, 349, 466, 518
- flux_bottom, 354
- flux_top, 355
- Fm, 297
- Fodder Beet, 156
- forced, 340
- forced_flux, 490
- forced_LAI, 161
- forced_pressure, 490
- forced_T, 436
- ForcedCAI, 500
- ForcedLAI, 440
- format, 179, 307
- Foulum, 140
- Foulum AP, 213
- Foulum B, 213
- Foulum C, 213
- Foulum_slurry, 95
- fraction, 125
- fraction_total, 293
- fractions, 50, 60, 62, 214, 279, 513
- Freundlich, 89
- Froegraes, 156
- from, 68, 69, 72–74, 77, 84, 149, 150, 235, 355, 358, 360, 458, 460
- front, 448, 476
- fsa, 485
- full, 90, 119, 461
- fungicide, 130
- G, 286
- g, 416
- g C/cm², 417
- g C/cm³, 417
- g C/m², 417
- g DM/cm², 417
- g DM/m², 417
- g N/cm², 417
- g N/cm²/h, 417
- g N/cm³, 417

- g N/l, 417
- g N/m², 417
- g N/m²/h, 418
- g w.w./m², 418
- g/cm², 418
- g/cm²/h, 418
- g/cm²/mm, 418
- g/cm³, 418
- g/cm/h, 418
- g/ha, 418
- g/ha/d, 418
- g/ha/h, 418
- g/ha/mm, 419
- g/kg, 419
- g/m², 419
- g/m²/h, 419
- g/mol, 419
- g_a, 392
- g_H_s_c, 392
- g_H_shadow_c, 392
- g_H_sun_c, 392
- g_W_shadow_c, 392
- G-W_sun_c, 392
- g_W_sun_c, 392
- Gamma, 292
- gamma, 392
- Gamma25, 291
- gb_W_shadow, 392
- gb_W_sun, 392
- gbw, 293
- geometric, 105, 141
- Geometry, 258, 259, 304
- Geometry1D, 515
- GeometryRect, 515
- Gerwitz+Page74, 337
- get, 268
- gflux, 389
- GL, 296
- global_radiation, 449
- GlobRad, 451, 478
- gnuplot, 181, 307, 312
- goal_A, 175, 176
- GP1D, 335
- GP2D, 303, 336
- Graes, 156
- graph, 189, 308
- Grass, 156, 443
- Grass to grain, 156
- Green Pea, 157
- Groundwater, 139
- groundwater, 191
- growth, 337
- GrowthRespiration, 508
- gs, 293
- gs_ms, 293
- gs_vector, 292
- H, 294
- h, 216–219, 221, 223, 224, 227, 267, 268, 279, 359, 419, 487, 493
- h⁻¹, 419
- H2OExtraction, 496
- H2OUpt, 496
- h3_soil, 115
- h_aquifer, 191, 194
- h_b, 216, 217, 220–223, 227
- h_bottom, 115
- H_c_a, 393
- h_fc, 433
- h_frozen, 486
- h_ice, 359, 489
- h_lim, 141, 346
- h_m, 226
- H_shadow, 393
- H_soil, 393
- H_sun, 393
- h_wp, 495
- h_x, 495
- ha, 388, 419
- Hairsine91, 299
- handle, 306, 351, 366, 369
- Hansen, 403
- Hardwood, 443
- Hargreaves, 286
- harmonic, 105, 141
- Harvest, 153, 482
- harvest, 75, 84, 233, 301
- harvest_annual, 65
- harvest_base, 74
- harvest_C, 139
- harvest_DM, 139
- harvest_N, 139
- harvest_perennial, 66
- Harvesting, 502
- heat, 258
- heat_factor, 49, 59, 61, 263, 264, 276, 326, 512
- heatrect, 197
- Height, 109, 441, 497, 500
- height, 81, 144, 149, 150, 170, 191, 194, 352, 439
- height_end, 113, 253
- height_max, 161
- height_start, 113, 253
- Heptachlor, 131
- herbicide, 130
- hest_dybstroelse, 96
- hl, 389

- hmovie, 308
- hoens_dybstroelse, 96
- hoens_gylle, 96
- hoens_staldgoedning, 96
- HorHeat, 199, 514
- horizon, 199, 268, 271, 311, 491, 492
- horizons, 492
- horizontal, 437
- horse_manure, 98
- hour, 65–67, 69, 70, 86, 146, 147, 513
- hourly, 146
- hours, 76, 86, 146, 419, 477
- hPa, 419
- hs, 389
- hs_vector, 292
- humus, 201, 203–206, 208–211, 214, 281
- HvsDS, 499
- HvsLAI, 439
- HvsWStem, 499
- hydraulic, 199, 215, 310
- hypostomatous, 391
- hypres, 226
- I, 493
- L_bottom, 493
- L_wall, 493
- identifier, 270
- identity, 266, 377
- idso_jackson, 262
- if, 75, 147, 269
- ignore, 134
- Ikuwala Maize, 156
- immobile, 322, 329
- incoming_Long_radiation, 112
- incoming_NIR_radiation, 112
- incoming_PAR_radiation, 112
- incoming_Total_radiation, 112
- incorporate_fertilizer, 74
- IncWLeaf, 508
- IncWRoot, 508
- IncWSOrg, 508
- IncWStem, 508
- index, 334, 335, 355, 356
- indirect, 459
- infiltration, 113
- init, 278
- InitCAI, 348
- initial, 94, 123
- initial_C, 122, 269
- initial_C_per_N, 50, 60, 62
- initial_C_primary, 126
- initial_C_secondary, 125
- initial_fraction, 50
- initial_h, 488
- initial_M, 126
- initial_M_primary, 127
- initial_M_secondary, 126
- initial_Ms, 127
- initial_NH4, 269
- initial_NO3, 270
- initial_SOM, 277
- initial_T, 486
- initial_table, 192
- initial_Theta, 488
- initial_value, 164
- initial_weight, 348
- initial_zero, 267
- initialized, 93
- inline, 468
- input, 280, 381, 516
- insecticide, 131
- install_directory, 516
- IntcpCap, 498, 501
- IntDSRelRtRes, 506
- integer, 229
- IntegerCondClause, 518
- interception_capacity, 231, 440
- interesting_content, 351
- internal_ponding, 493
- interval, 358
- intervals, 311, 315, 514
- Ioxynil, 131
- IPU, 131
- IRb0, 317
- IRd0, 317
- irrigate_base, 76
- irrigate_overhead, 76
- irrigate_subsoil, 76
- irrigate_surface, 77
- Irrigation, 138, 517
- irrigation, 68
- irrigation_delay, 70
- irrigation_overhead, 109
- irrigation_overhead_temperature, 109
- irrigation_rest, 69
- irrigation_subsoil, 109
- irrigation_subsoil_permanent, 108
- irrigation_surface, 109
- irrigation_surface_temperature, 109
- irrigation_total, 109
- irrigation_year, 70
- ISSS3, 204
- ISSS4, 205
- iterations, 115
- J, 413
- J/cm²/h, 413
- J_matrix, 130, 515

- J_primary, 130
- J_secondary, 130
- J_tertiary, 130, 515
- jm, 293
- Jm_vector, 292
- Jynde vad, 140
- Jynde vad Ap, 213
- Jynde vad C, 213
- K, 89, 175, 176, 216–219, 221, 223, 224, 227, 228, 326, 346, 413, 488
- k, 263, 264, 336, 337
- k₁₀, 263, 264
- k_{AB}, 328, 401
- K_aquitard, 191, 194
- K_{at_h}, 216–219, 221, 223, 224, 227
- K_{average}, 432, 434, 436
- K_b, 217, 220, 222
- k_{BA}, 328, 401
- K_{clay}, 89, 90, 330
- K_d, 330
- k_{desorption}, 330
- K_{fast}, 327
- k_{half}, 481
- K_{ice}, 514
- K_{NH4}, 276
- K_{NO3}, 276
- K_{OC}, 89, 90, 330
- k_{qb}, 58
- k_{qr}, 58
- K_{sat}, 216–224, 226, 227
- K_{snow_factor}, 485
- k_{sorption}, 330
- K_{to_pipes}, 170, 194
- K_{wall_relative}, 115
- K_{water}, 514
- kalkuner_dybstroelse, 96
- kartoffelfrugtsaft, 96
- Kc25, 294
- kd, 324
- KE, 324, 325
- Kedge, 436
- key, 200
- kg, 420
- kg C/ha, 420
- kg C/ha/d, 420
- kg C/ha/h, 420
- kg DM/ha, 420
- kg DM/m², 420
- kg m⁻² s⁻¹, 421
- kg N/ha, 420
- kg N/ha/d, 420
- kg N/ha/h, 420
- kg N/ha/mm, 421
- kg N/ha/y, 421
- kg w.w./ha, 421
- kg/ha, 421
- kg/ha/h, 421
- kg/ha/mm, 421
- kg/ha/y, 421
- kg/m, 421
- kg/m², 422
- kg/m²/s, 422
- kg/m³, 422
- kg/m/s, 422
- kg/mol, 422
- kg/Pa, 421
- kgN/ha/year, 422
- KH, 323
- kj, 296
- kn, 339
- Ko25, 294
- komposteret_husholdningsaffald, 97
- kPa, 420
- kr, 324
- kvaeg_ajle, 97
- kvaeg_dybstroelse, 97
- kvaeg_gylle, 97
- kvaeg_staldgoedning, 97
- kvaeggylle_separeret_tyk, 97
- kvaeggylle_separeret_tynd, 97
- kvaeggylle_usepareret, 98
- L, 169, 194
- l, 217, 219, 220, 223, 224, 226
- l/ha, 422
- l/ha/h, 422
- L/kg, 413
- l/kg, 422
- L0, 334, 336, 337
- L00, 337
- L_{ge}, 55
- L_{gm}, 56
- L_{gv}, 55
- L_{gx}, 55
- L_{i0}, 261
- L_{ia}, 261
- L_n, 261
- L_{ym}, 56
- LADm, 500
- lag, 127
- LAI, 292, 347, 391, 439
- LAI_vector, 292
- LAI Dist0, 499
- LAI Dist1, 499
- LAI factor, 442
- LAIvsDAY, 441, 442
- LAIvsDay, 161

- LAIvsH, 439, 500, 501
- LAIvsTS, 161
- lambda, 215, 216, 221, 222, 375, 376, 391
- Langmuir, 89, 175
- Large.RtDR, 505
- last_cut, 503
- last_time, 154
- lateral, 169
- latest, 66
- LaTeX, 179
- Latitude, 452, 479
- layers, 94
- LE_atm, 393
- LE_shadow, 393
- LE_sun, 393
- lea, 389
- Leaf, 502, 509
- leaf, 75, 80
- leaf_C, 484
- leaf_DM, 484
- leaf_N, 484
- leaf_width, 498, 501
- LeafAI, 499
- LeafAIMod, 498
- LeafGrowthRespiration, 508
- LeafMaintRespiration, 508
- LeafPhot, 154
- leafPhotN, 293
- LeafRespiration, 508
- left, 184, 186, 448, 476
- legend, 182
- lel, 389
- length, 411
- let, 270
- Leuning, 374
- LfDR, 505
- LfRtRelRtRes, 506
- library, 351
- limit, 311
- limits, 214
- linear, 90, 176, 197, 399
- Litter, 139
- litter, 231
- litter_am, 442
- litter_decompose, 128
- litter_ea, 110
- litter_ep, 110
- litter_in, 128
- litter_out, 128
- litter_storage, 125
- litter_tillage, 128
- litter_water_capacity, 110
- litter_water_in, 110
- litter_water_out, 110
- litter_water_storage, 108
- litter_water_temperature, 110
- ln, 270
- LocalDetentionCapacity, 491
- location, 137
- lock, 93
- log, 233
- log10, 270
- log_prefix, 303
- log_time_columns, 302
- Longitude, 452, 479
- loop, 469
- loss, 66
- lr, 433
- luminous_intensity, 412
- lysimeter, 193
- M, 126, 515
- m, 89, 374–376, 422
- m s^{-1} , 423
- m^{-1} , 423
- m^{-2} , 423
- $\text{m}^{-2} \text{ kg s}^{-1}$, 423
- m^2 , 423
- m^3 , 423
- m^3/cm^3 , 423
- m^3/kg , 423
- m/s , 423
- m1, 485
- m2, 485
- M2B, 113
- M_BaC, 221
- M_BaC_Bimodal, 222
- M_C, 222
- M_error, 129
- M_intervals, 228
- M_primary, 129
- M_Q, 399
- M_secondary, 129
- M_vG, 223
- M_vG_compact, 224
- M_vGp, 225
- MA, 325
- MACRO, 220
- macro, 253, 397
- mactrans, 255, 397
- maintenance, 60
- MaintRespiration, 508
- Maize, 156, 232
- makkink, 286
- manager, 303
- manure, 98
- map, 78
- MARKVAND, 78

- markvand, 77
- mass, 173, 411
- matrix, 114
- matrix_solid, 257
- matrix_solute, 257
- matrix_water, 258, 259
- max, 183, 230, 270, 375, 376
- max_absolute_difference, 114, 431, 434, 436
- max_exfiltration_gradient, 487
- max_iter, 312
- max_iteration, 391
- max_iterations, 114, 431, 434, 436
- max_iterations_timestep_reduction_factor, 436
- max_number_of_small_time_steps, 434, 436
- max_pressure_potential, 436
- max_rain, 449
- max_relative_difference, 115, 431, 434, 436
- max_sink_change, 487
- max_sink_secondary, 134
- max_sink_solute, 134
- max_sink_total, 134
- max_svat_absolute_difference, 108
- max_svat_iterations, 108
- max_time_step_reductions, 431, 434, 435
- maxEdiff, 108
- MaxPen, 494
- MaxRootingDepth, 492
- MaxTDay, 452, 479
- maxTdiff, 108
- MaxWidth, 495
- MCP, 131
- mday, 147, 513
- measure, 379, 380
- measured_file, 304
- medium_sand, 202, 204, 208
- medium_silt, 203
- merge, 370, 470
- message, 63, 71, 78, 79, 86
- mf, 485
- mg, 423
- Mg DM/ha, 414
- Mg DM/ha/h, 414
- mg N/kg dry soil, 424
- mg N/l, 424
- Mg w.w./ha, 414
- mg/g, 424
- Mg/ha, 414
- mg/ha, 424
- mg/ha/d, 424
- mg/ha/h, 424
- mg/L, 424
- mg/l, 424
- mg/m², 424
- microsecond, 513
- microseconds, 146, 477
- middle, 83
- Millet, 232
- min, 183, 230, 270, 375
- min_AM_C, 278
- min_AM_N, 278
- min_B, 176
- min_dist, 315
- min_iter, 311
- min_light_fraction, 154
- min_PAR, 291
- min_pressure_potential, 436
- min_root_crop, 358
- min_root_density, 357
- min_sink_total, 134
- min_sun_angle, 109
- MinDens, 337
- mineral, 101
- mineral_form_factor, 514
- minimal_timestep, 303
- minimize, 311
- minimum_weight, 73
- mink_raev_gylle, 98
- mink_raev_staldgoedning, 98
- minute, 147, 513
- minutely, 147
- minutes, 145, 424, 477
- missing, 85, 184, 186, 305, 309, 313, 366, 368, 455, 464, 467, 471
- missing_value, 237, 309
- missing_years, 457
- MIT3, 202
- MIT7, 203
- mix, 79
- MJ/d/m², 413
- MJ/h/m², 413
- MJ/m²/d, 413
- MJ/m²/h, 413
- ml/g, 425
- mm, 425
- mm⁻¹, 425
- mm/d, 425
- mm/h, 425
- mm/s, 425
- mm_dd, 148
- mm_dd_base, 147
- Mmax, 323
- Mmax_tillage_factor, 323
- mmol/kg, 425
- mmol/m², 425
- mmol/m²/s, 425
- MNA, 374

- mobile, 329
- mod, 230
- mod_alpha, 225
- mod_C, 227
- mod_K_sat, 225
- mod_n, 225
- model, 351
- models, 307
- mol, 425
- mol/kg, 426
- mol/m², 426
- mol/m²/s, 426
- MolenWesseling, 171
- Mollerup, 197, 403, 435
- month, 64–70, 86, 147, 148, 513
- monthly, 148
- Moody, 171
- Movement, 138
- movement, 257
- MPa, 414
- MPa⁻¹, 414
- MR, 325
- mrprime, 485
- Ms, 127, 323
- msg_number_of_small_time_steps, 434, 436
- mtprime, 485
- multi, 134, 189, 343, 349
- MV_Crop, 55
- MV_Soil, 57
- MxNH4Up, 495
- MxNO3Up, 495
- my_max, 175
- my_max_clay, 89
- my_max_OC, 90

- N, 49, 59, 61, 131, 133, 277, 512
- n, 219, 220, 224, 226
- N-dist, 292
- N25S, 102
- N2O, 329
- N2O_fraction, 263
- N_actual, 162, 441
- N_added, 482
- N_AM, 507
- N_b, 162
- N_balance, 382
- N_demand, 162, 442
- N_flowering, 162
- N_fraction, 348
- N_litter, 442
- n_ma, 221
- N_per_LAI, 442
- N_removed, 482
- N_uptake, 442

- name, 64, 76, 93, 113, 114, 177, 236, 268, 310, 343, 396, 407, 409, 450, 517
- NCrop, 507
- NDead, 507
- ndif, 386
- negate, 351
- net_radiation, 108, 261
- netlong_brunt, 388
- NetPhotosynthesis, 507
- netrad_brunt, 388
- new dg, 430
- next, 146
- next_large, 303
- NfLeafCnc, 510
- NfNCnt, 512
- NfRootCnc, 511
- NfSOrgCnc, 510
- NfStemCnc, 510
- ng/cm³, 426
- ng/kPa, 426
- ng/L, 426
- ng/l, 426
- ng/mm/h, 426
- ng/MPa, 426
- NH4, 71, 90, 132, 329
- NH4_fraction, 95, 102
- NH4_root_min, 511
- NH4_source, 281
- NH4DryDep, 453, 480
- NH4Extraction, 496
- NH4Upt, 496
- NH4WetDep, 453, 479
- nil, 79
- NIRext, 497, 501
- NIRref, 497, 501
- Nitrate, 103
- Nitrification, 200
- nitrification, 263, 329
- nitrogen_stress, 512
- nitrogen_stress_days, 483, 511
- nitrogen_stress_limit, 509
- NLeaf, 506
- Nleaf_vector, 292
- NO3, 71, 132, 329
- NO3_fraction, 95
- NO3_root_min, 511
- NO3_source, 281
- NO3DryDep, 453, 480
- NO3Extraction, 496
- NO3Upt, 496
- NO3WetDep, 453, 480
- none, 47, 90, 119, 135, 170, 171, 197, 231, 254, 282, 299, 346, 363,

- 393, 396, 404, 405, 426, 431, 433, 445, 454, 461
- NoOfIntervals, 108
- normalize, 201, 203–206, 208–210, 214
- not, 117, 148
- Note, 453, 480
- novogro-30_slam, 98
- novogro_slam, 98
- NP, 102
- NPK01, 102
- NPK02, 103
- NRoot, 506
- NSOrg, 506
- NStem, 506
- nu.1, 388
- nu.2, 388
- nu.3, 388
- null, 343
- number, 177, 265, 356, 377
- numbers, 117, 235
- numeric, 214
- nutrient, 131, 135
- Offset, 499
- offset, 192, 193, 350, 368, 430, 460
- old, 54, 397
- om, 93
- Onion, 156
- Onion - planting of sets, 157
- operand, 148, 151, 230, 268, 270–272
- operands, 117, 118, 144, 148, 229, 230, 265, 270
- operation, 68
- or, 118, 148
- organic, 94, 275
- OrganicMatter, 139
- origin, 306
- original, 85, 184, 187, 306, 309, 313, 366, 368, 456, 464, 467, 472
- Osvaldo, 304
- outer, 476
- output, 302, 381
- output_base, 310
- output_width, 310
- overflow_warn, 433
- overhead, 72
- P, 324
- Pa, 414
- Pa^{-1} , 414
- paab, 296
- panic, 79
- PAR_, 293
- par_file_prefix, 304
- par_file_suffix, 304
- par_width, 304
- parameter, 311
- parameter_names, 235
- PARext, 497, 500
- Park82, 299
- PARref, 497, 500
- PARrel, 499
- pars, 304
- parser, 283
- part, 93
- partial, 461
- partial_day, 495
- partial_day_length, 287
- partial_soil_temperature, 495
- Partit, 154
- Partition, 509
- path, 350, 516
- PAverage, 453, 480
- Pcb_NIR, 318
- Pcb_PAR, 318
- Pea, 157
- PenClayFac, 494
- Pendimethalin, 131
- penetration, 79
- PenPar1, 494
- PenPar2, 494
- period, 149, 383
- periodic, 149
- permanent, 231, 441
- pesticide, 130
- pesticides, 133
- pet, 107, 285
- pF, 430
- pg/L, 427
- Ph_NIR, 318
- Ph_PAR, 318
- phenology, 287
- PhotEff1, 290
- photosynthesis, 291
- pig_manure, 98
- pig_slurry, 99
- Pioneer Maize, 157
- pipe, 193
- pipe_level, 170
- pipe_outlet, 170
- pipe_position, 114, 170, 194
- plant, 82
- plant_distance, 82
- plant_position, 82
- PLF, 163
- plf, 271
- plowing, 83
- pluck, 80

PM, 286
 PMSW, 385
 pn_vector, 292
 points, 271, 468
 polygon, 476
 pond, 491
 pond_ea, 110
 pond_ep, 110
 pond_max, 254, 395
 pond_section, 490
 pond_damp, 299, 322
 porosity, 81
 pos_dim, 314, 467, 472
 Potato, 158
 Potato; Agria, 157
 Potato; FertOrgaNic, 157
 Potato; Folva, 157
 Potato; Koege, 157
 Potato; SCRI, 157
 Potato; SCRI - Aarhus, 158
 Potato; Triada, 158
 PotCanopyAss, 507
 potential, 69, 70, 150, 327
 potential_evapotranspiration_dry, 286
 potential_evapotranspiration_wet, 286
 potential_fast, 327
 potential_N, 162
 PotRtDpt, 495
 pow, 271
 ppm, 427
 ppm dry soil, 430
 prata, 262
 prec, 391
 Precip, 452, 478
 PrecipCorrect, 453, 480
 precipitation, 145, 449
 PrecipScale, 454, 480
 precision, 73, 378, 381, 383
 pressesoft_groentpillefabrik, 99
 pressure, 141, 346
 pressure_barrier, 396
 pressure_end, 253, 395
 pressure_initiate, 253, 395
 pressure_limit, 395
 pressure_table, 191, 194
 previous, 303
 primary, 65, 328
 print_C, 234
 print_data, 380
 print_dimension, 234, 236
 print_header, 234, 236
 print_initial, 235
 print_N, 234
 print_parameterizations, 307
 print_sum, 383
 print_tags, 234, 236
 print_time, 303
 Prod, 154
 Production, 504
 production_delay, 503
 production_stress, 111, 496
 profile, 182, 471
 progn, 80
 program, 301
 progress, 405
 Ps_NIR, 317
 Ps_PAR, 317
 Psa, 485
 Pscb_NIR, 318
 Pscb_PAR, 318
 Pscd_NIR, 318
 Pscd_PAR, 318
 pstress, 390
 PtLeafCnc, 510
 PtNCnt, 512
 PtRootCnc, 511
 PtSOrgCnc, 510
 PtStemCnc, 510
 q, 333, 487, 489
 Q10k, 296
 Q10rd, 296
 Q10vm, 296
 q-p, 489
 q-primary, 489
 q-s, 486
 q-secondary, 489
 q-x, 437
 q-z, 437
 Qeff, 297
 quartz_form_factor, 514
 r_a, 388
 r_aa, 388
 r_aastab1, 389
 r_aastab2, 389
 R_abs_shadow, 393
 R_abs_soil, 392
 R_abs_sun, 393
 r_ac, 389
 r_as, 389
 r_astab, 388
 R_bottom, 493
 R_crust, 493
 R_eq_abs_shadow, 393
 R_eq_abs_soil, 393
 R_eq_abs_sun, 393
 r_Leaf, 505

- R_max, 481
- R_mixing, 491
- r_pore_min, 201
- r_Root, 505
- r_sc, 390
- r_sc_1, 389
- r_sc_2, 389
- r_sc_js, 390
- r_SOrg, 505
- r_Stem, 505
- R_wall, 493
- Rad, 495
- rad, 169, 429
- raddist, 108, 317
- rain, 450
- rainergy, 319, 323, 324
- range, 271
- rate, 348
- raw, 407, 409
- rb, 285, 286
- rcmax, 387
- rcmin_const, 387
- rcmin_star, 390
- reaction, 133, 321
- record_separator, 237
- rectangle, 257
- redox_height, 327
- ref_alpha, 225
- ref_K_sat, 225
- ref_n, 225
- reference_evapotranspiration, 285, 449
- reference_evapotranspiration_wet, 286
- RefEvap, 452, 479
- relative_humidity, 450
- release, 322, 348
- RelHum, 452, 479
- RelRateRtRes, 506
- remaining_time, 76
- ReMobilDS, 504
- ReMobilRt, 504
- remove_residuals, 66
- repeat, 80
- RepThrs, 288
- RepTSum, 288
- require_top, 381
- Res, 293
- reserve, 459
- reserved, 155
- ReservoirConstant, 491
- reset_mday, 143
- reset_month, 143
- reset_offset, 369
- residuals_C_root, 140
- residuals_C_soil, 140
- residuals_C_top, 140
- residuals_DM, 139
- residuals_N_root, 140
- residuals_N_soil, 140
- residuals_N_top, 140
- residue, 232
- Respiration, 507
- respiration, 482
- rho_1, 485
- rho_2, 485
- rho_a, 392
- rho_s, 485
- richards, 431, 433
- Ridge, 493
- ridge, 80, 81, 490
- right, 184, 186, 448, 476
- Rn, 286
- Root, 154, 161, 441, 502, 509
- root, 47, 101, 279
- root_am, 162
- root_DM, 162, 441
- root_N, 162
- rootdens, 333, 494
- RootGrowthRespiration, 508
- RootMaintRespiration, 508
- rootmatch, 312
- RootRespiration, 508
- RootSystem, 494
- rotavation, 79
- row_distance, 314, 336, 338
- row_position, 82, 303, 313, 336, 338
- row_width, 82, 303
- rs_max, 498, 501
- rs_min, 498, 501
- Rsqr, 379
- RsqrW, 380
- RSR, 509
- RtDR, 505
- RtDR_T_factor, 505
- rubiscoNdist, 339
- Rug, 158
- run, 305, 311, 516
- running, 149
- runoff, 491
- Rxylem, 495
- Rye, 158
- Ryegrass, 158
- Ryegrass-SLOW, 51
- S, 113, 170, 195, 294, 486, 515
- s, 391, 427
- s⁻¹, 427
- S_AB, 326, 328
- S_C, 167

- S_decompose, 129
- S_decompose_primary, 129
- S_decompose_secondary, 129
- S_drain, 129, 488, 515
- S_exchange, 129
- S_external, 129
- S_F, 55
- S_forward_sink, 489
- S_forward_total, 489
- S_ice, 489
- S_ice_water, 489
- S_incorp, 489
- S_Le, 56
- S_Lm, 56
- S_Lr, 56
- S_Lx, 56
- S_N, 167
- S_p, 254, 489, 515
- S_permanent, 127, 487
- S_primary, 129
- S_root, 129, 489
- S_secondary, 129
- S_sorption, 331
- S_sorption_primary, 331
- S_sorption_secondary, 331
- S_sum, 489
- S_tertiary, 129
- S_transform, 130
- samples, 184
- sand, 201, 205, 206, 209, 210
- satterlund, 262
- scope, 71, 72, 138, 146, 302, 311, 341
- scopesel, 343
- ScreenHeight, 452, 479
- seasons, 67
- second, 147, 513
- second_year_compensation, 73
- second_year_utilization, 95, 139
- secondary, 64, 328, 345
- secondary_domain, 200
- secondly, 149
- seconds, 145, 427, 477
- Seed, 154
- seed, 82, 347
- seed_bed_preparation, 79
- seed_C, 140
- seed_N, 140
- select, 234, 349
- sequence, 81
- set_heat_source, 81
- set_porosity, 81
- set_surface_detention_capacity, 81
- SHA12, 375
- SHA14, 376
- shadow_stomata_conductance, 440
- ShadowLeafTemperature, 111
- ShldResC, 504
- show_data, 314
- show_match, 314
- SI, 411
- SIfactor, 412
- sigma_NIR, 317
- sigma_PAR, 317
- Silage Maize, 156
- silence, 518
- Silomajs, 158
- silt, 201, 204–207, 209, 210
- SimCAI, 500
- simple, 161, 383, 407
- simplex, 311
- sink_cell, 130, 489
- sink_sorbed, 257
- size, 182
- slagtekyllinger_dybstroelse, 99
- slagtesvin_ajle, 99
- slagtesvin_dybstroelse, 99
- slagtesvin_gylle, 99
- slagtesvin_staldgoedning, 99
- slurry, 95
- SMB, 59
- smb, 277
- SMB-FAST, 60
- SMB-SLOW, 60
- smb_tillage_factor, 278
- Snow, 108, 484
- snow, 450
- snow_ea, 109
- snow_ep, 109
- snow_fraction, 449
- snow_in, 128
- snow_out, 128
- snow_storage, 125
- snow_water_in, 109
- snow_water_in_temperature, 109
- snow_water_out, 109
- snow_water_out_temperature, 110
- soer_ajle, 99
- soer_dybstroelse, 100
- soer_gylle, 100
- soer_staldgoedning, 100
- Soil, 138, 492
- soil, 47, 77, 183, 263
- Soil chemical, 249
- Soil mineral-N balance, 382
- Soil N balance, 382
- Soil NH₄⁺ balance, 382
- Soil nitrogen, 241
- Soil NO₃⁻ balance, 382

- Soil organic-N balance, 382
- Soil water, 244
- Soil water balance, 383
- soil_depth, 304
- soil_ea, 110
- soil_enrichment_factor, 331
- soil_ep, 110
- soil_h, 271
- soil_heat_capacity, 269
- soil_heat_conductivity, 269
- soil_inorganic_N_above, 149
- soil_K, 269
- soil_temperature, 495
- soil_temperature_above, 149
- soil_Theta, 269
- soil_water_content_above, 150
- soil_water_pressure_above, 150
- SoilDepth, 315
- SoilHeat, 138, 486
- SoilLayer, 491
- SoilRegion, 492
- SoilWater, 138, 487
- solid, 132
- solubility, 123
- solubility_infiltration_factor, 123
- solute, 76, 114, 257, 263, 321, 330, 361, 517
- solute_failure_level, 257
- solute_infiltration, 113
- solute_mass, 396
- solute_storage, 396
- solver, 197, 363, 391, 403, 435
- SOM, 61
- som, 277
- SOM-FAST, 62
- SOM-FAST-OLD, 62
- SOM-INERT, 62
- SOM-SLOW, 62
- SOM-SLOW-OLD, 62
- SOM_C_per_N, 200
- SOM_fractions, 200
- SOM_limit_lower, 280
- SOM_limit_upper, 281
- SOM_limit_where, 280
- som_pool, 167
- som_tillage_factor, 278
- sorbed, 321, 330
- SOrg, 502
- sorg, 75, 80
- sorg_C, 484
- sorg_DM, 484
- sorg_height, 503
- sorg_N, 484
- SOrgAI, 500
- SOrgAIMod, 498
- SOrgGrowthRespiration, 508
- SOrgMaintRespiration, 508
- SOrgPhotEff, 499
- SOrgRespiration, 508
- sorption, 330
- source, 185, 188, 193, 272, 365, 370, 371, 459, 465, 470, 473
- source_average, 272
- source_increase, 272
- source_sum, 272
- source_unique, 272
- sow, 82, 84
- spar, 388
- spec, 351
- specific_AI, 232
- speed, 482
- spildevandsslam, 100
- SpLAI, 498
- SpLAIfac, 347
- spray, 64, 82, 127
- spray_index, 68
- spring, 161
- Spring Barley, 158
- Spring Barley; Foulum, 159
- Spring Rape, 159
- Spring Wheat, 159
- spring_LAI, 162
- SpRtLength, 314, 333
- SpSOrgAI, 498
- SpStemAI, 499
- sqr, 230, 272
- sqrt, 272
- srad, 391
- Ssnow, 484
- SSOC, 391
- stabilizing_method, 403
- state, 101, 119, 487
- static, 193
- Station, 453, 480
- Stem, 502, 509
- stem, 75, 80
- stem_C, 483
- stem_DM, 483
- stem_N, 483
- StemAI, 500
- StemAIMod, 499
- StemGrowthRespiration, 508
- StemMaintRespiration, 508
- StemPhotEff, 499
- StemRes, 504
- StemRespiration, 508
- step, 470
- Stomatacon, 292

- stomatacon, 373
- stop, 83, 302
- string, 377
- string-equal, 118
- StringerCondClause, 518
- stub, 75
- stubble_cultivation, 79
- style, 367, 369–371, 464, 466, 468, 470, 472, 473
- submodels_and_attribute, 351
- subsoil, 72
- Sugar Beet, 159
- sum_name, 383
- summary, 234, 379
- sun-shade, 317
- sun_LAI_fraction, 111
- sun_LAI_fraction_total, 393
- sun_NIR, 111
- sun_PAR, 111
- SunLeafTemperature, 111
- sunlit, 155
- sunlit_stomata_conductance, 440
- Surface, 138, 453, 480, 490
- surface, 72, 328
- surface_AB, 328
- surface_decompose, 128
- surface_decompose_halftime, 124
- surface_decompose_rate, 124
- surface_immobile, 128
- surface_in, 128
- surface_mixture, 128
- surface_out, 128
- surface_release, 129, 323
- surface_runoff, 128
- surface_solute, 128
- surface_sorption, 331
- surface_storage, 125
- surface_tillage, 128
- surface_transform, 128
- surface_water, 140
- Sv, 295
- svat, 107, 385
- svinegyllsepareret_tykh, 100
- svinegyllsepareret_tynd, 100
- svinegyllsepareret, 100
- swap, 83
- Swater, 485
- swinbank, 262
- switch, 493
- T, 279, 486, 491
- t, 83
- T w.w./ha, 414
- t/ha, 427
- T_0, 392
- T_c, 392
- T_factor, 481
- T_freezing, 487
- T_max, 451, 478
- T_min, 451, 478
- T_s, 392
- T_shadow, 392
- T_sum, 78, 161
- T_sun, 392
- T_thawing, 487
- T_top, 486
- table, 83, 163, 192, 193, 236, 454
- table_high, 490
- table_low, 489
- tabular, 315
- tag, 85, 184, 187, 306, 309, 310, 313, 350, 366, 368, 369, 380, 456, 464, 467, 469, 471, 514, 517
- tag_density, 314
- tag_x, 314
- tag_z_max, 314
- tag_z_min, 314
- tags, 310
- tair, 391
- TaLim, 445
- TAmplitude, 452, 479
- target, 73, 518
- TaSum, 445
- TAverage, 452, 479
- tcan, 389
- TempEff, 294, 297
- TempEff1, 289
- TempEff2, 289
- temperature, 76, 145, 149, 518
- TempOffset, 454, 481
- Tertiary, 257
- tertiary, 395
- then, 75, 147, 269
- thermodynamic_temperature, 412
- Theta, 271, 487, 493
- theta, 294, 296
- theta_0_20, 390
- Theta_b, 217, 220, 222
- theta_c, 387
- Theta_error, 436
- Theta_fc, 218
- Theta_fo, 57
- Theta_fu, 57
- Theta_pre, 493
- Theta_primary, 489
- Theta_res, 215, 216, 219–222, 224–226
- Theta_sat, 215–227
- Theta_secondary, 489

- theta_w, 387
- Theta_wo, 57
- Theta_wp, 218
- Theta_wu, 57
- tillage, 68, 127, 488
- tillage_age, 139
- tillage_C_soil, 282
- tillage_C_top, 282
- tillage_index, 68
- tillage_N_soil, 282
- tillage_N_top, 281
- tillage_replenish_all, 323
- tillage_water, 109
- Time, 513
- time, 150, 163, 185, 302, 379, 380, 411, 460, 483
- time_columns, 235
- time_left, 518
- time_offset, 367, 369
- time_step_reduction, 431, 434, 435
- Timestep, 453, 477, 480
- timestep, 151, 302, 365, 368
- TimeZone, 452, 479
- title, 181, 366, 369–371, 379, 464, 465, 467–470, 472, 473
- tleaf, 389
- TLLeafEff, 511
- TLRootEff, 511
- tmax, 388
- tmin, 388
- to, 69, 70, 72–74, 77, 84, 149, 150, 235, 354, 358, 360, 458, 460, 471
- top, 184, 186, 447, 475
- top_C, 50
- top_CO2, 282
- top_DM, 282
- top_loss, 129
- top_N, 50
- top_soil, 268, 271, 311
- top_storage, 129
- top_summary, 281
- Toplevel, 516
- topsoil, 227
- tortuosity, 199, 399
- total_C, 282
- total_C_fraction, 95, 101
- total_ea, 109
- total_ep, 109
- total_N, 282
- total_N_fraction, 95, 101
- total_NIR, 111
- total_PAR, 111
- total_water_use, 503
- trace, 133, 134
- trafficable, 144
- transform, 167, 325, 401
- transport, 403
- tref, 387
- true, 118, 151
- tskin, 389
- TSum, 287
- TSum_above, 143
- TSum_limit, 143
- TSum_now, 143
- turnover_factor, 200
- turnover_halftime, 50, 60, 62, 277, 513
- turnover_rate, 50, 60, 62, 277, 512
- type, 119, 183
- u_ref, 391
- ublas, 363
- ug, 427
- ug/ha, 427
- ug/ha/d, 428
- ug/ha/h, 428
- ug/L, 427
- ug/l, 428
- ug/L/MPa, 427
- ug/m², 428
- ug/m²/h, 428
- ug/m³, 428
- ui, 405, 516
- uifilter, 407
- uiitem, 409
- um, 428
- uniform, 340
- unit, 238, 242, 247, 249, 411
- upstream_weight, 404
- uptake, 48
- USDA3, 206
- USDA7, 207
- use, 459
- use_h_old, 142
- use_secondary, 346
- use_wet, 285
- USPRA3, 208
- USPRA4, 209
- ustar_raa, 390
- ustar_raastab1, 390
- ustar_raastab2, 390
- uz1d, 431
- uzmodel, 433
- uzrect, 435
- v+h, 437
- V_b, 78
- V_e, 78
- V_I, 78

- V_r, 78
- V_u, 78
- Vaarbyg, 159
- valid, 367, 465
- value, 76, 81, 113, 114, 123, 126, 127, 163, 177, 178, 201, 230, 236, 266, 267, 305, 310, 311, 339, 340, 352, 377, 378, 380, 396, 450, 486, 488, 517, 518
- values, 118
- vapor_flux_factor, 231, 232
- vapor_pressure, 450
- VapPres, 452, 479
- variable_pool, 280
- variable_pool_2, 280
- vector, 185
- Vegetation, 139
- vegetation, 439
- VegThrs, 288
- VegTSum, 288
- Vernal, 154
- vernalization, 445
- vertical, 258, 437
- very_coarse_sand, 208
- very_fine_sand, 207
- Vinterbyg, 159
- Vinterhvede, 159
- Vinterraps, 159
- Vm_vector, 292
- Vmax, 293
- volatilization, 95, 102
- volume, 71, 72, 74, 77, 84, 234, 353, 357, 359, 447, 492, 518
- volume_base, 356
- vS_S, 91
- W, 414
- w, 387
- W m⁻², 415
- W/cm², 415
- W/m², 415
- wait, 85
- wait_days, 86
- wait_hours, 86
- wait_mm_dd, 86
- wait_period, 86
- warn_known, 266
- warning, 86
- water, 115, 150, 359
- water_capacity, 232
- water_factor, 50, 59, 61, 263, 264, 276, 326, 512
- water_factor_fast, 327
- water_failure_level, 257
- water_height, 396
- water_interval, 359
- water_productivity, 484
- water_stress, 496
- water_stress_days, 483, 496
- water_stress_effect, 154
- water_volume, 360, 396
- Wclover, 159
- Wclover-SLOW, 51
- wday, 514
- WDead, 506
- weather, 138, 165, 286, 301, 449
- Weatherdata, 477
- week, 514
- weekly, 151
- weight, 94, 95, 101, 102, 144, 278, 306, 380
- Wesseling, 171
- wet, 285
- what, 67, 315
- when, 184, 187, 233, 235, 350, 467, 472
- where, 181, 233, 234, 236, 277, 283, 307, 315, 381, 383
- while, 87
- White cabbage, 160
- White cabbage - transplanted, 160
- Width, 338
- Wind, 452, 479
- wind, 450
- wind_speed_field, 111
- wind_speed_weather, 111
- Winter Barley, 160
- Winter Barley; Foulum, 160
- Winter Barley; Koge, 160
- Winter Rape, 160
- Winter Wheat, 160
- Winter Wheat; Eest, 160
- Winter Wheat; Foulum, 160
- with, 367, 369–371, 464, 466, 468–470, 472, 473
- with-column, 87
- WLeaf, 506
- write, 315
- WRoot, 304, 338, 506
- wse, 315, 461
- WSF, 373
- WSOrg, 506
- WStem, 506
- x, 137, 169, 182, 194, 258, 271, 273, 310, 352, 463, 465, 466, 469, 476
- x2max, 188
- x2min, 188
- x_dimension, 469, 471

x_end, 313
X_ice, 488
X_ice.buffer, 488
x_max, 309
x_min, 309
x_offset, 312, 314
x_pond, 493
x_start, 313
x_tic_max, 310
x_tics, 310
xbar, 465
xmax, 188
xmin, 188
Xn, 291
xor, 118
xplus, 115, 515
xy, 187
xycombine, 472
xysource, 463

y, 137, 182, 271, 310, 353, 464, 465, 469
y2max, 185, 188
y2min, 185, 188
y_dimension, 469, 471
y_half, 461
y_max, 310
y_min, 309
ybar, 465
yday, 151, 513
year, 151, 441, 442, 513
year_of_last_harvest, 67
yearly, 151
YearlyLAI, 442
ymax, 185, 188
ymin, 185, 188
yolo, 228

z, 258, 267, 352, 466, 476, 493
z0_def, 387
z3_lowest, 115
z_0, 56
z_0b, 391
z_0s, 387
Z_aquitard, 191, 194
z_end, 313
z_m, 56
z_max, 310
z_min, 310
z_mixing, 491
z_o, 57
z_pond, 493
z_top, 433
z_v, 56
z_xA, 56
z_xJ, 57
zero_gradient, 267
zeta, 387
zi, 324
zone, 475
zones, 492
zplus, 515