

SWAT

Generated by Doxygen 1.8.16



<b>1 SWAT</b>	<b>1</b>
<b>2 Modules Index</b>	<b>7</b>
2.1 Modules List . . . . .	7
<b>3 Data Type Index</b>	<b>9</b>
3.1 Data Types List . . . . .	9
<b>4 File Index</b>	<b>11</b>
4.1 File List . . . . .	11
<b>5 Module Documentation</b>	<b>15</b>
5.1 parm Module Reference . . . . .	15
5.1.1 Detailed Description . . . . .	101
5.1.2 Variable Documentation . . . . .	101
5.1.2.1 igropt . . . . .	101
<b>6 Data Type Documentation</b>	<b>103</b>
6.1 parm::fcgd Interface Reference . . . . .	103
6.2 parm::regres Interface Reference . . . . .	103
6.3 parm::tair Interface Reference . . . . .	103
6.4 parm::theta Interface Reference . . . . .	103
<b>7 File Documentation</b>	<b>105</b>
7.1 albedo.f90 File Reference . . . . .	105
7.1.1 Detailed Description . . . . .	105
7.1.2 Function/Subroutine Documentation . . . . .	105
7.1.2.1 albedo() . . . . .	105
7.2 allocate_parms.f90 File Reference . . . . .	106
7.2.1 Detailed Description . . . . .	106
7.3 alph.f90 File Reference . . . . .	106
7.3.1 Detailed Description . . . . .	106
7.3.2 Function/Subroutine Documentation . . . . .	106
7.3.2.1 alph() . . . . .	106
7.4 anfert.f90 File Reference . . . . .	107
7.4.1 Detailed Description . . . . .	107
7.4.2 Function/Subroutine Documentation . . . . .	107
7.4.2.1 anfert() . . . . .	107
7.5 apply.f90 File Reference . . . . .	107
7.5.1 Detailed Description . . . . .	108
7.5.2 Function/Subroutine Documentation . . . . .	108
7.5.2.1 apply() . . . . .	108
7.6 ascrv.f90 File Reference . . . . .	108
7.6.1 Detailed Description . . . . .	108

---

7.6.2 Function/Subroutine Documentation	108
7.6.2.1 ascrv()	109
7.7 atri.f90 File Reference	109
7.7.1 Detailed Description	109
7.7.2 Function/Subroutine Documentation	109
7.7.2.1 atri()	110
7.8 aunif.f90 File Reference	110
7.8.1 Detailed Description	110
7.8.2 Function/Subroutine Documentation	110
7.8.2.1 aunif()	110
7.9 autoirr.f90 File Reference	111
7.9.1 Detailed Description	111
7.9.2 Function/Subroutine Documentation	111
7.9.2.1 autoirr()	111
7.10 bacteria.f90 File Reference	111
7.10.1 Detailed Description	112
7.10.2 Function/Subroutine Documentation	112
7.10.2.1 bacteria()	112
7.11 bmpinit.f90 File Reference	112
7.11.1 Detailed Description	112
7.11.2 Function/Subroutine Documentation	112
7.11.2.1 bmpinit()	112
7.12 burnop.f90 File Reference	113
7.12.1 Detailed Description	113
7.12.2 Function/Subroutine Documentation	113
7.12.2.1 burnop()	113
7.13 canopyint.f90 File Reference	113
7.13.1 Detailed Description	114
7.14 caps.f90 File Reference	114
7.14.1 Detailed Description	114
7.14.2 Function/Subroutine Documentation	114
7.14.2.1 caps()	114
7.15 carbon_new.f90 File Reference	114
7.15.1 Detailed Description	115
7.15.2 Function/Subroutine Documentation	115
7.15.2.1 carbon()	115
7.16 carbon_zhang2.f90 File Reference	116
7.16.1 Detailed Description	116
7.16.2 Function/Subroutine Documentation	116
7.16.2.1 carbon_zhang2()	116
7.17 cfactor.f90 File Reference	116
7.17.1 Detailed Description	116

---

7.18 clgen.f90 File Reference . . . . .	117
7.18.1 Detailed Description . . . . .	117
7.18.2 Function/Subroutine Documentation . . . . .	117
7.18.2.1 clgen() . . . . .	117
7.19 clcon.f90 File Reference . . . . .	117
7.19.1 Detailed Description . . . . .	117
7.19.2 Function/Subroutine Documentation . . . . .	118
7.19.2.1 clcon() . . . . .	118
7.20 command.f90 File Reference . . . . .	118
7.20.1 Detailed Description . . . . .	118
7.20.2 Function/Subroutine Documentation . . . . .	118
7.20.2.1 command() . . . . .	118
7.21 conapply.f90 File Reference . . . . .	119
7.21.1 Detailed Description . . . . .	119
7.21.2 Function/Subroutine Documentation . . . . .	119
7.21.2.1 conapply() . . . . .	119
7.22 confert.f90 File Reference . . . . .	119
7.22.1 Detailed Description . . . . .	120
7.22.2 Function/Subroutine Documentation . . . . .	120
7.22.2.1 confert() . . . . .	120
7.23 crackflow.f90 File Reference . . . . .	120
7.23.1 Detailed Description . . . . .	120
7.24 crackvol.f90 File Reference . . . . .	120
7.24.1 Detailed Description . . . . .	121
7.25 curno.f90 File Reference . . . . .	121
7.25.1 Detailed Description . . . . .	121
7.25.2 Function/Subroutine Documentation . . . . .	121
7.25.2.1 curno() . . . . .	121
7.26 dailycn.f90 File Reference . . . . .	122
7.26.1 Detailed Description . . . . .	122
7.27 decay.f90 File Reference . . . . .	122
7.27.1 Detailed Description . . . . .	122
7.27.2 Function/Subroutine Documentation . . . . .	122
7.27.2.1 decay() . . . . .	122
7.28 depstor.f90 File Reference . . . . .	123
7.28.1 Detailed Description . . . . .	123
7.28.2 Function/Subroutine Documentation . . . . .	123
7.28.2.1 depstor() . . . . .	123
7.29 dormant.f90 File Reference . . . . .	123
7.29.1 Detailed Description . . . . .	123
7.29.2 Function/Subroutine Documentation . . . . .	124
7.29.2.1 dormant() . . . . .	124

7.30 drains.f90 File Reference	124
7.30.1 Detailed Description	124
7.30.2 Function/Subroutine Documentation	124
7.30.2.1 drains()	124
7.31 dstn1.f90 File Reference	125
7.31.1 Detailed Description	125
7.31.2 Function/Subroutine Documentation	125
7.31.2.1 dstn1()	125
7.32 ee.f90 File Reference	125
7.32.1 Detailed Description	126
7.32.2 Function/Subroutine Documentation	126
7.32.2.1 ee()	126
7.33 eiusle.f90 File Reference	126
7.33.1 Detailed Description	126
7.34 enrsb.f90 File Reference	126
7.34.1 Detailed Description	127
7.34.2 Function/Subroutine Documentation	127
7.34.2.1 enrsb()	127
7.35 estimate_ksat.f90 File Reference	127
7.35.1 Detailed Description	127
7.35.2 Function/Subroutine Documentation	127
7.35.2.1 estimate_ksat()	128
7.36 etact.f90 File Reference	128
7.36.1 Detailed Description	128
7.37 etpot.f90 File Reference	128
7.37.1 Detailed Description	128
7.37.2 Function/Subroutine Documentation	129
7.37.2.1 etpot()	129
7.38 expo.f90 File Reference	129
7.38.1 Detailed Description	129
7.38.2 Function/Subroutine Documentation	129
7.38.2.1 expo()	129
7.39 fert.f90 File Reference	130
7.39.1 Detailed Description	130
7.39.2 Function/Subroutine Documentation	130
7.39.2.1 fert()	130
7.40 finalbal.f90 File Reference	130
7.40.1 Detailed Description	130
7.41 gcycl.f90 File Reference	131
7.41.1 Detailed Description	131
7.42 getallo.f90 File Reference	131
7.42.1 Detailed Description	131

7.43 graze.f90 File Reference . . . . .	131
7.43.1 Detailed Description . . . . .	132
7.43.2 Function/Subroutine Documentation . . . . .	132
7.43.2.1 graze() . . . . .	132
7.44 grow.f90 File Reference . . . . .	132
7.44.1 Detailed Description . . . . .	132
7.44.2 Function/Subroutine Documentation . . . . .	132
7.44.2.1 grow() . . . . .	132
7.45 gwmod.f90 File Reference . . . . .	133
7.45.1 Detailed Description . . . . .	133
7.45.2 Function/Subroutine Documentation . . . . .	133
7.45.2.1 gwmod() . . . . .	133
7.46 gwmod_deep.f90 File Reference . . . . .	133
7.46.1 Detailed Description . . . . .	134
7.46.2 Function/Subroutine Documentation . . . . .	134
7.46.2.1 gwmod_deep() . . . . .	134
7.47 h2omgt_init.f90 File Reference . . . . .	134
7.47.1 Detailed Description . . . . .	134
7.48 harvestop.f90 File Reference . . . . .	134
7.48.1 Detailed Description . . . . .	135
7.48.2 Function/Subroutine Documentation . . . . .	135
7.48.2.1 harvestop() . . . . .	135
7.49 harvkillop.f90 File Reference . . . . .	135
7.49.1 Detailed Description . . . . .	135
7.49.2 Function/Subroutine Documentation . . . . .	135
7.49.2.1 harvkillop() . . . . .	135
7.50 headout.f90 File Reference . . . . .	136
7.50.1 Detailed Description . . . . .	136
7.51 hmeas.f90 File Reference . . . . .	136
7.51.1 Detailed Description . . . . .	136
7.52 hruaa.f90 File Reference . . . . .	136
7.52.1 Detailed Description . . . . .	137
7.52.2 Function/Subroutine Documentation . . . . .	137
7.52.2.1 hruaa() . . . . .	137
7.53 hruallo.f90 File Reference . . . . .	137
7.53.1 Detailed Description . . . . .	137
7.54 hrumon.f90 File Reference . . . . .	137
7.54.1 Detailed Description . . . . .	138
7.55 hruyr.f90 File Reference . . . . .	138
7.55.1 Detailed Description . . . . .	138
7.56 hydroinit.f90 File Reference . . . . .	138
7.56.1 Detailed Description . . . . .	138

7.57 icl.f90 File Reference . . . . .	138
7.57.1 Detailed Description . . . . .	139
7.57.2 Function/Subroutine Documentation . . . . .	139
7.57.2.1 icl() . . . . .	139
7.58 impnd_init.f90 File Reference . . . . .	139
7.58.1 Detailed Description . . . . .	139
7.59 impndmon.f90 File Reference . . . . .	139
7.59.1 Detailed Description . . . . .	140
7.60 impndyr.f90 File Reference . . . . .	140
7.60.1 Detailed Description . . . . .	140
7.61 irrigate.f90 File Reference . . . . .	140
7.61.1 Detailed Description . . . . .	140
7.61.2 Function/Subroutine Documentation . . . . .	140
7.61.2.1 irrigate() . . . . .	140
7.62 irrsb.f90 File Reference . . . . .	141
7.62.1 Detailed Description . . . . .	141
7.62.2 Function/Subroutine Documentation . . . . .	141
7.62.2.1 irrsb() . . . . .	141
7.63 jdt.f90 File Reference . . . . .	141
7.63.1 Detailed Description . . . . .	142
7.63.2 Function/Subroutine Documentation . . . . .	142
7.63.2.1 jdt() . . . . .	142
7.64 killop.f90 File Reference . . . . .	142
7.64.1 Detailed Description . . . . .	142
7.64.2 Function/Subroutine Documentation . . . . .	142
7.64.2.1 killop() . . . . .	143
7.65 lid_cistern.f90 File Reference . . . . .	143
7.65.1 Detailed Description . . . . .	143
7.65.2 Function/Subroutine Documentation . . . . .	143
7.65.2.1 lid_cistern() . . . . .	143
7.66 lid_greenroof.f90 File Reference . . . . .	144
7.66.1 Detailed Description . . . . .	144
7.66.2 Function/Subroutine Documentation . . . . .	144
7.66.2.1 lid_greenroof() . . . . .	144
7.67 lid_por pavement.f90 File Reference . . . . .	144
7.67.1 Detailed Description . . . . .	145
7.67.2 Function/Subroutine Documentation . . . . .	145
7.67.2.1 lid_por pavement() . . . . .	145
7.68 lid_raingarden.f90 File Reference . . . . .	145
7.68.1 Detailed Description . . . . .	145
7.68.2 Function/Subroutine Documentation . . . . .	146
7.68.2.1 lid_raingarden() . . . . .	146



7.69 lidinit.f90 File Reference . . . . .	146
7.69.1 Detailed Description . . . . .	146
7.69.2 Function/Subroutine Documentation . . . . .	146
7.69.2.1 lidinit() . . . . .	146
7.70 lids.f90 File Reference . . . . .	147
7.70.1 Detailed Description . . . . .	147
7.70.2 Function/Subroutine Documentation . . . . .	147
7.70.2.1 lids() . . . . .	147
7.71 lwqdef.f90 File Reference . . . . .	148
7.71.1 Detailed Description . . . . .	148
7.71.2 Function/Subroutine Documentation . . . . .	148
7.71.2.1 lwqdef() . . . . .	148
7.72 main.f90 File Reference . . . . .	148
7.72.1 Detailed Description . . . . .	148
7.73 modparm.f90 File Reference . . . . .	149
7.73.1 Detailed Description . . . . .	234
7.74 ndenit.f90 File Reference . . . . .	235
7.74.1 Detailed Description . . . . .	235
7.74.2 Function/Subroutine Documentation . . . . .	235
7.74.2.1 ndenit() . . . . .	235
7.75 newtillmix.f90 File Reference . . . . .	235
7.75.1 Detailed Description . . . . .	236
7.75.2 Function/Subroutine Documentation . . . . .	236
7.75.2.1 newtillmix() . . . . .	236
7.76 nfix.f90 File Reference . . . . .	236
7.76.1 Detailed Description . . . . .	236
7.76.2 Function/Subroutine Documentation . . . . .	237
7.76.2.1 nfix() . . . . .	237
7.77 nitvol.f90 File Reference . . . . .	237
7.77.1 Detailed Description . . . . .	237
7.77.2 Function/Subroutine Documentation . . . . .	237
7.77.2.1 nitvol() . . . . .	237
7.78 nlch.f90 File Reference . . . . .	238
7.78.1 Detailed Description . . . . .	238
7.78.2 Function/Subroutine Documentation . . . . .	238
7.78.2.1 nlch() . . . . .	238
7.79 nminrl.f90 File Reference . . . . .	238
7.79.1 Detailed Description . . . . .	238
7.79.2 Function/Subroutine Documentation . . . . .	239
7.79.2.1 nminrl() . . . . .	239
7.80 npup.f90 File Reference . . . . .	239
7.80.1 Detailed Description . . . . .	239

7.80.2 Function/Subroutine Documentation . . . . .	239
7.80.2.1 npup() . . . . .	239
7.81 nrain.f90 File Reference . . . . .	240
7.81.1 Detailed Description . . . . .	240
7.81.2 Function/Subroutine Documentation . . . . .	240
7.81.2.1 nrain() . . . . .	240
7.82 nup.f90 File Reference . . . . .	240
7.82.1 Detailed Description . . . . .	240
7.82.2 Function/Subroutine Documentation . . . . .	241
7.82.2.1 nup() . . . . .	241
7.83 nuts.f90 File Reference . . . . .	241
7.83.1 Detailed Description . . . . .	241
7.83.2 Function/Subroutine Documentation . . . . .	241
7.83.2.1 nuts() . . . . .	241
7.84 openwth.f90 File Reference . . . . .	242
7.84.1 Detailed Description . . . . .	242
7.85 operatn.f90 File Reference . . . . .	242
7.85.1 Detailed Description . . . . .	242
7.85.2 Function/Subroutine Documentation . . . . .	242
7.85.2.1 operatn() . . . . .	242
7.86 orgn.f90 File Reference . . . . .	243
7.86.1 Detailed Description . . . . .	243
7.86.2 Function/Subroutine Documentation . . . . .	243
7.86.2.1 orgn() . . . . .	243
7.87 orgncswat.f90 File Reference . . . . .	243
7.87.1 Detailed Description . . . . .	244
7.87.2 Function/Subroutine Documentation . . . . .	244
7.87.2.1 orgncswat() . . . . .	244
7.88 orgncswat2.f90 File Reference . . . . .	244
7.88.1 Detailed Description . . . . .	244
7.88.2 Function/Subroutine Documentation . . . . .	244
7.88.2.1 orgncswat2() . . . . .	245
7.89 origtile.f90 File Reference . . . . .	245
7.89.1 Detailed Description . . . . .	245
7.89.2 Function/Subroutine Documentation . . . . .	245
7.89.2.1 origtile() . . . . .	245
7.90 ovr_sed.f90 File Reference . . . . .	246
7.90.1 Detailed Description . . . . .	246
7.91 percmacro.f90 File Reference . . . . .	246
7.91.1 Detailed Description . . . . .	246
7.91.2 Function/Subroutine Documentation . . . . .	246
7.91.2.1 percmacro() . . . . .	246

7.92 percmain.f90 File Reference . . . . .	247
7.92.1 Detailed Description . . . . .	247
7.92.2 Function/Subroutine Documentation . . . . .	247
7.92.2.1 percmain() . . . . .	247
7.93 percmicro.f90 File Reference . . . . .	247
7.93.1 Detailed Description . . . . .	247
7.93.2 Function/Subroutine Documentation . . . . .	248
7.93.2.1 percmicro() . . . . .	248
7.94 pestlch.f90 File Reference . . . . .	248
7.94.1 Detailed Description . . . . .	248
7.94.2 Function/Subroutine Documentation . . . . .	248
7.94.2.1 pestlch() . . . . .	248
7.95 pesty.f90 File Reference . . . . .	249
7.95.1 Detailed Description . . . . .	249
7.95.2 Function/Subroutine Documentation . . . . .	249
7.95.2.1 pesty() . . . . .	249
7.96 pgen.f90 File Reference . . . . .	249
7.96.1 Detailed Description . . . . .	250
7.96.2 Function/Subroutine Documentation . . . . .	250
7.96.2.1 pgen() . . . . .	250
7.97 pgenhr.f90 File Reference . . . . .	250
7.97.1 Detailed Description . . . . .	250
7.98 pkq.f90 File Reference . . . . .	250
7.98.1 Detailed Description . . . . .	251
7.98.2 Function/Subroutine Documentation . . . . .	251
7.98.2.1 pkq() . . . . .	251
7.99 plantmod.f90 File Reference . . . . .	251
7.99.1 Detailed Description . . . . .	252
7.99.2 Function/Subroutine Documentation . . . . .	252
7.99.2.1 plantmod() . . . . .	252
7.100 plantop.f90 File Reference . . . . .	252
7.100.1 Detailed Description . . . . .	252
7.100.2 Function/Subroutine Documentation . . . . .	252
7.100.2.1 plantop() . . . . .	252
7.101 pmeas.f90 File Reference . . . . .	253
7.101.1 Detailed Description . . . . .	253
7.101.2 Function/Subroutine Documentation . . . . .	253
7.101.2.1 pmeas() . . . . .	253
7.102 pminrl.f90 File Reference . . . . .	253
7.102.1 Detailed Description . . . . .	254
7.102.2 Function/Subroutine Documentation . . . . .	254
7.102.2.1 pminrl() . . . . .	254

---

7.103 pminrl2.f90 File Reference . . . . .	254
7.103.1 Detailed Description . . . . .	254
7.103.2 Function/Subroutine Documentation . . . . .	254
7.103.2.1 pminrl2() . . . . .	254
7.104 psed.f90 File Reference . . . . .	255
7.104.1 Detailed Description . . . . .	255
7.104.2 Function/Subroutine Documentation . . . . .	255
7.104.2.1 psed() . . . . .	255
7.105 qman.f90 File Reference . . . . .	255
7.105.1 Detailed Description . . . . .	256
7.105.2 Function/Subroutine Documentation . . . . .	256
7.105.2.1 qman() . . . . .	256
7.106 rchaa.f90 File Reference . . . . .	256
7.106.1 Detailed Description . . . . .	256
7.106.2 Function/Subroutine Documentation . . . . .	257
7.106.2.1 rchaa() . . . . .	257
7.107 rchday.f90 File Reference . . . . .	257
7.107.1 Detailed Description . . . . .	257
7.108 rchmon.f90 File Reference . . . . .	257
7.108.1 Detailed Description . . . . .	257
7.108.2 Function/Subroutine Documentation . . . . .	258
7.108.2.1 rchmon() . . . . .	258
7.109 rchyr.f90 File Reference . . . . .	258
7.109.1 Detailed Description . . . . .	258
7.109.2 Function/Subroutine Documentation . . . . .	258
7.109.2.1 rchyr() . . . . .	258
7.110 readatmodep.f90 File Reference . . . . .	259
7.110.1 Detailed Description . . . . .	259
7.111 readbsn.f90 File Reference . . . . .	259
7.111.1 Detailed Description . . . . .	259
7.112 readchm.f90 File Reference . . . . .	259
7.112.1 Detailed Description . . . . .	259
7.112.2 Function/Subroutine Documentation . . . . .	260
7.112.2.1 readchm() . . . . .	260
7.113 readcnst.f90 File Reference . . . . .	260
7.113.1 Detailed Description . . . . .	260
7.113.2 Function/Subroutine Documentation . . . . .	260
7.113.2.1 readcnst() . . . . .	260
7.114 readfcst.f90 File Reference . . . . .	261
7.114.1 Detailed Description . . . . .	261
7.115 readfert.f90 File Reference . . . . .	261
7.115.1 Detailed Description . . . . .	261

---

7.116 readfig.f90 File Reference . . . . .	261
7.116.1 Detailed Description . . . . .	262
7.117 readfile.f90 File Reference . . . . .	262
7.117.1 Detailed Description . . . . .	262
7.118 readgw.f90 File Reference . . . . .	262
7.118.1 Detailed Description . . . . .	262
7.118.2 Function/Subroutine Documentation . . . . .	262
7.118.2.1 readgw() . . . . .	262
7.119 readhru.f90 File Reference . . . . .	263
7.119.1 Detailed Description . . . . .	263
7.119.2 Function/Subroutine Documentation . . . . .	263
7.119.2.1 readhru() . . . . .	263
7.120 readinpt.f90 File Reference . . . . .	263
7.120.1 Detailed Description . . . . .	264
7.121 readlup.f90 File Reference . . . . .	264
7.121.1 Detailed Description . . . . .	264
7.122 readlwq.f90 File Reference . . . . .	264
7.122.1 Detailed Description . . . . .	264
7.122.2 Function/Subroutine Documentation . . . . .	264
7.122.2.1 readlwq() . . . . .	264
7.123 readmgt.f90 File Reference . . . . .	265
7.123.1 Detailed Description . . . . .	265
7.123.2 Function/Subroutine Documentation . . . . .	265
7.123.2.1 readmgt() . . . . .	265
7.124 readmon.f90 File Reference . . . . .	265
7.124.1 Detailed Description . . . . .	266
7.125 readops.f90 File Reference . . . . .	266
7.125.1 Detailed Description . . . . .	266
7.125.2 Function/Subroutine Documentation . . . . .	266
7.125.2.1 readops() . . . . .	266
7.126 readpest.f90 File Reference . . . . .	266
7.126.1 Detailed Description . . . . .	267
7.127 readplant.f90 File Reference . . . . .	267
7.127.1 Detailed Description . . . . .	267
7.128 readpnd.f90 File Reference . . . . .	267
7.128.1 Detailed Description . . . . .	267
7.128.2 Function/Subroutine Documentation . . . . .	267
7.128.2.1 readpnd() . . . . .	267
7.129 readres.f90 File Reference . . . . .	268
7.129.1 Detailed Description . . . . .	268
7.129.2 Function/Subroutine Documentation . . . . .	268
7.129.2.1 readres() . . . . .	268

7.130 readrte.f90 File Reference . . . . .	268
7.130.1 Detailed Description . . . . .	269
7.131 readru.f90 File Reference . . . . .	269
7.131.1 Detailed Description . . . . .	269
7.131.2 Function/Subroutine Documentation . . . . .	269
7.131.2.1 readru() . . . . .	269
7.132 readsdr.f90 File Reference . . . . .	269
7.132.1 Detailed Description . . . . .	270
7.132.2 Function/Subroutine Documentation . . . . .	270
7.132.2.1 readsdr() . . . . .	270
7.133 readsepticbz.f90 File Reference . . . . .	270
7.133.1 Detailed Description . . . . .	270
7.133.2 Function/Subroutine Documentation . . . . .	270
7.133.2.1 readsepticbz() . . . . .	271
7.134 readseptwq.f90 File Reference . . . . .	272
7.134.1 Detailed Description . . . . .	272
7.134.2 Function/Subroutine Documentation . . . . .	272
7.134.2.1 readseptwq() . . . . .	272
7.135 readsno.f90 File Reference . . . . .	272
7.135.1 Detailed Description . . . . .	273
7.135.2 Function/Subroutine Documentation . . . . .	273
7.135.2.1 readsno() . . . . .	273
7.136 readsol.f90 File Reference . . . . .	273
7.136.1 Detailed Description . . . . .	273
7.136.2 Function/Subroutine Documentation . . . . .	273
7.136.2.1 readsol() . . . . .	273
7.137 readsub.f90 File Reference . . . . .	274
7.137.1 Detailed Description . . . . .	274
7.137.2 Function/Subroutine Documentation . . . . .	274
7.137.2.1 readsub() . . . . .	274
7.138 readswq.f90 File Reference . . . . .	274
7.138.1 Detailed Description . . . . .	275
7.139 readtill.f90 File Reference . . . . .	275
7.139.1 Detailed Description . . . . .	275
7.140 readurban.f90 File Reference . . . . .	275
7.140.1 Detailed Description . . . . .	275
7.141 readwgn.f90 File Reference . . . . .	275
7.141.1 Detailed Description . . . . .	276
7.141.2 Function/Subroutine Documentation . . . . .	276
7.141.2.1 readwgn() . . . . .	276
7.142 readwus.f90 File Reference . . . . .	276
7.142.1 Detailed Description . . . . .	276

7.142.2 Function/Subroutine Documentation	276
7.142.2.1 readwus()	277
7.143 readwwq.f90 File Reference	277
7.143.1 Detailed Description	277
7.144 readyr.f90 File Reference	277
7.144.1 Detailed Description	277
7.144.2 Function/Subroutine Documentation	277
7.144.2.1 readyr()	277
7.145 resetlu.f90 File Reference	278
7.145.1 Detailed Description	278
7.146 rhgen.f90 File Reference	278
7.146.1 Detailed Description	278
7.147 rootfr.f90 File Reference	278
7.147.1 Detailed Description	279
7.147.2 Function/Subroutine Documentation	279
7.147.2.1 rootfr()	279
7.148 rsedaa.f90 File Reference	279
7.148.1 Detailed Description	279
7.148.2 Function/Subroutine Documentation	279
7.148.2.1 rsedaa()	279
7.149 rseday.f90 File Reference	280
7.149.1 Detailed Description	280
7.150 rsedmon.f90 File Reference	280
7.150.1 Detailed Description	280
7.150.2 Function/Subroutine Documentation	280
7.150.2.1 rsedmon()	280
7.151 rsedyr.f90 File Reference	281
7.151.1 Detailed Description	281
7.152 rteinit.f90 File Reference	281
7.152.1 Detailed Description	281
7.153 sat_excess.f90 File Reference	281
7.153.1 Detailed Description	282
7.153.2 Function/Subroutine Documentation	282
7.153.2.1 sat_excess()	282
7.154 sched_mgt.f90 File Reference	282
7.154.1 Detailed Description	282
7.154.2 Function/Subroutine Documentation	282
7.154.2.1 sched_mgt()	282
7.155 schedule_ops.f90 File Reference	283
7.155.1 Detailed Description	283
7.155.2 Function/Subroutine Documentation	283
7.155.2.1 schedule_ops()	283

7.156 sim_inityr.f90 File Reference . . . . .	283
7.156.1 Detailed Description . . . . .	284
7.157 simulate.f90 File Reference . . . . .	284
7.157.1 Detailed Description . . . . .	284
7.158 slrgen.f90 File Reference . . . . .	284
7.158.1 Detailed Description . . . . .	284
7.158.2 Function/Subroutine Documentation . . . . .	284
7.158.2.1 slrgen() . . . . .	284
7.159 smeas.f90 File Reference . . . . .	285
7.159.1 Detailed Description . . . . .	285
7.160 snom.f90 File Reference . . . . .	285
7.160.1 Detailed Description . . . . .	285
7.160.2 Function/Subroutine Documentation . . . . .	285
7.160.2.1 snom() . . . . .	285
7.161 soil_chem.f90 File Reference . . . . .	286
7.161.1 Detailed Description . . . . .	286
7.161.2 Function/Subroutine Documentation . . . . .	286
7.161.2.1 soil_chem() . . . . .	286
7.162 soil_phys.f90 File Reference . . . . .	286
7.162.1 Detailed Description . . . . .	287
7.162.2 Function/Subroutine Documentation . . . . .	287
7.162.2.1 soil_phys() . . . . .	287
7.163 soil_write.f90 File Reference . . . . .	287
7.163.1 Detailed Description . . . . .	287
7.163.2 Function/Subroutine Documentation . . . . .	287
7.163.2.1 soil_write() . . . . .	287
7.164 solp.f90 File Reference . . . . .	288
7.164.1 Detailed Description . . . . .	288
7.164.2 Function/Subroutine Documentation . . . . .	288
7.164.2.1 solp() . . . . .	288
7.165 solt.f90 File Reference . . . . .	288
7.165.1 Detailed Description . . . . .	289
7.166 std1.f90 File Reference . . . . .	289
7.166.1 Detailed Description . . . . .	289
7.167 std2.f90 File Reference . . . . .	289
7.167.1 Detailed Description . . . . .	289
7.168 std3.f90 File Reference . . . . .	289
7.168.1 Detailed Description . . . . .	290
7.169 storeinitial.f90 File Reference . . . . .	290
7.169.1 Detailed Description . . . . .	290
7.170 subbasin.f90 File Reference . . . . .	290
7.170.1 Detailed Description . . . . .	290



7.170.2 Function/Subroutine Documentation . . . . .	290
7.170.2.1 subbasin() . . . . .	290
7.171 submon.f90 File Reference . . . . .	291
7.171.1 Detailed Description . . . . .	291
7.172 subyr.f90 File Reference . . . . .	291
7.172.1 Detailed Description . . . . .	291
7.173 surface.f90 File Reference . . . . .	291
7.173.1 Detailed Description . . . . .	292
7.173.2 Function/Subroutine Documentation . . . . .	292
7.173.2.1 surface() . . . . .	292
7.174 surfst_h2o.f90 File Reference . . . . .	292
7.174.1 Detailed Description . . . . .	292
7.174.2 Function/Subroutine Documentation . . . . .	292
7.174.2.1 surfst_h2o() . . . . .	293
7.175 surq_daycn.f90 File Reference . . . . .	293
7.175.1 Detailed Description . . . . .	293
7.175.2 Function/Subroutine Documentation . . . . .	293
7.175.2.1 surq_daycn() . . . . .	293
7.176 surq_greenampt.f90 File Reference . . . . .	294
7.176.1 Detailed Description . . . . .	294
7.176.2 Function/Subroutine Documentation . . . . .	294
7.176.2.1 surq_greenampt() . . . . .	294
7.177 swbl.f90 File Reference . . . . .	294
7.177.1 Detailed Description . . . . .	294
7.177.2 Function/Subroutine Documentation . . . . .	295
7.177.2.1 swbl() . . . . .	295
7.178 swu.f90 File Reference . . . . .	295
7.178.1 Detailed Description . . . . .	295
7.178.2 Function/Subroutine Documentation . . . . .	295
7.178.2.1 swu() . . . . .	295
7.179 tgen.f90 File Reference . . . . .	296
7.179.1 Detailed Description . . . . .	296
7.179.2 Function/Subroutine Documentation . . . . .	296
7.179.2.1 tgen() . . . . .	296
7.180 tillfactor.f90 File Reference . . . . .	296
7.180.1 Detailed Description . . . . .	297
7.180.2 Function/Subroutine Documentation . . . . .	297
7.180.2.1 tillfactor() . . . . .	297
7.181 tmeas.f90 File Reference . . . . .	297
7.181.1 Detailed Description . . . . .	298
7.182 tran.f90 File Reference . . . . .	298
7.182.1 Detailed Description . . . . .	298

7.182.2 Function/Subroutine Documentation	298
7.182.2.1 tran()	298
7.183 tstr.f90 File Reference	298
7.183.1 Detailed Description	299
7.183.2 Function/Subroutine Documentation	299
7.183.2.1 tstr()	299
7.184 ttcoef.f90 File Reference	299
7.184.1 Detailed Description	299
7.184.2 Function/Subroutine Documentation	299
7.184.2.1 ttcoef()	299
7.185 ttcoef_wway.f90 File Reference	300
7.185.1 Detailed Description	300
7.186 varinit.f90 File Reference	300
7.186.1 Detailed Description	300
7.186.2 Function/Subroutine Documentation	300
7.186.2.1 varinit()	300
7.187 vbl.f90 File Reference	301
7.187.1 Detailed Description	301
7.187.2 Function/Subroutine Documentation	301
7.187.2.1 vbl()	301
7.188 volq.f90 File Reference	302
7.188.1 Detailed Description	302
7.188.2 Function/Subroutine Documentation	302
7.188.2.1 volq()	302
7.189 water_hru.f90 File Reference	303
7.189.1 Detailed Description	303
7.189.2 Function/Subroutine Documentation	303
7.189.2.1 water_hru()	303
7.190 wattable.f90 File Reference	303
7.190.1 Detailed Description	303
7.191 weatgn.f90 File Reference	304
7.191.1 Detailed Description	304
7.191.2 Function/Subroutine Documentation	304
7.191.2.1 weatgn()	304
7.192 wmeas.f90 File Reference	304
7.192.1 Detailed Description	304
7.193 wndgen.f90 File Reference	305
7.193.1 Detailed Description	305
7.193.2 Function/Subroutine Documentation	305
7.193.2.1 wndgen()	305
7.194 writea.f90 File Reference	305
7.194.1 Detailed Description	305

7.194.2 Function/Subroutine Documentation . . . . .	306
7.194.2.1 writea() . . . . .	306
7.195 writed.f90 File Reference . . . . .	306
7.195.1 Detailed Description . . . . .	306
7.196 writem.f90 File Reference . . . . .	306
7.196.1 Detailed Description . . . . .	306
7.196.2 Function/Subroutine Documentation . . . . .	307
7.196.2.1 writem() . . . . .	307
7.197 xmon.f90 File Reference . . . . .	307
7.197.1 Detailed Description . . . . .	307
7.198 ysed.f90 File Reference . . . . .	307
7.198.1 Detailed Description . . . . .	307
7.198.2 Function/Subroutine Documentation . . . . .	308
7.198.2.1 ysed() . . . . .	308
7.199 zero0.f90 File Reference . . . . .	308
7.199.1 Detailed Description . . . . .	308
7.200 zero1.f90 File Reference . . . . .	308
7.200.1 Detailed Description . . . . .	308
7.201 zero2.f90 File Reference . . . . .	309
7.201.1 Detailed Description . . . . .	309
7.202 zero_urban.f90 File Reference . . . . .	309
7.202.1 Detailed Description . . . . .	309
7.203 zeroini.f90 File Reference . . . . .	309
7.203.1 Detailed Description . . . . .	309
<b>Bibliography</b>	<b>311</b>
<b>Index</b>	<b>313</b>



# Chapter 1

## SWAT

An upgraded SWAT 2012 revision 670 code

### Objectives

- Standard indentation and translation to Fortran 90 by using `findent`. See the `translate-fortran90.pl` perl script file (:heavy\_check\_mark:)
- Exhaustive use of the "implicit none" directive to detect bad variable usage (:heavy\_check\_mark:)
- Generate a GNU `Make` makefile and compile with GNU `GFortran`. See the `generate-makefile.pl` perl script file (:heavy\_check\_mark:)
- Remove non-used variables and format labels (:heavy\_check\_mark:)
- Detect and solve all uninitialized variables (:heavy\_check\_mark: :construction:, some proposed solutions could be incorrect)
- Remove unneeded variable initializations (:heavy\_check\_mark:) as:  

```
j=0 ! this line is not necessary
j=ihru
```
- Remove redundant code (:heavy\_check\_mark:)
- Exhaustive use of the "parameter" directive on constants (:heavy\_check\_mark:)
- Remove global counters (as `i`, `ihru`, `iihru` or `idum` in module `parm`). Using local counters or passing values as argument are preferred (:construction:)
- Generate a detailed list of issues detected in the original code (:heavy\_check\_mark:, see at the end of this README)
- Remove obsolete commented code (:x:)
- Update variable descriptions in comments (:construction:, a lot of work)
- Standardize comments by using Doxygen style in order to generate documentation. See at `latex/refman.pdf` (:construction:, a lot of work)

## Required tools

- `GFortran` (to compile the source code)
- `Make` (to build the executable file)
- `Perl` (optional: to execute the perl scripts to update the makefile or to translate original files to Fortran 90)
- `Findent` (optional: to translate original files to Fortran 90 with a standard indentation)
- `Doxygen` (optional: to generate a reference programming manual from source code)
- `TeX Live` or `MiKTeX` (optional: to generate a reference programming manual from source code)
- On Microsoft Windows systems you have to install `MSYS2` and the required utilities ( `GFortran` and `Make`). You can follow detailed instructions in `install-unix`

## Instructions to generate Fortran 90 style code from original code

In order to generate Fortran 90 style code with standard indentation from original code you have to type on a UNIX type terminal (you need `Perl` and `Findent`):

```
$ perl translate-fortran90.pl
```

## Instructions to generate an initial GNU make Makefile

Type on the UNIX type terminal, when translated the original code to Fortran 90 style (you need `Perl`):

```
$ perl generate-makefile.pl
```

## Instructions to generate an executable to test

Type on the UNIX type terminal (you need `GFortran` and `Make`)

- In UNIX type operative systems:

```
$ make
```

- In a `MSYS2` terminal in Microsoft Windows:

```
$ EXE=".exe" LDFLAGS="-static" make
```

- Cross-compiling a 32 bits Microsoft Windows executable in a UNIX type operative system:

```
$ prefix="i686-w64-mingw32-" EXE=".exe" LDFLAGS="-static" make
```

- Cross-compiling a 64 bits Microsoft Windows executable in a UNIX type operative system:

```
$ prefix="x86_64-w64-mingw32-" EXE=".exe" LDFLAGS="-static" make
```

## Instructions to generate an optimized executable file

Type on the UNIX type terminal (you need `GFortran` and `Make`)

- In UNIX type operative systems:

```
$ CFLAGS="-march=native -flto" LDFLAGS="-flto" make strip
```

- In a `MSYS2` terminal in Microsoft Windows:

```
$ EXE=".exe" CFLAGS="-flto" LDFLAGS="-flto -static" make strip
```

- Cross-compiling a 32 bits Microsoft Windows executable in a UNIX type operative system:

```
$ prefix="i686-w64-mingw32-" EXE=".exe" CFLAGS="-flto" LDFLAGS="-flto -static" make strip
```

- Cross-compiling a 64 bits Microsoft Windows executable in a UNIX type operative system:

```
$ prefix="x86_64-w64-mingw32-" EXE=".exe" CFLAGS="-flto" LDFLAGS="-flto -static" make strip
```

## Instructions to generate a reference programming manual from source code

Type on the UNIX type terminal (you need `Doxygen` and `TeX Live` or `MiKTeX`):

```
$ make latex/refman.pdf
```

The reference programming manual file `latex/refman.pdf` is generated from source code in PDF format

## Issues in the original source code

This is a list of possible issues detected in the original source code. These issues have been mostly detected by the `GFortran` compiler warnings. Some of them could not arise because the logic of the variables is not possible.

- In `biofilm.f`:

- `dcoef` is used but not initialized. `dcoef=3` as in `watqual.f`? Then, I propose at beginning: `real*8, parameter :: dcoef = 3.`

- In `bmp_ri_pond.f`:

- `qseep` and `qet` could be used not initialized at lines 133 and 134. However the problem only arises for `nstep<1`

- In `bmp_sand_filter.f`:

- `sed_removed` at line 342 could be used not initialized if `sfsedstdev<=0`

- In `bpm_sed_pond.f`:

- `bmp_sed_pond` seems to be `bmp_sed_pond` at line 186

- In `bmp_wet_pond.f`:

- `hvol` could be used not initialized in `ext_dpth` subroutine at line 267 in first bucle iteration

- In `clicon.f`:
  - `tmxbsb`, `tmnbsb`, `rbsb`, `rstpbsb`, `rhdbbsb`, `rabsb`, `rmxbsb`, `daylbsb`, `fradbsb` and `u10bsb` could be used not initialized at 186-207 lines
- In `conapply.f`:
  - `k` and `kk` could be used not initialized at 121-122 lines if `iday_pest(j) /= ipst_freq(j)` and `curyr > nyskip`
- In `confert.f`:
  - `ifrt` seems to be it at line 214
- In `curno.f`:
  - `smxold` could be used not initialized if `cn1(h) <= 1.e-6` and `curyr /= 0` at line 96
- In `drains.f`:
  - `nlayer` could be used not initialized at line 23. However, the problem only arises if it is not set in the previous bucle (`mlyr <= 1` or `sol_z(j1, j) <= 0`)
- In `etact.f`:
  - `sev` could be used not initialized at line 286 if `dep >= esd` and `ly == 2`
- In `filter.f`:
  - `remove21` seems to be `remove2` at line 316
- In `grass_wway.f`:
  - `sf_depth` and `sf_sed` could be used not initialized at lines 133 and 137 if `sf_area > 0` and `sf_area <= 1.e-6`
- In `headout.f`:
  - `hedr` array of column titles is written out of defined bounds at lines 118, 119, 121 and 133. It is written to `mrcho` (set to 62 in `allocate_parms.f` line 59) but in `modparm.f` the bound of `hedr` array is set to 46 (line 663)
- In `hhnoqual.f`:
  - `algon` seems to be `algcon` at line 190
- In `hhwatqual.f`
  - `orgnpin` seems to be `orgpin` at line 278
  - `thour = 1.0` at line 377 overwrites previous `thour` calculation. It is wrong
- In `hmeas.f`:
  - `rhdbbsb` could be used not initialized at line 84
- In `killop.f`:
  - `ff1` and `ff2` are used but not initialized at lines 167 and 267. They are set in `harvkillop.f` file (lines 257-258). They have to be included in `modparm.f` to share `harvkillop.f` values? or they have to be redefined as in `harvkillop.f`?
- In `NCsed_leach.f90`:
  - `perc_clyr` could be used not initialized at line 221 if `sol_nly(j) < 2`
- In `nrain.f`:
  - `no2pcp` seems to be `no3pcp` at line 72



- In pmeas.f:
  - rbsb could be used not initialized at line 143
  - flag could be used not initialized if 'a==' 'at line 210 -rainsb could be used not initialized, however only if nstep<=0'
- In pminrl2.f:
  - at line 95 a comma is necessary between base and vara
  - ssp could be used not initialized at line 196 if xx<=1.e-6
- In pothole.f:
  - solp\_tileo could be used not initialized at line 593 if pot\_vol(j)<=1.e-6 or potvol\_← tile<=1.e-6
- In potholehr.f:
  - potflow seems to be potflwo at line 447
- In readatmodep.f:
  - momax=12\*nbyr is defined at line 65 but not used. It has to be mo\_max? but then, it overwrites the file read
- In readops.f:
  - year = 0. seems to be iyear = 0 at line 98
  - mg13 seems to be mgt13 at line 206
- In readpnd.f:
  - vsetsetlpnd seems to be velsetlpnd at line 279
- In readru.f:
  - tck is used but not initialized at line 79
- In readsepticbz.f:
  - at line 135 4. e-8 seems to be 4.e-8
- In rewind\_init.f:
  - orig\_tnylda is used but not initialized at line 174
- In routels.f:
  - dstor is used but not initialized at line 134. It has to be calculated as in watbal.f? or as in the commented line 109?
  - latqout and gwqout could be used not initialized at lines 142-143
- In rtbact.f:
  - netwtr could be used not initialized at line 124, however only if nstep<1
- In rthpest.f:
  - thour=1.0 at line 183 overwrites previous thour calculation. It is wrong
  - frsol and frsrb could be used not initialized at lines 289-290 if hrtwtr(ii)>0.001 and hrtwtr(ii)/(idt\*60)<=0.01
- In rtpest.f:
  - tday=1.0 at line 180 overwrites previous tday calculation. It is wrong
- In sched\_mgt.f:

- `< =` seems to be `<=` at 202 line
  - `husc` and `igrow` at lines 264-265 are used but not initialized. `husc` has to be `phu_op(iop, ihru)` has in `readmgt.f?` `igrow` has to be `igro(ihru)` has in `readmgt.f?`
- In `smeas.f`:
  - `rabsb` could be used not initialized at line 86
- In `sweep.f`:
  - `fr_curb` is used but not initialized at line 56. It has to be added to `modparm.f` to share result with `sched_mgt.f?` or it has to be `mgt5op(nop(ihru), ihru)` as in `sched_mgt.f?`
- In `tmeas.f`:
  - `tmxbsb` and `tmnbsb` could be used not initialized at lines 109-110
- In `transfer.f`:
  - `ratio, xx` and `ratio1` could be used not initialized at lines 236, 239 and 241 if `ihout==2`
- In `wmeas.f`:
  - `u10bsb` could be used not initialized at line 85
- In `zero0.f`:
  - `sol_sumn03` seems to be `sol_sumno3` at line 508
- In `zero_urban.f`:
  - `stp_stagdis` seems to be `stp_stagdis` at line 84
  - `subdr_kg` seems to be `subdr_km` at line 149
  - `spl_eros` is not defined at line 21, it could be `eros_spl?`

## Chapter 2

# Modules Index

### 2.1 Modules List

Here is a list of all documented modules with brief descriptions:

<a href="#">parm</a>	Main module containing the global variables . . . . .	<a href="#">15</a>
----------------------	---	--------------------



## Chapter 3

# Data Type Index

### 3.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">parm::fcgd</a>	.....	103
<a href="#">parm::regres</a>	.....	103
<a href="#">parm::tair</a>	.....	103
<a href="#">parm::theta</a>	.....	103



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

albedo.f90	105
allocate_parms.f90	106
alph.f90	106
anfert.f90	107
apply.f90	107
ascrv.f90	108
atri.f90	109
aunif.f90	110
autoirr.f90	111
bacteria.f90	111
bmpinit.f90	112
burnop.f90	113
canopyint.f90	113
caps.f90	114
carbon_new.f90	114
carbon_zhang2.f90	116
cfactor.f90	116
clgen.f90	117
clicon.f90	117
command.f90	118
conapply.f90	119
confert.f90	119
crackflow.f90	120
crackvol.f90	120
curno.f90	121
dailycn.f90	122
decay.f90	122
depstor.f90	123
dormant.f90	123
drains.f90	124
dstn1.f90	125
ee.f90	125
eiusle.f90	126
enrsb.f90	126
estimate_ksat.f90	127

etact.f90	128
etpot.f90	128
expo.f90	129
fert.f90	130
finalbal.f90	130
gcycl.f90	131
getallo.f90	131
graze.f90	131
grow.f90	132
gwmod.f90	133
gwmod_deep.f90	133
h2omgt_init.f90	134
harvestop.f90	134
harvkillop.f90	135
headout.f90	136
hmeas.f90	136
hruaa.f90	136
hruallo.f90	137
hrumon.f90	137
hruyr.f90	138
hydroinit.f90	138
icl.f90	138
impnd_init.f90	139
impndmon.f90	139
impndyr.f90	140
irrigate.f90	140
irrsup.f90	141
jdt.f90	141
killop.f90	142
lid_cistern.f90	143
lid_greenroof.f90	144
lid_poropavement.f90	144
lid_raingarden.f90	145
lidinit.f90	146
lids.f90	147
lwqdef.f90	148
main.f90	148
modparm.f90	149
ndenit.f90	235
newtillmix.f90	235
nfix.f90	236
nitvol.f90	237
nlch.f90	238
nminrl.f90	238
npup.f90	239
nrain.f90	240
nup.f90	240
nuts.f90	241
openwth.f90	242
operatn.f90	242
orgn.f90	243
orgncswat.f90	243
orgncswat2.f90	244
origtile.f90	245
ovr_sed.f90	246
percmacro.f90	246
percmmain.f90	247
percmicro.f90	247



pestlch.f90	248
pesty.f90	249
pgen.f90	249
pgenhr.f90	250
pkq.f90	250
plantmod.f90	251
plantop.f90	252
pmeas.f90	253
pminrl.f90	253
pminrl2.f90	254
psed.f90	255
qman.f90	255
rchaaf.f90	256
rchday.f90	257
rchmon.f90	257
rchyr.f90	258
readatmodep.f90	259
readbsn.f90	259
readchm.f90	259
readcnst.f90	260
readfcst.f90	261
readfert.f90	261
readfig.f90	261
readfile.f90	262
readgw.f90	262
readhru.f90	263
readingpt.f90	263
readlup.f90	264
readlwq.f90	264
readmgt.f90	265
readmon.f90	265
readops.f90	266
readpest.f90	266
readplant.f90	267
readpnd.f90	267
readres.f90	268
readrte.f90	268
readru.f90	269
readsdr.f90	269
readsepticbz.f90	270
readseptwq.f90	272
readsno.f90	272
readsol.f90	273
readsub.f90	274
readswq.f90	274
readtill.f90	275
readurban.f90	275
readwgn.f90	275
readwus.f90	276
readwwq.f90	277
readyr.f90	277
resetlu.f90	278
rhgen.f90	278
rootfr.f90	278
rsedaa.f90	279
rseday.f90	280
rsedmon.f90	280
rsedyr.f90	281

rteinit.f90	281
sat_excess.f90	281
sched_mgt.f90	282
schedule_ops.f90	283
sim_inityr.f90	283
simulate.f90	284
slrgen.f90	284
smeas.f90	285
snom.f90	285
soil_chem.f90	286
soil_phys.f90	286
soil_write.f90	287
solp.f90	288
solt.f90	288
std1.f90	289
std2.f90	289
std3.f90	289
storeinitial.f90	290
subbasin.f90	290
submon.f90	291
subyr.f90	291
surface.f90	291
surfst_h2o.f90	292
surq_daycn.f90	293
surq_greenampt.f90	294
swbl.f90	294
swu.f90	295
tgen.f90	296
tillfactor.f90	296
tmeas.f90	297
tran.f90	298
tstr.f90	298
ttcoef.f90	299
ttcoef_wway.f90	300
varinit.f90	300
vbl.f90	301
volq.f90	302
water_hru.f90	303
watable.f90	303
weatgn.f90	304
wmeas.f90	304
wndgen.f90	305
writea.f90	305
writed.f90	306
writem.f90	306
xmon.f90	307
ysed.f90	307
zero0.f90	308
zero1.f90	308
zero2.f90	309
zero_urban.f90	309
zeroini.f90	309

## Chapter 5

# Module Documentation

### 5.1 parm Module Reference

main module containing the global variables

#### Data Types

- interface [fcgd](#)
- interface [regres](#)
- interface [tair](#)
- interface [theta](#)

#### Variables

- integer, parameter [mvaro](#) = 33  
*max number of variables routed through the reach*
- integer, parameter [mhruo](#) = 79  
*maximum number of variables written to HRU output file (output.hru) (none)*
- integer, parameter [mrcho](#) = 62  
*maximum number of variables written to reach output file (.rch) (none)*
- integer, parameter [msub](#) = 24  
*maximum number of variables written to subbasin output file (output.sub) (none)*
- integer, parameter [mstdo](#) = 113  
*max number of variables summarized in output.std*
- integer, parameter [motot](#) = 600
- character(len=80), parameter [prog](#) = "SWAT Sep 7 VER 2018/Rev 670"  
*SWAT program header string (name and version)*
- character(len=13), dimension([mhruo](#)), parameter [heds](#) = (/ "PRECIPmm", "SNOFALLmm", "SNOMELTmm", "IRRmm", "PETmm", "ETmm", "SW\_INITmm", "SW\_ENDmm", "PERCmm", "GW\_RCHGmm", "DA\_RCHGmm", "REVAPmm", "SA\_IRRmm", "DA\_IRRmm", "SA\_STmm", "DA\_STmm", "SURQ\_GENmm", "SURQ\_CNTmm", "TLOSSmm", "LATQGENmm", "GW\_Qmm", "WYLDmm", "DAILYCN", "TMP\_AVdgC", "TMP\_MXdgC", "TMP\_MNdgc", "SOL\_TMPdgC", "SOLARMJ/m2", "SYLDt/ha", "USLEt/ha", "N\_APPkg/ha", "P\_APPkg/ha", "NAUTOkg/ha", "PAUTOkg/ha", "NGRZkg/ha", "PGRZkg/ha", "NCFRTkg/ha", "PCFRTkg/ha", "NRAINkg/ha", "NFIKkg/ha", "F-MNkg/ha", "A-MNkg/ha", "A-SNkg/ha", "F-MPkg/ha", "AO-LPkg/ha", "L-APkg/ha", "A-SPkg/ha", "DNITkg/ha", "NUPkg/ha", "PUPkg/ha", "ORGNkg/ha", "ORGPkg/ha", "SEDPkg/ha", "NSURQkg/ha", "NLATQkg/ha", "NO3Lkg/ha", "NO3GWkg/ha", "SOLPkg/ha", "P\_GWkg/ha", "W\_STRS", "TMP\_STRS", "N\_STRS", "P\_STRS", "BIOMt/ha", "LAI", "YLDt/ha", "BACTPct", "BACTLPct", "WTAB CLIm", "WTAB SOLm", "SNOmm", "CMUPkg/ha", "CMTOTkg/ha", "QTILEmm", "TNO3kg/ha", "LNO3kg/ha", "GW\_Q\_Dmm", "LATQCNTmm", "TVAPkg/ha"/)



- real \*8, dimension(:), allocatable **nperco**  
*nitrate percolation coefficient (0-1)*  
*0:concentration of nitrate in surface runoff is zero*  
*1:percolate has same concentration of nitrate as surface runoff*
- real \*8, dimension(:), allocatable **surlag**  
*Surface runoff lag time. This parameter is needed in subbasins where the time of concentration is greater than 1 day. SURLAG is used to create a "storage" for surface runoff to allow the runoff to take longer than 1 day to reach the subbasin outlet (days)*
- real \*8, dimension(:), allocatable **co\_p**
- real \*8, dimension(:), allocatable **cmn**  
*rate factor for humus mineralization on active organic N*
- real \*8, dimension(:), allocatable **phoskd**  
*phosphorus soil partitioning coefficient. Ratio of soluble phosphorus in surface layer attached to sediment to phosphorus dissolved in soil water*
- real \*8, dimension(:), allocatable **psp**  
*phosphorus availability index. The fraction of fertilizer P remaining in labile pool after initial rapid phase of P sorption (none)*
- real \*8, dimension(:), allocatable **sdnco**  
*denitrification threshold: fraction of field capacity triggering denitrification*
- real \*8 **r2adj\_bsn**  
*basinwide retention parameter adjustment factor (greater than 1)*
- real \*8 **pst\_kg**  
*amount of pesticide applied to HRU (kg/ha)*
- real \*8 **yield**  
*yield (dry weight) (kg)*
- real \*8 **burn\_frlb**  
*fraction of biomass and residue that burn(input in management file) range (0 - 1.0) (none)*
- real \*8 **yieldgrn**
- real \*8 **yieldbms**
- real \*8 **yieldtbr**
- real \*8 **yieldn**
- real \*8 **yieldp**
- real \*8 **hi\_bms**
- real \*8 **hi\_rsd**
- real \*8 **yieldrsd**
- real \*8, dimension(:), allocatable **l\_k1**
- real \*8, dimension(:), allocatable **l\_k2**
- real \*8, dimension(:), allocatable **l\_lambda**
- real \*8, dimension(:), allocatable **l\_beta**
- real \*8, dimension(:), allocatable **l\_gama**
- real \*8, dimension(:), allocatable **l\_harea**
- real \*8, dimension(:), allocatable **l\_vleng**
- real \*8, dimension(:), allocatable **l\_vslope**
- real \*8, dimension(:), allocatable **l\_ktc**
- real \*8, dimension(:), allocatable **biofilm\_mumax**
- real \*8, dimension(:), allocatable **biofilm\_kinv**
- real \*8, dimension(:), allocatable **biofilm\_klw**
- real \*8, dimension(:), allocatable **biofilm\_kla**
- real \*8, dimension(:), allocatable **biofilm\_cdet**
- real \*8, dimension(:), allocatable **biofilm\_bm**
- real \*8, dimension(:, :), allocatable **hru\_rufr**
- real \*8, dimension(:, :), allocatable **daru\_km**
- real \*8, dimension(:, :), allocatable **ru\_k**
- real \*8, dimension(:, :), allocatable **ru\_c**

- real \*8, dimension(:,:), allocatable **ru\_eiq**
- real \*8, dimension(:,:), allocatable **ru\_ovsl**
- real \*8, dimension(:,:), allocatable **ru\_a**
- real \*8, dimension(:,:), allocatable **ru\_ovs**
- real \*8, dimension(:,:), allocatable **ru\_ktc**
- real \*8, dimension(:), allocatable **gwq\_ru**
- real \*8, dimension(:), allocatable **qdayout**
- integer, dimension(:), allocatable **ils2**
- integer, dimension(:), allocatable **ils2flag**
- integer **ipest**  
*pesticide identification number from pest.dat (none)*
- integer **iru**
- integer **mru**
- integer **irch**
- integer **isub**
- integer **mhyd\_bsn**
- integer **ils\_nofig**
- integer **mhru1**
- integer, dimension(:), allocatable **mhyd1**
- integer, dimension(:), allocatable **irtun**
- real \*8 **wshd\_sepno3**
- real \*8 **wshd\_sepnh3**
- real \*8 **wshd\_seporgn**
- real \*8 **wshd\_sepfon**
- real \*8 **wshd\_seporgp**
- real \*8 **wshd\_sepfop**
- real \*8 **wshd\_sepsolp**
- real \*8 **wshd\_sepbod**
- real \*8 **wshd\_sepmm**
- integer, dimension(:), allocatable **isep\_hru**
- real \*8 **fixco**  
*nitrogen fixation coefficient*
- real \*8 **nfixmx**  
*maximum daily n-fixation (kg/ha)*
- real \*8 **res\_stlr\_co**  
*reservoir sediment settling coefficient*
- real \*8 **rsd\_covco**  
*residue cover factor for computing fraction of cover*
- real \*8 **vcrit**  
*critical velocity*
- real \*8 **wshd\_snob**  
*average amount of water stored in snow at the beginning of the simulation for the entire watershed (mm H2O)*
- real \*8 **wshd\_sw**  
*water in soil at beginning of simulation, or\ average amount of water stored in soil for the entire watershed, or\ difference between mass balance calculated from watershed averages and actual value for water in soil at end of simulation (goal is to have wshd\_sw = 0.) (mm H2O)*
- real \*8 **wshd\_pndfr**  
*fraction of watershed area which drains into ponds (none)*
- real \*8 **wshd\_pndsed**  
*total amount of suspended sediment in ponds in the watershed (metric tons), or mass balance discrepancy for pond sediment expressed as loading per unit hectare of drainage area (metric tons/ha)*
- real \*8 **wshd\_pndv**

- total volume of water in ponds in the watershed ( $m^3$ ), or mass balance discrepancy for pond water volume expressed as depth over drainage area (mm H<sub>2</sub>O)*
- real \*8 **percop**  
*pesticide percolation coefficient (0-1)*  
*0: concentration of pesticide in surface runoff is zero*  
*1: percolate has same concentration of pesticide as surface runoff*
  - real \*8 **wshd\_resfr**  
*fraction of watershed area that drains into reservoirs (none)*
  - real \*8 **wshd\_pndha**  
*watershed area in hectares which drains into ponds (ha)*
  - real \*8 **wshd\_resha**  
*watershed area in hectares which drains into reservoirs (ha)*
  - real \*8 **wshd\_wetfr**  
*fraction of watershed area which drains into wetlands (none)*
  - real \*8 **wshd\_fminp**  
*average annual amount of mineral P applied in watershed (kg P/ha)*
  - real \*8 **wshd\_fnh3**  
*average annual amount of NH<sub>3</sub>-N applied in watershed (kg N/ha)*
  - real \*8 **wshd\_fno3**  
*average annual amount of NO<sub>3</sub>-N applied in watershed (kg N/ha)*
  - real \*8 **wshd\_forgn**  
*average annual amount of organic N applied in watershed (kg N/ha)*
  - real \*8 **wshd\_ftotn**  
*average annual amount of N (mineral & organic) applied in watershed (kg N/ha)*
  - real \*8 **wshd\_forgp**  
*average annual amount of organic P applied in watershed (kg P/ha)*
  - real \*8 **wshd\_ftotp**  
*average annual amount of P (mineral & organic) applied in watershed (kg P/ha)*
  - real \*8 **wshd\_yldn**  
*amount of nitrogen removed from soil in watershed in the yield (kg N/ha)*
  - real \*8 **wshd\_yldp**  
*amount of phosphorus removed from soil in watershed in the yield (kg P/ha)*
  - real \*8 **wshd\_fixn**  
*average annual amount of nitrogen added to plant biomass via fixation (kg N/ha)*
  - real \*8 **wshd\_pup**  
*average annual amount of plant uptake of phosphorus (kg P/ha)*
  - real \*8 **wshd\_nstrs**  
*average annual number of nitrogen stress units in watershed (stress units)*
  - real \*8 **wshd\_pstrs**  
*average annual number of phosphorus stress units in watershed (stress units)*
  - real \*8 **wshd\_tstrs**  
*average annual number of temperature stress units in watershed (stress units)*
  - real \*8 **wshd\_wstrs**  
*average annual number of water stress units in watershed (stress units)*
  - real \*8 **wshd\_astrs**
  - real \*8 **ffcb**  
*initial soil water content expressed as a fraction of field capacity*
  - real \*8 **wshd\_dnit**  
*average annual amount of nitrogen lost from nitrate pool due to denitrification in watershed (kg N/ha)*
  - real \*8 **wshd\_hmn**  
*average annual amount of nitrogen moving from active organic to nitrate pool in watershed (kg N/ha)*

- real \*8 **wshd\_hmp**  
*average annual amount of phosphorus moving from organic to labile pool in watershed (kg P/ha)*
- real \*8 **wshd\_rmn**  
*average annual amount of nitrogen moving from fresh organic (residue) to nitrate and active organic pools in watershed (kg N/ha)*
- real \*8 **wshd\_rwn**  
*average annual amount of nitrogen moving from active organic to stable organic pool in watershed (kg N/ha)*
- real \*8 **wdpq**  
*die-off factor for persistent bacteria in soil solution (1/day)*
- real \*8 **wshd\_rmp**  
*average annual amount of phosphorus moving from fresh organic (residue) to labile and organic pools in watershed (kg P/ha)*
- real \*8 **wshd\_nitn**  
*average annual amount of nitrogen moving from the NH<sub>3</sub> to the NO<sub>3</sub> pool by nitrification in the watershed (kg N/ha)*
- real \*8 **wshd\_voln**  
*average annual amount of nitrogen lost by ammonia volatilization in watershed (kg N/ha)*
- real \*8 **wshd\_pal**  
*average annual amount of phosphorus moving from labile mineral to active mineral pool in watershed (kg P/ha)*
- real \*8 **wshd\_pas**  
*average annual amount of phosphorus moving from active mineral to stable mineral pool in watershed (kg P/ha)*
- real \*8 **wof\_p**  
*fraction of persistent bacteria on foliage that is washed off by a rainfall event (none)*
- real \*8 **wshd\_raino3**  
*average annual amount of NO<sub>3</sub> added to soil by rainfall in watershed (kg N/ha)*
- real \*8 **wshd\_plch**  
*average annual amount of phosphorus leached into second soil layer (kg P/ha)*
- real \*8 **ressedc**
- real \*8 **basno3f**
- real \*8 **basorgnf**
- real \*8 **wshd\_pinlet**
- real \*8 **wshd\_ptile**
- real \*8 **sftmp**  
*Snowfall temperature (deg C)*
- real \*8 **smfmn**  
*Minimum melt rate for snow during year (Dec. 21) where deg C refers to the air temperature. (mm/deg C/day)*
- real \*8 **smfmx**  
*Maximum melt rate for snow during year (June 21) where deg C refers to the air temperature. SMFMX and SMFMN allow the rate of snow melt to vary through the year. These parameters are accounting for the impact of soil temperature on snow melt. (mm/deg C/day)*
- real \*8 **smtmp**  
*Snow melt base temperature. Mean air temperature at which snow melt will occur. (deg C)*
- real \*8 **wgpg**  
*growth factor for persistent bacteria in soil solution (1/day)*
- real \*8 **basminpf**
- real \*8 **basorgpf**
- real \*8 **wdlpq**  
*die-off factor for less persistent bacteria in soil solution (1/day)*
- real \*8 **wshd\_ressed**  
*total amount of suspended sediment in reservoirs in the watershed (metric tons), or mass balance discrepancy for reservoir sediment expressed as loading per unit hectare of drainage area (metric tons/ha)*
- real \*8 **wshd\_resv**



- total volume of water in all reservoirs in the watershed ( $m^3$ ),  
or mass balance discrepancy for reservoir water volume expressed as depth over drainage area (mm H<sub>2</sub>O)*
- real \*8 **basminpi**
  - average amount of phosphorus initially in the mineral P pool in watershed soil (kg P/ha)*
- real \*8 **basno3i**
  - average amount of nitrogen initially in the nitrate pool in watershed soil (kg N/ha)*
- real \*8 **basorgni**
  - average amount of nitrogen initially in the organic N pool in watershed soil (kg N/ha)*
- real \*8 **wdps**
  - die-off factor for persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **wglpq**
  - growth factor for less persistent bacteria in soil solution (1/day)*
- real \*8 **basorgpi**
  - average amount of phosphorus initially in the organic P pool in watershed soil (kg P/ha)*
- real \*8 **peakr**
  - peak runoff rate for the day in HRU ( $m^3/s$ )*
- real \*8 **albdays**
  - albedo of ground for the day in HRU, the fraction of the solar radiation reflected at the soil surface back into space (none)*
- real \*8 **pndsedin**
  - sediment inflow to the pond from HRU (metric tons)*
- real \*8 **sw\_excess**
  - amount of water stored in soil layer on the current day that exceeds field capacity (gravity drained water) (mm H<sub>2</sub>O)*
- real \*8 **timp**
  - Snow pack temperature lag factor (0-1)  
1 = no lag (snow pack temp=current day air temp) as the lag factor goes to zero, the snow pack's temperature will be less influenced by the current day's air temperature.*
- real \*8 **wt\_shall**
  - shallow water table depth above the impervious layer (mm H<sub>2</sub>O)*
- real \*8 **wtabelo**
- real \*8 **tilep**
- real \*8 **sq\_rto**
- real \*8 **qtile**
  - drainage tile flow in HRU soil layer for the day (mm H<sub>2</sub>O)*
- real \*8 **infilpcp**
  - amount of precipitation that infiltrates into soil (enters soil) (mm H<sub>2</sub>O)*
- real \*8 **crk**
  - percolation due to crack flow (mm H<sub>2</sub>O)*
- real \*8 **fixn**
  - amount of nitrogen added to the plant biomass via fixation on the day in HRU (kg N/ha)*
- real \*8 **latlyr**
  - amount of water in lateral flow in layer in HRU for the day (mm H<sub>2</sub>O)*
- real \*8 **snofall**
  - amount of precipitation falling as freezing rain/snow on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **snomlt**
  - amount of water in snow melt for the day in HRU (mm H<sub>2</sub>O)*
- real \*8 **tloss**
  - amount of water removed from surface runoff via transmission losses on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **pndloss**
- real \*8 **wetloss**
- real \*8 **potloss**
- real \*8 **lpndloss**

- real \*8 **lwetloss**
- real \*8 **bioday**  
*biomass generated on current day in HRU (kg)*
- real \*8 **cfertn**  
*total amount of nitrogen applied to soil during continuous fertilizer operation in HRU on day (kg N/ha)*
- real \*8 **cfertp**  
*amount of phosphorus applied to soil during continuous fertilizer operation in HRU on day (kg P/ha)*
- real \*8 **fertn**  
*total amount of nitrogen applied to soil in HRU on day (kg N/ha)*
- real \*8 **sepcday**  
*micropore percolation from bottom of the soil layer on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **sol\_rd**  
*current rooting depth (mm)*
- real \*8 **sedrch**
- real \*8 **sepcrk**  
*water entering cracks in soil (mm H<sub>2</sub>O)*
- real \*8 **sepcrktot**
- real \*8 **fertno3**
- real \*8 **fertnh3**
- real \*8 **fertorgn**
- real \*8 **fertsolp**
- real \*8 **fertorgp**
- real \*8 **wgps**  
*growth factor for persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **qdfr**  
*fraction of water yield that is surface runoff (none)*
- real \*8 **fertp**  
*total amount of phosphorus applied to soil in HRU on day (kg P/ha)*
- real \*8 **grazn**  
*amount of nitrogen added to soil in grazing on the day in HRU (kg N/ha)*
- real \*8 **grazp**  
*amount of phosphorus added to soil in grazing on the day in HRU (kg P/ha)*
- real \*8 **soxy**  
*saturation dissolved oxygen concentration (mg/L)*
- real \*8 **sdti**
- real \*8 **rtwtr**
- real \*8 **ressa**
- real \*8 **wdlps**  
*die-off factor for less persistent bacteria absorbed to soil particles (1/day)*
- real \*8 **wglps**  
*growth factor for less persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **da\_km**  
*area of the watershed in square kilometers (km<sup>2</sup>)*
- real \*8 **rttime**
- real \*8 **rchdep**
- real \*8 **rtevp**
- real \*8 **rttlc**
- real \*8 **resflwi**
- real \*8 **wdprch**  
*die-off factor for persistent bacteria in streams (1/day)*
- real \*8 **resflwo**
- real \*8 **respcp**

- real \*8 **resev**
- real \*8 **ressep**
- real \*8 **ressedi**
- real \*8 **ressedo**
- real \*8 **dtot**
- real \*8 **pperco\_bsn**  
phosphorus percolation coefficient. Ratio of soluble phosphorus in surface to soluble phosphorus in percolate
- real \*8 **nperco\_bsn**  
basin nitrate percolation coefficient (0-1)  
0:concentration of nitrate in surface runoff is zero  
1:percolate has same concentration of nitrate as surface runoff
- real \*8 **rsdco**  
residue decomposition coefficient. The fraction of residue which will decompose in a day assuming optimal moisture, temperature, C:N ratio, and C:P ratio
- real \*8 **voltot**  
total volume of cracks expressed as depth per unit area (mm)
- real \*8 **phoskd\_bsn**
- real \*8 **msk\_x**  
weighting factor controlling relative importance of inflow rate and outflow rate in determining storage on reach
- real \*8 **volcrmin**  
minimum crack volume allowed in any soil layer (mm), or  
minimum soil volume in profile (mm)
- real \*8 **bactkdq**  
bacteria soil partitioning coefficient. Ratio of solution bacteria in surface layer to solution bacteria in runoff soluble and sorbed phase in surface runoff.
- real \*8 **wdpf**  
die-off factor for persistent bacteria on foliage (1/day)
- real \*8 **canev**  
amount of water evaporated from canopy storage (mm H2O)
- real \*8 **precipday**  
precipitation, or effective precipitation reaching soil surface, for the current day in HRU (mm H2O)
- real \*8 **uno3d**  
plant nitrogen deficiency for day in HRU (kg N/ha)
- real \*8 **usle**  
daily soil loss predicted with USLE equation (metric tons/ha)
- real \*8 **rcn**  
concentration of nitrogen in the rainfall (mg/L)
- real \*8 **surlag\_bsn**
- real \*8 **thbact**  
temperature adjustment factor for bacteria die-off/growth
- real \*8 **wlpq20**  
overall rate change for less persistent bacteria in soil solution (1/day)
- real \*8 **wlps20**  
overall rate change for less persistent bacteria adsorbed to soil particles (1/day)
- real \*8 **wpq20**  
overall rate change for persistent bacteria in soil solution (1/day)
- real \*8 **wps20**  
overall rate change for persistent bacteria adsorbed to soil particles (1/day)
- real \*8 **bactrop**  
persistent bacteria transported to main channel with surface runoff (# colonies/ha)
- real \*8 **bactsedp**  
persistent bacteria transported with sediment in surface runoff (# colonies/ha)

- real \*8 **wgpf**  
growth factor for persistent bacteria on foliage (1/day)
- real \*8 **bactlchlp**  
less persistent bacteria removed from soil surface layer by percolation (# colonies/ha)
- real \*8 **bactlchp**  
persistent bacteria removed from soil surface layer by percolation (# colonies/ha)
- real \*8 **enratio**  
enrichment ratio calculated for current day in HRU (none)
- real \*8 **pndpcp**  
precipitation on pond during day ( $m^3 H_2O$ )
- real \*8 **wetpcp**
- real \*8 **wetsep**  
seepage from wetland bottom for day ( $m^3 H_2O$ )
- real \*8 **pndev**  
evaporation from pond on day ( $m^3 H_2O$ )
- real \*8 **pndflwi**  
volume of water flowing into pond on day ( $m^3 H_2O$ )
- real \*8 **pndsedo**  
sediment leaving pond during day (metric tons)
- real \*8 **pndsep**  
seepage from pond on day ( $m^3 H_2O$ )
- real \*8 **wetev**  
evaporation from wetland for day ( $m^3 H_2O$ )
- real \*8 **wetflwi**  
volume of water flowing in wetland on day ( $m^3 H_2O$ )
- real \*8 **wetsedo**  
sediment loading from wetland for day (metric tons)
- real \*8 **da\_ha**  
drainage area of watershed in hectares (ha)
- real \*8 **pndflwo**  
volume of water flowing out of pond on day ( $m^3 H_2O$ )
- real \*8 **vpd**  
vapor pressure deficit (kPa)
- real \*8 **wetflwo**  
volume of water flowing out wetland on day ( $m^3 H_2O$ )
- real \*8 **wetsedi**  
sediment loading to wetland for day (metric tons)
- real \*8 **evlai**  
leaf area index at which no evaporation occurs. This variable is used in ponded HRUs where evaporation from the water surface is restricted by the plant canopy cover. Evaporation from the water surface equals potential ET when LAI = 0 and decreased linearly to 0 when LAI = EVLAI
- real \*8 **evrch**  
Reach evaporation adjustment factor. Evaporation from the reach is multiplied by EVRCH. This variable was created to limit the evaporation predicted in arid regions.
- real \*8 **wdlpf**  
die-off factor for less persistent bacteria on foliage (1/day)
- real \*8 **ep\_day**  
actual amount of transpiration that occurs on day in HRU (mm H<sub>2</sub>O)
- real \*8 **pet\_day**  
potential evapotranspiration on current day in HRU (mm H<sub>2</sub>O)
- real \*8 **bactrolp**

- less persistent bacteria transported to main channel with surface runoff (# colonies/ha)*

  - real \*8 **bactsedlp**
- less persistent bacteria transported with sediment in surface runoff (# colonies/ha)*

  - real \*8 **adj\_pkr**
- peak rate adjustment factor in the subbasin. Used in the MUSLE equation to account for impact of peak flow on erosion (none)*

  - real \*8 **n\_updis**
- nitrogen uptake distribution parameter. This parameter controls the amount of nitrogen removed from the different soil layer layers by the plant. In particular, this parameter allows the amount of nitrogen removed from the surface layer via plant uptake to be controlled. While the relationship between UBN and N removed from the surface layer is affected by the depth of the soil profile, in general, as UBN increases the amount of N removed from the surface layer relative to the amount removed from the entire profile increases*

  - real \*8 **nactfr**
- nitrogen active pool fraction. The fraction of organic nitrogen in the active pool (none)*

  - real \*8 **p\_updis**
- phosphorus uptake distribution parameter This parameter controls the amount of phosphorus removed from the different soil layers by the plant. In particular, this parameter allows the amount of phosphorus removed from the surface layer via plant uptake to be controlled. While the relationship between UBP and P uptake from the surface layer is affected by the depth of the soil profile, in general, as UBP increases the amount of P removed from the surface layer relative to the amount removed from the entire profile increases*

  - real \*8 **snoev**
- amount of water in snow lost through sublimation on current day in HRU (mm H2O)*

  - real \*8 **sno3up**
- amount of nitrate moving upward in the soil profile in watershed (kg N/ha)*

  - real \*8 **reactw**
- actual amount of evaporation (soil et) that occurs on day in HRU (mm H2O)*

  - real \*8 **es\_day**
- average annual change in the number of less persistent bacteria colonies in soil solution in watershed (# cfu/m<sup>2</sup>)*

  - real \*8 **sdiengrolpq**
- average annual change in the number of less persistent bacteria colonies on soil particles in watershed (# cfu/m<sup>2</sup>)*

  - real \*8 **sdiengrolps**
- average annual change in the number of persistent bacteria colonies in soil solution in watershed (# cfu/m<sup>2</sup>)*

  - real \*8 **sdiengropq**
- average annual change in the number of persistent bacteria colonies on soil particles in watershed (# cfu/m<sup>2</sup>)*

  - real \*8 **sdiengrops**
- fraction for less persistent bacteria on foliage that is washed off by a rainfall event (none)*

  - real \*8 **wof\_lp**
- maximum amount of transpiration (plant et) that can occur on day in HRU (mm H2O)*

  - real \*8 **ep\_max**
- average annual number of less persistent bacteria lost from soil surface layer by percolation (# cfu/m<sup>2</sup>)*

  - real \*8 **sbactrop**
  - real \*8 **sbactrolp**
  - real \*8 **sbactsedp**
  - real \*8 **sbactsedlp**
  - real \*8 **sbactlchlp**
- average annual number of persistent bacteria lost from soil surface layer by percolation (# cfu/m<sup>2</sup>)*

  - real \*8 **psp\_bsn**
  - real \*8 **rchwtr**
  - real \*8 **resuspst**
  - real \*8 **setlpst**
  - real \*8 **bsprev**
- surface runoff lagged from prior day of simulation (mm H2O)*

- real \*8 **bssprev**  
lateral flow lagged from prior day of simulation (mm H2O)
- real \*8 **spadyo**
- real \*8 **spadyev**
- real \*8 **spadysp**
- real \*8 **spadyrfv**
- real \*8 **spadyosp**
- real \*8 **qday**  
amount of surface runoff loading to main channel from HRU on current day (mm H2O)
- real \*8 **al5**  
fraction of total rainfall that occurs during 0.5h of highest intensity rain (none)
- real \*8 **no3pcp**  
nitrate added to the soil in rainfall (kg N/ha)
- real \*8 **pndsedc**  
net change in sediment in pond during day (metric tons)
- real \*8 **usle\_ei**  
USLE rainfall erosion index on day for HRU (100(ft-tn in)/(acre-hr))
- real \*8 **rcharea**
- real \*8 **volatpst**
- real \*8 **ubw**  
water uptake distribution parameter. This parameter controls the amount of water removed from the different soil layers by the plant. In particular, this parameter allows the amount of water removed from the surface layer via plant uptake to be controlled. While the relationship between UBW and H2O removed from the surface layer is affected by the depth of the soil profile, in general, as UBW increases the amount of water removed from the surface layer relative to the amount removed from the entire profile increases
- real \*8 **uobn**  
nitrogen uptake normalization parameter. This variable normalizes the nitrogen uptake so that the model can easily verify that upake from the different soil layers sums to 1.0
- real \*8 **uobp**  
phosphorus uptake normalization parameter. This variable normalizes the phosphorus uptake so that the model can easily verify that uptake from the different soil layers sums to 1.0
- real \*8 **uobw**  
water uptake normalization parameter. This variable normalizes the water uptake so that the model can easily verify that uptake from the different soil layers sums to 1.0
- real \*8 **wglpf**  
growth factor for less persistent bacteria on foliage (1/day)
- real \*8 **wetsedc**  
net change in sediment in wetland during day (metric tons)
- real \*8 **respesti**
- real \*8 **rcor**  
correction coefficient for generated rainfall to ensure that the annual means for generated and observed values are comparable (needed only if IDIST=1)
- real \*8 **rexp**  
value of exponent for mixed exponential rainfall distribution (needed only if IDIST=1)
- real \*8 **snocov1**  
1st shape parameter for snow cover equation. This parameter is determined by solving the equation for 50% snow cover
- real \*8 **snocov2**  
2nd shape parameter for snow cover equation. This parameter is determined by solving the equation for 95% snow cover
- real \*8 **snocovmx**  
Minimum snow water content that corresponds to 100% snow cover. If the snow water content is less than  $SNOC \leftrightarrow OVMX$ , then a certain percentage of the ground will be bare (mm H2O)

- real \*8 **lyrtile**  
*drainage tile flow in soil layer for day in HRU (mm H2O)*
- real \*8 **lyrtilex**
- real \*8 **sno50cov**  
*Fraction of SNOCOV MX that corresponds to 50% snow cover. SWAT assumes a nonlinear relationship between snow water and snow cover.*
- real \*8 **ai0**  
*ratio of chlorophyll-a to algal biomass (ug chla/mg alg)*
- real \*8 **ai1**  
*fraction of algal biomass that is nitrogen (mg N/mg alg)*
- real \*8 **ai2**  
*fraction of algal biomass that is phosphorus (mg P/mg alg)*
- real \*8 **ai3**  
*the rate of oxygen production per unit of algal photosynthesis (mg O2/mg alg)*
- real \*8 **ai4**  
*the rate of oxygen uptake per unit of algae respiration (mg O2/mg alg)*
- real \*8 **ai5**  
*the rate of oxygen uptake per unit of NH3 nitrogen oxidation (mg O2/mg N)*
- real \*8 **ai6**  
*the rate of oxygen uptake per unit of NO2 nitrogen oxidation (mg O2/mg N)*
- real \*8 **rhoq**  
*algal respiration rate (1/day or 1/hr)*
- real \*8 **tfact**  
*fraction of solar radiation computed in the temperature heat balance that is photosynthetically active*
- real \*8 **k\_l**  
*half-saturation coefficient for light (MJ/(m2\*hr))*
- real \*8 **k\_n**  
*Michaelis-menton half-saturation constant for nitrogen (mg N/L)*
- real \*8 **k\_p**  
*Michaelis-menton half saturation constant for phosphorus (mg P/L)*
- real \*8 **lambda0**  
*non-algal portion of the light extinction coefficient (1/m)*
- real \*8 **lambda1**  
*linear algal self-shading coefficient (1/(m\*ug chla/L))*
- real \*8 **lambda2**  
*nonlinear algal self-shading coefficient ((1/m)(ug chla/L)\*\*(-2/3))*
- real \*8 **mumax**  
*maximum specific algal growth rate (1/day or 1/hr)*
- real \*8 **p\_n**  
*algal preference factor for ammonia*
- real \*8 **rnum1**  
*variable to hold value for rnum1s(:) (none)*
- real \*8 **etday**  
*actual evapotranspiration occurring on day in HRU (mm H2O)*
- real \*8 **auton**  
*amount of nitrogen applied in auto-fert application (kg N/ha)*
- real \*8 **autop**  
*amount of phosphorus applied in auto-fert application (kg P/ha)*
- real \*8 **hmntl**  
*amount of nitrogen moving from active organic to nitrate pool in soil profile on current day in HRU (kg N/ha)*
- real \*8 **hmptl**

- amount of phosphorus moving from active organic to nitrate pool in soil profile on current day in HRU (kg P/ha)*

  - real \*8 **rmn2tl**
- amount of nitrogen moving from the fresh organic (residue) to the nitrate(80%) and active organic(20%) pools in soil profile on current day in HRU (kg N/ha)*

  - real \*8 **rwntl**
- amount of nitrogen moving from active organic to stable organic pool in soil profile on current day in HRU (kg N/ha)*

  - real \*8 **gwseep**
- amount of water recharging deep aquifer on current day in HRU (mm H2O)*

  - real \*8 **revapday**
- amount of water moving from the shallow aquifer into the soil profile or being taken up by plant roots in the shallow aquifer (mm H2O)*

  - real \*8 **rmp1tl**
- amount of phosphorus moving from the labile mineral pool to the active mineral pool in the soil profile on the current day in the HRU (kg P/ha)*

  - real \*8 **rmptl**
- amount of phosphorus moving from the fresh organic (residue) to the labile(80%) and organic(20%) pools in soil profile on current day in HRU (kg P/ha)*

  - real \*8 **roctl**
- amount of phosphorus moving from the active mineral pool to the stable mineral pool in the soil profile on the current day in the HRU (kg P/ha)*

  - real \*8 **wdntl**
- amount of nitrogen lost from nitrate pool by denitrification in soil profile on current day in HRU (kg N/ha)*

  - real \*8 **cmn\_bsn**
- die-off factor for less persistent bacteria in streams (1/day)*

  - real \*8 **reswtr**
- die-off factor for less persistent bacteria in streams (1/day)*

  - real \*8 **wdlprch**
- die-off factor for persistent bacteria in reservoirs (1/day)*

  - real \*8 **wdpres**
- die-off factor for persistent bacteria in reservoirs (1/day)*

  - real \*8 **petmeas**
- potential ET value read in for day (mm H2O)*

  - real \*8 **bury**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **difus**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **reactb**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **solpesto**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **wdlpres**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **sorpesto**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **spcon\_bsn**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **spexp\_bsn**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **solpesti**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **sorpesti**
- die-off factor for less persistent bacteria in reservoirs (1/day)*

  - real \*8 **msk\_co1**
- calibration coefficient to control impact of the storage time constant for the reach at bankfull depth (phi(10,:)) upon the storage time constant for the reach used in the Muskingum flow method*

  - real \*8 **msk\_co2**
- calibration coefficient to control impact of the storage time constant for the reach at 0.1 bankfull depth (phi(13,:)) upon the storage time constant for the reach used in the Muskingum flow method*

  - real \*8 **deepstp**
- depth of water in deep aquifer in HRU (mm H2O)*

  - real \*8 **shallstp**
- depth of water in shallow aquifer in HRU on previous day (mm H2O)*

  - real \*8 **snoprev**
- amount of water stored as snow on previous day (mm H2O)*



- real \*8 **swprev**  
*amount of water stored in soil profile in the HRU on the previous day (mm H2O)*
- real \*8 **ressolpo**
- real \*8 **resorgno**
- real \*8 **resorgpo**
- real \*8 **resno3o**
- real \*8 **reschlao**
- real \*8 **resno2o**
- real \*8 **potevmm**  
*volume of water evaporated from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **potflwo**  
*volume of water released to main channel from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **potpcpmm**  
*precipitation falling on pothole water body expressed as depth over HRU (mm H2O)*
- real \*8 **potsepmmm**  
*seepage from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **resnh3o**
- real \*8 **qdbank**
- real \*8 **bactminlp**  
*Threshold detection level for less persistent bacteria. When bacteria levels drop to this amount the model considers bacteria in the soil to be insignificant and sets the levels to zero (cfu/m<sup>2</sup>)*
- real \*8 **bactminp**  
*Threshold detection level for persistent bacteria. When bacteria levels drop to this amount the model considers bacteria in the soil to be insignificant and sets the levels to zero (cfu/m<sup>2</sup>)*
- real \*8 **trnsrch**  
*fraction of transmission losses from main channel that enter deep aquifer*
- real \*8 **wp20p\_plt**  
*overall rate change for persistent bacteria on foliage (1/day)*
- real \*8 **potsedo**  
*sediment released to main channel from HRU (metric tons/ha)*
- real \*8 **pest\_sol**
- real \*8 **bact\_swf**  
*fraction of manure containing active colony forming units (cfu)*
- real \*8 **bactmx**  
*bacteria percolation coefficient. Ratio of solution bacteria in surface layer to solution bacteria in percolate*
- real \*8 **cncoef**  
*plant ET curve number coefficient*
- real \*8 **wp20lp\_plt**  
*overall rate change for less persistent bacteria on foliage (1/day)*
- real \*8 **cdn\_bsn**
- real \*8 **sdnco\_bsn**
- real \*8 **bactmin**
- real \*8 **cn\_froz**  
*drainage coefficient (mm day<sup>-1</sup>)*
- real \*8 **dorm\_hr**  
*time threshold used to define dormant (hours)*
- real \*8 **smxco**  
*adjustment factor for max curve number s factor (0-1)*
- real \*8 **tb\_adj**  
*adjustment factor for subdaily unit hydrograph basetime*
- real \*8 **chla\_subco**  
*regional adjustment on sub chla\_a loading (fraction)*

- real \*8 [depimp\\_bsn](#)  
*depth to impervious layer. Used to model perched water tables in all HRUs in watershed (mm)*
- real \*8 [ddrain\\_bsn](#)  
*depth to the sub-surface drain (mm)*
- real \*8 [tdrain\\_bsn](#)  
*time to drain soil to field capacity (hours)*
- real \*8 **gdrain\_bsn**
- real \*8 **rch\_san**
- real \*8 **rch\_sil**
- real \*8 **rch\_cla**
- real \*8 **rch\_sag**
- real \*8 **rch\_lag**
- real \*8 **rch\_gra**
- real \*8 [hlife\\_ngw\\_bsn](#)  
*Half-life of nitrogen in groundwater? (days)*
- real \*8 **ch\_opco\_bsn**
- real \*8 **ch\_onco\_bsn**
- real \*8 [decr\\_min](#)  
*Minimum daily residue decay.*
- real \*8 [rcn\\_sub\\_bsn](#)  
*Concentration of nitrogen in the rainfall (mg/kg)*
- real \*8 **bc1\_bsn**
- real \*8 **bc2\_bsn**
- real \*8 **bc3\_bsn**
- real \*8 **bc4\_bsn**
- real \*8 **anion\_excl\_bsn**
- real \*8, dimension(:), allocatable [wat\\_tbl](#)  
*water table based on depth from soil surface (mm)*
- real \*8, dimension(:), allocatable **sol\_swpwt**
- real \*8, dimension(:, :), allocatable **vwt**
- real \*8 [re\\_bsn](#)  
*Effective radius of drains (range 3.0 - 40.0) (mm)*
- real \*8 [sdrain\\_bsn](#)  
*Distance bewtween two drain or tile tubes (range 7600.0 - 30000.0) (mm)*
- real \*8 **sstmaxd\_bsn**
- real \*8 [drain\\_co\\_bsn](#)  
*Drainage coefficient (range 10.0 - 51.0) (mm-day-1)*
- real \*8 [latksatf\\_bsn](#)  
*Multiplication factor to determine lateral ksat from SWAT ksat input value for HRU (range 0.01 - 4.0)*
- real \*8 [pc\\_bsn](#)  
*Pump capacity (def val = 1.042 mm h-1 or 25 mm day-1) (mm h-1)*
- integer [idlast](#)  
*number of days simulated in month (none)*
- integer **i\_subhw**
- integer **imgt**
- integer **iwtr**
- integer **ifrttyp**
- integer **mo\_atmo**
- integer **mo\_atmo1**
- integer **ifirstatmo**
- integer **iy\_r\_atmo**
- integer **iy\_r\_atmo1**

- integer **matmo**
- integer **mch**  
*maximum number of channels*
- integer **mcr**  
*maximum number of crops grown per year*
- integer **mcrdb**  
*maximum number of crops/landcover in database file (crop.dat)*
- integer **mfcst**  
*maximum number of forecast stations*
- integer **mfdv**  
*maximum number of fertilizers in fert.dat*
- integer **mhru**  
*maximum number of HRUs in watershed*
- integer **mhyd**  
*maximum number of hydrograph nodes*
- integer **mpdb**  
*maximum number of pesticides in pest.dat*
- integer **mrq**  
*maximum number of rainfall/temp gages (none)*
- integer **mcut**  
*maximum number of cuttings per year*
- integer **mqr**  
*maximum number of grazings per year*
- integer **mnr**  
*maximum number of years of rotation*
- integer **myr**  
*maximum number of years of simulation*
- integer **isubwq**  
*subbasin water quality code*  
*0 do not calculate algae/CBOD 1 calculate algae/CBOD drainmod tile equations*
- integer **fcst**
- integer **isproj**  
*special project code (none):*  
*1 test rewind (run simulation twice)*
- integer **nbyr**  
*number of calendar years simulated (none)*
- integer **irte**  
*water routing method (none):*  
*0 variable storage method*  
*1 Muskingum method*
- integer **nrch**  
*number of reaches in watershed (none)*
- integer **nres**  
*total number of reservoirs in watershed (none)*
- integer **nhru**  
*number of last HRU in previous subbasin or*  
*number of HRUs in watershed (none)*
- integer **i\_mo**  
*current month being simulated or month of next day of simulation (none)*
- integer **immo**  
*current cumulative month of simulation (none)*

- integer **mo**
- integer **wndsim**
  - wind speed input code (none)*
  - 1 measured data read for each subbasin*
  - 2 data simulated for each subbasin*
- integer **ihru**
  - HRU number (none)*
- integer **icode**
  - variable to hold value for icode(:) (none)*
- integer **ihout**
  - variable to hold value for ihouts(:) (none)*
- integer **inum1**
  - variable to hold value for inum1s(:) (subbasin number) (none)*
- integer **inum2**
  - variable to hold value for inum2s(:) (none)*
- integer **inum3**
  - variable to hold value for inum3s(:) (none)*
- integer **inum4**
  - variable to hold value for inum4s(:) (none)*
- integer **icfac**
  - icfac = 0 for C-factor calculation using Cmin (as described in manual)*
  - = 1 for new C-factor calculation from RUSLE (no minimum needed)*
- integer **inum5**
- integer **inum6**
- integer **inum7**
- integer **inum8**
- integer **mrech**
  - maximum number of rehour files*
- integer **nrgage**
  - number of raingage files (none)*
- integer **nrgfil**
  - number of rain gages per file (none)*
- integer **nrtot**
  - total number of rain gages (none)*
- integer **ntgage**
  - number of temperature gage files (none)*
- integer **ntgfil**
  - number of temperature gages per file (none)*
- integer **nttot**
  - total number of temperature gages (none)*
- integer **tmpsim**
  - temperature input code (none)*
  - 1 measured data read for each subbasin*
  - 2 data simulated for each subbasin*
- integer **icrk**
  - crack flow code*
  - 1: simulate crack flow in watershed*
- integer **irtpest**
  - number of pesticide to be routed through the watershed. Redefined to the sequence number of pesticide in NPNO(:) which is to be routed through the watershed (none)*
- integer **igropt**
  - Qual2E option for calculating the local specific growth rate of algae*
  - 1: multiplicative.*

- integer **lao**  
*Qual2E light averaging option. Qual2E defines four light averaging options. The only option currently available in SWAT is #2.*
- integer **npmx**  
*number of different pesticides used in the simulation (none)*
- integer **curyr**  
*current year in simulation (sequence) (none)*
- integer **itdrn**  
*tile drainage equations flag/code*  
*1 simulate tile flow using subroutine drains(wt\_shall)*  
*0 simulate tile flow using subroutine origtile(wt\_shall,d)*
- integer **iwt dn**  
*water table depth algorithms flag/code*  
*1 simulate wt\_shall using subroutine new water table depth routine*  
*0 simulate wt\_shall using subroutine original water table depth routine*
- integer **ismax**  
*maximum depressional storage selection flag/code (none)*  
*0 = static depressional storage (stmaxd) read from .bsn for the global value or .sdr for specific HRUs*  
*1 = dynamic storage (stmaxd) based on random roughness, tillage and cumulative rainfall intensity by [depstor.f90](#)*
- integer **iroutunit**  
*not being implemented in this version drainmod tile equations*
- integer **ires\_nut**
- integer **iclb**  
*auto-calibration flag*
- integer **mrecc**  
*maximum number of recnst files*
- integer **mrecd**  
*maximum number of recday files*
- integer **mrecm**  
*maximum number of recmon files*
- integer **mtill**  
*max number of tillage types in till.dat*
- integer **mudb**  
*maximum number of urban land types in urban.dat*
- integer **idist**  
*rainfall distribution code*  
*0 for skewed normal dist*  
*1 for mixed exponential distribution*
- integer **mrecy**  
*maximum number of recyear files*
- integer **nyskip**  
*number of years to skip output summarization and printing (none)*
- integer **slrsim**  
*solar radiation input code (none)*  
*1 measured data read for each subbasin*  
*2 data simulated for each subbasin*
- integer **ideg**  
*channel degradation code*  
*0: do not compute channel degradation*  
*1: compute channel degradation (downcutting and widening)*
- integer **ievent**  
*rainfall/runoff code (none)*  
*0 daily rainfall/curve number technique 1 sub-daily rainfall/Green&Ampt/hourly routing 3 sub-daily rainfall/↔ Green&Ampt/hourly routing*

- integer **ipet**  
*code for potential ET method (none)*  
*0 Priestley-Taylor method*  
*1 Penman/Monteith method*  
*2 Hargreaves method*  
*3 read in daily potential ET data*
- integer **iopera**
- integer **idaf**  
*beginning day of simulation (julian date)*
- integer **idal**  
*ending day of simulation (julian date)*
- integer **rhsim**  
*relative humidity input code (none)*  
*1 measured data read for each subbasin*  
*2 data simulated for each subbasin*
- integer **leapyr**  
*leap year flag (none)*  
*0 leap year*  
*1 regular year*
- integer **id1**  
*first day of simulation in current year (julian date)*
- integer **mo\_chk**  
*current month of simulation (none)*
- integer **nhtot**  
*total number of relative humidity records in file*
- integer **nstot**  
*total number of solar radiation records in file (none)*
- integer **nwtot**  
*total number of wind speed records in file*
- integer **ifirsts**  
*solar radiation data search code (none)*  
*0 first day of solar radiation data located in file*  
*1 first day of solar radiation data not located in file*
- integer **ifirsth**  
*relative humidity data search code (none)*  
*0 first day of relative humidity data located in file*  
*1 first day of relative humidity data not located in file*
- integer **ifirstw**  
*wind speed data search code (none)*  
*0 first day of wind speed data located in file*  
*1 first day of wind speed data not located in file*
- integer **icst**
- integer **ilog**  
*streamflow print code (none)*  
*0 print streamflow in reach*  
*1 print Log10 streamflow in reach*
- integer **itotr**  
*number of output variables printed (output.rch)*
- integer **iy**  
*current year of simulation (year)*
- integer **iwq**  
*stream water quality code*  
*0 do not model stream water quality*  
*1 model stream water quality (QUAL2E & pesticide transformations)*

- integer **iskip**  
*flag for calculations performed only for the first year of simulation (none)*
- integer **ifirstpet**  
*potential ET data search code (none)*  
*0 first day of potential ET data located in file*  
*1 first day of potential ET data not located in file*
- integer **iprp**  
*print code for output.pst file*  
*0 do not print pesticide output*  
*1 print pesticide output*
- integer **itotb**  
*number of output variables printed (output.sub)*
- integer **itots**  
*number of output variables printed (output.hru)*
- integer **itoth**  
*number of HRUs printed (output.hru/output.wtr)*
- integer **pcpsim**  
*rainfall input code (none)*  
*1 measured data read for each subbasin*  
*2 data simulated for each subbasin*
- integer **nd\_30**
- integer **iops**
- integer **iphr**
- integer **isto**
- integer **isol**
- integer **fcstcycles**  
*number of times forecast period is simulated (using different weather generator seeds each time)*
- integer **fcstday**  
*beginning date of forecast period (julian date)*
- integer **fcstyr**  
*beginning year of forecast period*
- integer **iscen**  
*scenarios counter*
- integer **subtot**  
*number of subbasins in watershed (none)*
- integer **ogen**
- integer **mapp**  
*maximum number of applications*
- integer **mlyr**  
*maximum number of soil layers*
- integer **mpst**  
*max number of pesticides used in wshed*
- integer **mres**  
*maximum number of reservoirs*
- integer **msub**  
*maximum number of subbasins*
- integer **igen**  
*random number generator seed code (none):*  
*0: use default numbers*  
*1: generate new numbers in every simulation*
- integer **iprint**  
*print code (none): 0=monthly, 1=daily, 2=annually*
- integer **iida**

- day being simulated (current julian date) (julian date)*

  - integer **icn**

*CN method flag (for testing alternative method):*  
*0 use traditional SWAT method which bases CN on soil moisture*  
*1 use alternative method which bases CN on plant ET*  
*2 use traditional SWAT method which bases CN on soil moisture but retention is adjusted for mildly-sloped tiled-drained watersheds.*
  - integer **ised\_det**

*max half-hour rainfall fraction calc option:*  
*0 generate max half-hour rainfall fraction from triangular distribution*  
*1 use monthly mean max half-hour rainfall fraction*
  - integer **fcstcnt**
  - integer **mtran**
  - integer **idtill**
  - integer, dimension(100) **ida\_lup**
  - integer, dimension(100) **iy\_r\_lup**
  - integer **no\_lup**
  - integer **no\_up**
  - integer **nostep**
  - character(len=8) **date**

*date simulation is performed where leftmost eight characters are set to a value of yyyyymmdd, where yyyy is the year, mm is the month and dd is the day*
  - character(len=10) **time**

*time simulation is performed where leftmost ten characters are set to a value of hhmmss.sss, where hh is the hour, mm is the minutes and ss.sss is the seconds and milliseconds*
  - character(len=5) **zone**

*time difference with respect to Coordinated Universal Time (ie Greenwich Mean Time)*
  - character(len=13) **calfile**

*name of file containing calibration parameters*
  - character(len=13) **rhfile**

*relative humidity file name (.hmd)*
  - character(len=13) **slrfile**

*solar radiation file name (.slr)*
  - character(len=13) **wndfile**

*wind speed file name (.wnd)*
  - character(len=13) **petfile**

*potential ET file name (.pet)*
  - character(len=13) **atmofile**
  - character(len=13) **lucfile**
  - character(len=13) **septdb**

*name of septic tank database file (septwq1.dat)*
  - character(len=13) **dpd\_file**
  - character(len=13) **wpd\_file**
  - character(len=13) **rib\_file**
  - character(len=13) **sfb\_file**
  - character(len=13) **lid\_file**
  - integer, dimension(9) **idg**

*array location of random number seed used for a given process*
  - integer, dimension(:), allocatable **ifistr**
  - integer, dimension(:), allocatable **ifisthr**
  - integer, dimension(8) **values**



*values(1): year simulation is performed*  
*values(2): month simulation is performed*  
*values(3): day in month simulation is performed*  
*values(4): time difference with respect to Coordinated Universal Time (ie Greenwich Mean Time)*  
*values(5): hour simulation is performed*  
*values(6): minute simulation is performed*  
*values(7): second simulation is performed*  
*values(8): millisecond simulation is performed*

- integer, dimension(13) **ndays**  
*julian date for last day of preceding month (where the array location is the number of the month). The dates are for leap years (julian date)*
- integer **mapex**
- real \*8, dimension(:), allocatable **flodaya**
- real \*8, dimension(:), allocatable **seddaya**
- real \*8, dimension(:), allocatable **orgndaya**
- real \*8, dimension(:), allocatable **orgpdaya**
- real \*8, dimension(:), allocatable **no3daya**
- real \*8, dimension(:), allocatable **minpdaya**
- real \*8, dimension(:), allocatable **hi\_targ**  
*harvest index target of cover defined at planting ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable **bio\_targ**  
*biomass target (kg/ha)*
- real \*8, dimension(:), allocatable **tnyld**  
*modifier for autofertilization target nitrogen content for plant (kg N/kg yield)*
- integer, dimension(:), allocatable **idapa**
- integer, dimension(:), allocatable **iypa**
- integer, dimension(:), allocatable **ifirsta**
- integer, dimension(100) **mo\_transb**
- integer, dimension(100) **mo\_transe**
- integer, dimension(100) **ih\_tran**
- integer **msdb**  
*maximum number of sept wq data database (none)*
- integer **iseptic**
- real \*8, dimension(:), allocatable **sptqs**  
*flow rate of the septic tank effluent per capita (m3/d)*
- real \*8, dimension(:), allocatable **percp**
- real \*8, dimension(:), allocatable **sptbodconcs**  
*Biological Oxygen Demand of the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptssconcs**  
*concentration of total suspended solid in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **spttnconcs**  
*concentration of total nitrogen in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptnh4concs**  
*concentration of total phosphorus of the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptno3concs**  
*concentration of nitrate in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptno2concs**  
*concentration of nitrite in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptorgnconcs**  
*concentration of organic nitrogen in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **spttpconcs**  
*concentration of total phosphorus in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **sptminps**

- concentration of mineral phosphorus in the septic tank effluent (mg/l)*

  - real \*8, dimension(:), allocatable [sptorgps](#)
- concentration of organic phosphorus in the septic tank effluent (mg/l)*

  - real \*8, dimension(:), allocatable [sptfcolis](#)
- concentration of the faecal coliform in the septic tank effluent (cfu/100ml)*

  - real \*8, dimension(:), allocatable **failyr**
  - real \*8, dimension(:), allocatable **qstemm**
  - real \*8, dimension(:), allocatable [bio\\_bod](#)
- BOD concentration in biozone (kg/ha)*

  - real \*8, dimension(:), allocatable [biom](#)
- biomass of live bacteria in biozone (kg/ha)*

  - real \*8, dimension(:), allocatable [rbiom](#)
- daily change in biomass of live bacteria (kg/ha)*

  - real \*8, dimension(:), allocatable **bio\_amn**
  - real \*8, dimension(:), allocatable [fcoli](#)
- concentration of the fecal coliform in the biozone septic tank effluent (cfu/100ml)*

  - real \*8, dimension(:), allocatable **bio\_ntr**
  - real \*8, dimension(:), allocatable **bz\_perc**
  - real \*8, dimension(:), allocatable [sep\\_cap](#)
- number of permanent residents in the house (none)*

  - real \*8, dimension(:), allocatable [plqm](#)
- plaque in biozone (kg/ha)*

  - real \*8, dimension(:), allocatable **bz\_area**
  - real \*8, dimension(:), allocatable [bz\\_z](#)
- depth of biozone layer (mm)*

  - real \*8, dimension(:), allocatable [bz\\_thk](#)
- thickness of biozone (mm)*

  - real \*8, dimension(:), allocatable [bio\\_bd](#)
- density of biomass (kg/m<sup>3</sup>)*

  - real \*8, dimension(:), allocatable [cmup\\_kgh](#)
- current soil carbon for first soil layer (kg/ha)*

  - real \*8, dimension(:), allocatable [cmtot\\_kgh](#)
- current soil carbon integrated - aggregating (kg/ha)*

  - real \*8, dimension(:), allocatable [coeff\\_denitr](#)
- denitrification rate coefficient (none)*

  - real \*8, dimension(:), allocatable [coeff\\_bod\\_dc](#)
- BOD decay rate coefficient (m<sup>3</sup>/day)*

  - real \*8, dimension(:), allocatable [coeff\\_bod\\_conv](#)
- BOD to live bacteria biomass conversion factor (none)*

  - real \*8, dimension(:), allocatable [coeff\\_fc1](#)
- field capacity calibration parameter 1 (none)*

  - real \*8, dimension(:), allocatable [coeff\\_fc2](#)
- field capacity calibration parameter 2 (none)*

  - real \*8, dimension(:), allocatable [coeff\\_fecal](#)
- fecal coliform bacteria decay rate coefficient (m<sup>3</sup>/day)*

  - real \*8, dimension(:), allocatable [coeff\\_mrt](#)
- mortality rate coefficient (none)*

  - real \*8, dimension(:), allocatable [coeff\\_nitr](#)
- nitrification rate coefficient (none)*

  - real \*8, dimension(:), allocatable [coeff\\_plq](#)
- conversion factor for plaque from TDS (none)*

- real \*8, dimension(:), allocatable [coeff\\_rsp](#)  
*respiration rate coefficient (none)*
- real \*8, dimension(:), allocatable [coeff\\_slg1](#)  
*slough-off calibration parameter (none)*
- real \*8, dimension(:), allocatable [coeff\\_slg2](#)  
*slough-off calibration parameter (none)*
- real \*8, dimension(:), allocatable **coeff\_pdistrb**
- real \*8, dimension(:), allocatable **coeff\_solpslp**
- real \*8, dimension(:), allocatable **coeff\_solpintc**
- real \*8, dimension(:), allocatable **coeff\_psorpmx**
- integer, dimension(:), allocatable [isep\\_typ](#)  
*septic system type (none)*
- integer, dimension(:), allocatable [i\\_sep](#)  
*soil layer where biozone exists (none)*
- integer, dimension(:), allocatable [isep\\_opt](#)  
*septic system operation flag (1=active, 2=failing, 3 or 0=not operated) (none)*
- integer, dimension(:), allocatable **sep\_tsincefail**
- integer, dimension(:), allocatable **isep\_tfail**
- integer, dimension(:), allocatable **isep\_iyr**
- integer, dimension(:), allocatable **sep\_strm\_dist**
- integer, dimension(:), allocatable **sep\_den**
- real \*8, dimension(:), allocatable **sol\_sumno3**
- real \*8, dimension(:), allocatable **sol\_sumsolp**
- real \*8, dimension(:), allocatable **strsw\_sum**
- real \*8, dimension(:), allocatable **strstmp\_sum**
- real \*8, dimension(:), allocatable **strsn\_sum**
- real \*8, dimension(:), allocatable **strsp\_sum**
- real \*8, dimension(:), allocatable **strsa\_sum**
- real \*8, dimension(:), allocatable **spill\_hru**
- real \*8, dimension(:), allocatable **tile\_out**
- real \*8, dimension(:), allocatable **hru\_in**
- real \*8, dimension(:), allocatable **spill\_precip**
- real \*8, dimension(:), allocatable **pot\_seep**
- real \*8, dimension(:), allocatable **pot\_evap**
- real \*8, dimension(:), allocatable **pot\_sedin**
- real \*8, dimension(:), allocatable [pot\\_solp](#)  
*soluble P loss rate in the pothole (.01 - 0.5) (1/d)*
- real \*8, dimension(:), allocatable **pot\_solpi**
- real \*8, dimension(:), allocatable **pot\_orgp**
- real \*8, dimension(:), allocatable **pot\_orgpi**
- real \*8, dimension(:), allocatable **pot\_orgn**
- real \*8, dimension(:), allocatable **pot\_orgni**
- real \*8, dimension(:), allocatable **pot\_mps**
- real \*8, dimension(:), allocatable **pot\_mpsi**
- real \*8, dimension(:), allocatable **pot\_mpa**
- real \*8, dimension(:), allocatable **pot\_mpai**
- real \*8, dimension(:), allocatable **pot\_no3i**
- real \*8, dimension(:), allocatable **precip\_in**
- real \*8, dimension(:), allocatable **tile\_sedo**
- real \*8, dimension(:), allocatable **tile\_no3o**
- real \*8, dimension(:), allocatable **tile\_solpo**
- real \*8, dimension(:), allocatable **tile\_orgno**
- real \*8, dimension(:), allocatable **tile\_orgpo**

- real \*8, dimension(:), allocatable **tile\_minpso**
- real \*8, dimension(:), allocatable **tile\_minpao**
- integer **ia\_b**
- integer **ihumus**
- integer **itemp**
- integer **isnow**
- integer, dimension(46) **ipdvar**  
*output variable codes for output.rch file (none)*
- integer, dimension(mhruo) **ipdvas**  
*output variable codes for output.hru file (none)*
- integer, dimension(msub0) **ipdvab**  
*output variable codes for output.sub file (none)*
- integer, dimension(:), allocatable **ipdhru**  
*HRUs whose output information will be printed to the output.hru and output.wtr files.*
- real \*8, dimension(mstdo) **wshddayo**  
*wshddayo(1) average amount of precipitation in watershed for the day (mm H2O)*  
*wshddayo(3) surface runoff in watershed for day (mm H2O)*  
*wshddayo(4) lateral flow contribution to streamflow in watershed for day (mm H2O)*  
*wshddayo(5) water percolation past bottom of soil profile in watershed for day (mm H2O)*  
*wshddayo(6) water yield to streamflow from HRUs in watershed for day (mm H2O)*  
*wshddayo(7) actual evapotranspiration in watershed for day (mm H2O)*  
*wshddayo(12) sediment yield from HRUs in watershed for day (metric tons or metric tons/ha)*  
*wshddayo(35) amount of water stored in soil profile in watershed at end of day (mm H2O)*  
*wshddayo(41) organic P loading to stream in watershed for day (kg P/ha)*  
*wshddayo(42) nitrate loading to stream in surface runoff in watershed for day (kg N/ha)*  
*wshddayo(43) soluble P loading to stream in watershed for day (kg P/ha)*  
*wshddayo(44) plant uptake of N in watershed for day (kg N/ha)*  
*wshddayo(45) nitrate loading to stream in lateral flow in watershed for day (kg N/ha)*  
*wshddayo(46) nitrate percolation past bottom of soil profile in watershed for day (kg N/ha)*  
*wshddayo(104) groundwater contribution to stream in watershed on day (mm H2O)*  
*wshddayo(108) potential evapotranspiration in watershed on day (mm H2O)*  
*wshddayo(109) drainage tile flow contribution to stream in watershed on day (mm H2O)*
- real \*8, dimension(mstdo) **wshdmono**  
*watershed monthly output array (see definitions for wshddayo array elements) (varies)*  
*wshdmono(1) average amount of precipitation in watershed for the month (mm H2O)*  
*wshdmono(3) surface runoff in watershed for month (mm H2O)*  
*wshdmono(4) lateral flow contribution to streamflow in watershed for month (mm H2O)*  
*wshdmono(5) water percolation past bottom of soil profile in watershed for month (mm H2O)*  
*wshdmono(6) water yield to streamflow from HRUs in watershed for month (mm H2O)*  
*wshdmono(7) actual evapotranspiration in watershed for month (mm H2O)*  
*wshdmono(8) average maximum temperature in watershed for the month (deg C)*  
*wshdmono(9) average minimum temperature in watershed for the month (deg C)*  
*wshdmono(12) sediment yield from HRUs in watershed for the month (metric tons)*  
*wshdmono(39) freezing rain/snow fall in watershed for the month (mm H2O)*  
*wshdmono(40) organic N loading to stream in watershed for the month (kg N/ha)*  
*wshdmono(41) organic P loading to stream in watershed for the month (kg P/ha)*  
*wshdmono(42) nitrate loading to stream in surface runoff in watershed for the month (kg N/ha)*  
*wshdmono(43) soluble P loading to stream in watershed for the month (kg P/ha)*  
*wshdmono(44) plant uptake of N in watershed for the month (kg N/ha)*  
*wshdmono(45) nitrate loading to stream in lateral flow in watershed for the month (kg N/ha)*  
*wshdmono(46) nitrate percolation past bottom of soil profile in watershed for the month (kg N/ha)*  
*wshdmono(104) groundwater contribution to stream in watershed for the month (mm H2O)*  
*wshdmono(108) potential evapotranspiration in watershed for the month (mm H2O)*  
*wshdmono(109) drainage tile flow contribution to stream in watershed for the month (mm H2O)*
- real \*8, dimension(mstdo) **wshdyro**  
*watershed annual output array (varies)*  
*wshdyro(1) average amount of precipitation in watershed for the year (mm H2O)*  
*wshdyro(3) surface runoff in watershed for year (mm H2O)*  
*wshdyro(4) lateral flow contribution to streamflow in watershed for year (mm H2O)*

- wshdyro(5)* water percolation past bottom of soil profile in watershed for year (mm H<sub>2</sub>O)
- wshdyro(6)* water yield to streamflow from HRUs in watershed for year (mm H<sub>2</sub>O)
- wshdyro(7)* actual evapotranspiration in watershed for year (mm H<sub>2</sub>O)
- wshdyro(8)* average maximum temperature in watershed for the year (deg C)
- wshdyro(9)* average minimum temperature in watershed for the year (deg C)
- wshdyro(12)* sediment yield from HRUs in watershed for the year (metric tons)
- wshdyro(40)* organic N loading to stream in watershed for the year (kg N/ha)
- wshdyro(41)* organic P loading to stream in watershed for the year (kg P/ha)
- wshdyro(42)* nitrate loading to stream in surface runoff in watershed for the year (kg N/ha)
- wshdyro(43)* soluble P loading to stream in watershed for the year (kg P/ha)
- wshdyro(44)* plant uptake of N in watershed for the year
- wshdyro(45)* nitrate loading to stream in lateral flow in watershed for the year (kg N/ha)
- wshdyro(46)* nitrate percolation past bottom of soil profile in watershed for the year (kg N/ha)
- wshdyro(104)* groundwater contribution to stream in watershed for the year (mm H<sub>2</sub>O)
- wshdyro(108)* potential evapotranspiration in watershed for the year (mm H<sub>2</sub>O)
- wshdyro(109)* drainage tile flow contribution to stream in watershed for the year (mm H<sub>2</sub>O)
- real \*8, dimension(16) **fcstaao**
- real \*8, dimension(**mstdo**) **wshdaao**
  - watershed average annual output array (varies)*
  - wshdaao(1)* precipitation in watershed (mm H<sub>2</sub>O)
  - wshdaao(3)* surface runoff loading to main channel in watershed (mm H<sub>2</sub>O)
  - wshdaao(4)* lateral flow loading to main channel in watershed (mm H<sub>2</sub>O)
  - wshdaao(5)* percolation of water out of root zone in watershed (mm H<sub>2</sub>O)
  - wshdaao(7)* actual evapotranspiration in watershed (mm H<sub>2</sub>O)
  - wshdaao(13)* sediment loading to ponds in watershed (metric tons)
  - wshdaao(14)* sediment loading from ponds in watershed (metric tons)
  - wshdaao(15)* net change in sediment level in ponds in watershed (metric tons)
  - wshdaao(19)* evaporation from ponds in watershed (m<sup>3</sup> H<sub>2</sub>O)
  - wshdaao(20)* seepage from ponds in watershed (m<sup>3</sup> H<sub>2</sub>O)
  - wshdaao(21)* precipitation on ponds in watershed (m<sup>3</sup> H<sub>2</sub>O)
  - wshdaao(22)* volume of water entering ponds in watershed (m<sup>3</sup> H<sub>2</sub>O)
  - wshdaao(23)* volume of water leaving ponds in watershed (m<sup>3</sup> H<sub>2</sub>O)
  - wshdaao(38)* transmission losses in watershed (mm H<sub>2</sub>O)
- real \*8, dimension(:, :), allocatable **wpstdayo**
- real \*8, dimension(:, :), allocatable **wpstmono**
- real \*8, dimension(:, :), allocatable **wpstyro**
- real \*8, dimension(:, :), allocatable **bio\_hv**
  - harvested biomass (dry weight) (kg/ha)*
- real \*8, dimension(:, :), allocatable **yldkg**
  - yield (dry weight) by crop type in the HRU (kg/ha)*
- real \*8, dimension(:, :), allocatable **rchmono**
  - reach monthly output array (varies)*
  - rchmono(1,:) flow into reach during month (m<sup>3</sup>/s)*
  - rchmono(2,:) flow out of reach during month (m<sup>3</sup>/s)*
  - rchmono(3,:) sediment transported into reach during month (metric tons)*
  - rchmono(4,:) sediment transported out of reach during month (metric tons)*
  - rchmono(5,:) sediment concentration in outflow during month (mg/L)*
  - rchmono(6,:) organic N transported into reach during month (kg N)*
  - rchmono(7,:) organic N transported out of reach during month (kg N)*
  - rchmono(8,:) organic P transported into reach during month (kg P)*
  - rchmono(9,:) organic P transported out of reach during month (kg P)*
  - rchmono(10,:) evaporation from reach during month (m<sup>3</sup>/s)*
  - rchmono(11,:) transmission losses from reach during month (m<sup>3</sup>/s)*
  - rchmono(12,:) conservative metal #1 transported out of reach during month (kg)*
  - rchmono(13,:) conservative metal #2 transported out of reach during month (kg)*
  - rchmono(14,:) conservative metal #3 transported out of reach during month (kg)*
  - rchmono(15,:) nitrate transported into reach during month (kg N)*
  - rchmono(16,:) nitrate transported out of reach during month (kg N)*
  - rchmono(17,:) soluble P transported into reach during month (kg P)*
  - rchmono(18,:) soluble P transported out of reach during month (kg P)*
  - rchmono(19,:) soluble pesticide transported into reach during month (mg pst)*

*rchmono(20,:)* soluble pesticide transported out of reach during month (mg pst)  
*rchmono(21,:)* sorbed pesticide transported into reach during month (mg pst)  
*rchmono(22,:)* sorbed pesticide transported out of reach during month (mg pst)  
*rchmono(23,:)* amount of pesticide lost through reactions in reach during month (mg pst)  
*rchmono(24,:)* amount of pesticide lost through volatilization from reach during month (mg pst)  
*rchmono(25,:)* amount of pesticide settling out of reach to bed sediment during month (mg pst)  
*rchmono(26,:)* amount of pesticide resuspended from bed sediment to reach during month (mg pst)  
*rchmono(27,:)* amount of pesticide diffusing from reach to bed sediment during month (mg pst)  
*rchmono(28,:)* amount of pesticide in sediment layer lost through reactions during month (mg pst)  
*rchmono(29,:)* amount of pesticide in sediment layer lost through burial during month (mg pst)  
*rchmono(30,:)* chlorophyll-a transported into reach during month (kg chla)  
*rchmono(31,:)* chlorophyll-a transported out of reach during month (kg chla)  
*rchmono(32,:)* ammonia transported into reach during month (kg N)  
*rchmono(33,:)* ammonia transported out of reach during month (kg N)  
*rchmono(34,:)* nitrite transported into reach during month (kg N)  
*rchmono(35,:)* nitrite transported out of reach during month (kg N)  
*rchmono(36,:)* CBOD transported into reach during month (kg O2)  
*rchmono(37,:)* CBOD transported out of reach during month (kg O2)  
*rchmono(38,:)* dissolved oxygen transported into reach during month (kg O2)  
*rchmono(39,:)* dissolved oxygen transported out of reach during month (kg O2)  
*rchmono(40,:)* persistent bacteria transported out of reach during month (kg bact)  
*rchmono(41,:)* less persistent bacteria transported out of reach during month (kg bact)  
*rchmono(43,:)* total N (org N + no3 + no2 + nh4 outs) (kg)  
*rchmono(44,:)* total P (org P + sol p outs) (kg)

- real \*8, dimension(:,,:), allocatable **rchyro**

*reach* annual output array (varies)  
*rchyro(1,:)* flow into reach during year ( $m^3/s$ )  
*rchyro(2,:)* flow out of reach during year ( $m^3/s$ )  
*rchyro(3,:)* sediment transported into reach during year (metric tons)  
*rchyro(4,:)* sediment transported out of reach during year (metric tons)  
*rchyro(5,:)* sediment concentration in outflow during year (mg/L)  
*rchyro(6,:)* organic N transported into reach during year (kg N)  
*rchyro(7,:)* organic N transported out of reach during year (kg N)  
*rchyro(8,:)* organic P transported into reach during year (kg P)  
*rchyro(9,:)* organic P transported out of reach during year (kg P)  
*rchyro(10,:)* evaporation from reach during year ( $m^3/s$ )  
*rchyro(11,:)* transmission losses from reach during year ( $m^3/s$ )  
*rchyro(12,:)* conservative metal #1 transported out of reach during year (kg)  
*rchyro(13,:)* conservative metal #2 transported out of reach during year (kg)  
*rchyro(14,:)* conservative metal #3 transported out of reach during year (kg)  
*rchyro(15,:)* nitrate transported into reach during year (kg N)  
*rchyro(16,:)* nitrate transported out of reach during year (kg N)  
*rchyro(17,:)* soluble P transported into reach during year (kg P)  
*rchyro(18,:)* soluble P transported out of reach during year (kg P)  
*rchyro(19,:)* soluble pesticide transported into reach during year (mg pst)  
*rchyro(20,:)* soluble pesticide transported out of reach during year (mg pst)  
*rchyro(21,:)* sorbed pesticide transported into reach during year (mg pst)  
*rchyro(22,:)* sorbed pesticide transported out of reach during year (mg pst)  
*rchyro(23,:)* amount of pesticide lost through reactions in reach during year!> (mg pst)  
*rchyro(24,:)* amount of pesticide lost through volatilization from reach during year (mg pst)  
*rchyro(25,:)* amount of pesticide settling out of reach to bed sediment during year (mg pst)  
*rchyro(26,:)* amount of pesticide resuspended from bed sediment to reach during year (mg pst)  
*rchyro(27,:)* amount of pesticide diffusing from reach to bed sediment during year (mg pst)  
*rchyro(28,:)* amount of pesticide in sediment layer lost through reactions during year (mg pst)  
*rchyro(29,:)* amount of pesticide in sediment layer lost through burial during year (mg pst)  
*rchyro(30,:)* chlorophyll-a transported into reach during year (kg chla)  
*rchyro(31,:)* chlorophyll-a transported out of reach during year (kg chla)  
*rchyro(32,:)* ammonia transported into reach during year (kg N)  
*rchyro(33,:)* ammonia transported out of reach during year (kg N)  
*rchyro(34,:)* nitrite transported into reach during year (kg N)  
*rchyro(35,:)* nitrite transported out of reach during year (kg N)  
*rchyro(36,:)* CBOD transported into reach during year (kg O2)  
*rchyro(37,:)* CBOD transported out of reach during year (kg O2)



*rchyro(38,:) dissolved oxygen transported into reach during year (kg O2)*  
*rchyro(39,:) dissolved oxygen transported out of reach during year (kg O2)*  
*rchyro(40,:) persistent bacteria transported out of reach during year (kg bact)*  
*rchyro(41,:) less persistent bacteria transported out of reach during year (kg bact)*

- real \*8, dimension(:,:), allocatable **wpstaao**
- real \*8, dimension(:,:), allocatable **hrumono**

*HRU monthly output data array (varies)*

*hrumono(1,:) precipitation in HRU during month (mm H2O)*  
*hrumono(2,:) amount of precipitation falling as freezing rain/snow in HRU during month (mm H2O)*  
*hrumono(3,:) amount of snow melt in HRU during month (mm H2O)*  
*hrumono(4,:) amount of surface runoff to main channel from HRU during month (ignores impact of transmission losses) (mm H2O)*  
*hrumono(5,:) amount of lateral flow contribution to main channel from HRU during month (mm H2O)*  
*hrumono(6,:) amount of groundwater flow contribution to main channel from HRU during month (mm H2O)*  
*hrumono(7,:) amount of water moving from shallow aquifer to plants or soil profile in HRU during month (mm H2O)*  
*hrumono(8,:) amount of water recharging deep aquifer in HRU during month (mm H2O)*  
*hrumono(9,:) total amount of water entering both aquifers from HRU during month (mm H2O)*  
*hrumono(10,:) water yield (total amount of water entering main channel) from HRU during month (mm H2O)*  
*hrumono(11,:) amount of water percolating out of the soil profile and into the vadose zone in HRU during month (mm H2O)*  
*hrumono(12,:) actual evapotranspiration in HRU during month (mm H2O)*  
*hrumono(13,:) amount of transmission losses from tributary channels in HRU for month (mm H2O)*  
*hrumono(14,:) sediment yield from HRU for month (metric tons/ha)*  
*hrumono(15,:) actual amount of transpiration that occurs during month in HRU (mm H2O)*  
*hrumono(16,:) actual amount of evaporation (from soil) that occurs during month in HRU (mm H2O)*  
*hrumono(17,:) amount of nitrogen applied in continuous fertilizer operation during month in HRU (kg N/ha)*  
*hrumono(18,:) amount of phosphorus applied in continuous fertilizer operation during month in HRU (kg P/ha)*  
*hrumono(22,:) amount of irrigation water applied to HRU during month (mm H2O)*  
*hrumono(23,:) amount of water removed from shallow aquifer in HRU for irrigation during month (mm H2O)*  
*hrumono(24,:) amount of water removed from deep aquifer in HRU for irrigation during month (mm H2O)*  
*hrumono(25,:) potential evapotranspiration in HRU during month (mm H2O)*  
*hrumono(26,:) monthly amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)*  
*hrumono(27,:) monthly amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)*  
*hrumono(28,:) monthly amount of N (organic & mineral) auto-applied in HRU (kg N/ha)*  
*hrumono(29,:) monthly amount of P (organic & mineral) auto-applied in HRU (kg P/ha)*  
*hrumono(31,:) water stress days in HRU during month (stress days)*  
*hrumono(32,:) temperature stress days in HRU during month (stress days)*  
*hrumono(33,:) nitrogen stress days in HRU during month (stress days)*  
*hrumono(34,:) phosphorus stress days in HRU during month (stress days)*  
*hrumono(35,:) organic nitrogen in surface runoff in HRU during month (kg N/ha)*  
*hrumono(36,:) organic phosphorus in surface runoff in HRU during month (kg P/ha)*  
*hrumono(37,:) nitrate in surface runoff in HRU during month (kg N/ha)*  
*hrumono(38,:) nitrate in lateral flow in HRU during month (kg N/ha)*  
*hrumono(39,:) soluble phosphorus in surface runoff in HRU during month (kg P/ha)*  
*hrumono(40,:) amount of nitrogen removed from soil by plant uptake in HRU during month (kg N/ha)*  
*hrumono(41,:) nitrate percolating past bottom of soil profile in HRU during month (kg N/ha)*  
*hrumono(42,:) amount of phosphorus removed from soil by plant uptake in HRU during month (kg P/ha)*  
*hrumono(43,:) amount of phosphorus moving from labile mineral to active mineral pool in HRU during month (kg P/ha)*  
*hrumono(44,:) amount of phosphorus moving from active mineral to stable mineral pool in HRU during month (kg P/ha)*  
*hrumono(45,:) amount of nitrogen applied to HRU in fertilizer and grazing operations during month (kg N/ha)*  
*hrumono(46,:) amount of phosphorus applied to HRU in fertilizer and grazing operations during month (kg P/ha)*  
*hrumono(47,:) amount of nitrogen added to soil by fixation in HRU during month (kg N/ha)*  
*hrumono(48,:) amount of nitrogen lost by denitrification in HRU during month (kg N/ha)*  
*hrumono(49,:) amount of nitrogen moving from active organic to nitrate pool in HRU during month (kg N/ha)*  
*hrumono(50,:) amount of nitrogen moving from active organic to stable organic pool in HRU during month (kg N/ha)*  
*hrumono(51,:) amount of phosphorus moving from organic to labile mineral pool in HRU during month (kg P/ha)*  
*hrumono(52,:) amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during month (kg N/ha)*  
*hrumono(53,:) amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during month (kg P/ha)*  
*hrumono(54,:) amount of nitrogen added to soil in rain (kg N/ha)*

hrumono(61,:) daily soil loss predicted with USLE equation (metric tons/ha)  
 hrumono(63,:) less persistent bacteria transported to main channel from HRU during month (bacteria/ha)  
 hrumono(64,:) persistent bacteria transported to main channel from HRU during month (bacteria/ha)  
 hrumono(65,:) nitrate loading from groundwater in HRU to main channel during month (kg N/ha)  
 hrumono(66,:) soluble P loading from groundwater in HRU to main channel during month (kg P/ha)  
 hrumono(67,:) loading of mineral P attached to sediment in HRU to main channel during month (kg P/ha)

- real \*8, dimension(:,:), allocatable **rchdy**

rchdy(1,:) flow into reach on day ( $m^3/s$ )  
 rchdy(2,:) flow out of reach on day ( $m^3/s$ )  
 rchdy(3,:) evaporation from reach on day ( $m^3/s$ )  
 rchdy(4,:) transmission losses from reach on day ( $m^3/s$ )  
 rchdy(5,:) sediment transported into reach on day (metric tons)  
 rchdy(6,:) sediment transported out of reach on day (metric tons)  
 rchdy(7,:) sediment concentration in outflow (mg/L)  
 rchdy(8,:) organic N transported into reach on day (kg N)  
 rchdy(9,:) organic N transported out of reach on day (kg N)  
 rchdy(10,:) organic P transported into reach on day (kg P)  
 rchdy(11,:) organic P transported out of reach on day (kg P)  
 rchdy(12,:) nitrate transported into reach on day (kg N)  
 rchdy(13,:) nitrate transported out of reach on day (kg N)  
 rchdy(14,:) ammonia transported into reach on day (kg N)  
 rchdy(15,:) ammonia transported out of reach on day (kg N)  
 rchdy(16,:) nitrite transported into reach on day (kg N)  
 rchdy(17,:) nitrite transported out of reach on day (kg N)  
 rchdy(18,:) soluble P transported into reach on day (kg P)  
 rchdy(19,:) soluble P transported out of reach on day (kg P)  
 rchdy(20,:) chlorophyll-a transported into reach on day (kg chl a)  
 rchdy(21,:) chlorophyll-a transported out of reach on day (kg chl a)  
 rchdy(22,:) CBOD transported into reach on day (kg O<sub>2</sub>)  
 rchdy(23,:) CBOD transported out of reach on day (kg O<sub>2</sub>)  
 rchdy(24,:) dissolved oxygen transported into reach on day (kg O<sub>2</sub>)  
 rchdy(25,:) dissolved oxygen transported out of reach on day (kg O<sub>2</sub>)  
 rchdy(26,:) soluble pesticide transported into reach on day (mg pst)  
 rchdy(27,:) soluble pesticide transported out of reach on day (mg pst)  
 rchdy(28,:) sorbed pesticide transported into reach on day (mg pst)  
 rchdy(29,:) sorbed pesticide transported out of reach on day (mg pst)  
 rchdy(30,:) amount of pesticide lost through reactions in reach on day (mg pst)  
 rchdy(31,:) amount of pesticide lost through volatilization from reach on day (mg pst)  
 rchdy(32,:) amount of pesticide settling out of reach to bed sediment on day (mg pst)  
 rchdy(33,:) amount of pesticide resuspended from bed sediment to reach on day (mg pst)  
 rchdy(34,:) amount of pesticide diffusing from reach to bed sediment on day (mg pst)  
 rchdy(35,:) amount of pesticide in sediment layer lost through reactions on day (mg pst)  
 rchdy(36,:) amount of pesticide in sediment layer lost through burial on day (mg pst)  
 rchdy(37,:) amount of pesticide stored in river bed sediments (mg pst)  
 rchdy(38,:) persistent bacteria transported out of reach on day (kg bact)  
 rchdy(39,:) less persistent bacteria transported out of reach on day (kg bact)  
 rchdy(40,:) amount of conservative metal #1 transported out of reach on day (kg)  
 rchdy(41,:) amount of conservative metal #2 transported out of reach on day (kg)  
 rchdy(42,:) amount of conservative metal #3 transported out of reach on day (kg)  
 rchdy(43,:) total N (org N + no<sub>3</sub> + no<sub>2</sub> + nh<sub>4</sub> outs) (kg)  
 rchdy(44,:) total P (org P + sol p outs) (kg)

- real \*8, dimension(:,:), allocatable **hruyro**

HRU annual output array (varies) hruyro(1,:) precipitation in HRU during year (mm H<sub>2</sub>O)  
 hruyro(2,:) amount of precipitation falling as freezing rain/snow in HRU during year (mm H<sub>2</sub>O)  
 hruyro(3,:) amount of snow melt in HRU during year (mm H<sub>2</sub>O)  
 hruyro(4,:) amount of surface runoff to main channel from HRU during year (ignores impact of transmission losses) (mm H<sub>2</sub>O)  
 hruyro(5,:) amount of lateral flow contribution to main channel from HRU during year (mm H<sub>2</sub>O)  
 hruyro(6,:) amount of groundwater flow contribution to main channel from HRU during year (mm H<sub>2</sub>O)  
 hruyro(7,:) amount of water moving from shallow aquifer to plants or soil profile in HRU during year (mm H<sub>2</sub>O)  
 hruyro(8,:) amount of water recharging deep aquifer in HRU during year (mm H<sub>2</sub>O)  
 hruyro(9,:) total amount of water entering both aquifers from HRU during year (mm H<sub>2</sub>O)  
 hruyro(10,:) water yield (total amount of water entering main channel) from HRU during year (mm H<sub>2</sub>O)



- hruyro(11,:)* amount of water percolating out of the soil profile and into the vadose zone in HRU during year (mm H<sub>2</sub>O)
  - hruyro(12,:)* actual evapotranspiration in HRU during year (mm H<sub>2</sub>O)
  - hruyro(13,:)* amount of transmission losses from tributary channels in HRU for year (mm H<sub>2</sub>O)
  - hruyro(14,:)* sediment yield from HRU for year (metric tons/ha)
  - hruyro(15,:)* actual amount of transpiration that occurs during year in HRU (mm H<sub>2</sub>O)
  - hruyro(16,:)* actual amount of evaporation (from soil) that occurs during year in HRU (mm H<sub>2</sub>O)
  - hruyro(17,:)* amount of nitrogen applied in continuous fertilizer operation during year in HRU (kg N/ha)
  - hruyro(18,:)* amount of phosphorus applied in continuous fertilizer operation during year in HRU (kg P/ha)
  - hruyro(23,:)* amount of water removed from shallow aquifer in HRU for irrigation during year (mm H<sub>2</sub>O)
  - hruyro(24,:)* amount of water removed from deep aquifer in HRU for irrigation during year (mm H<sub>2</sub>O)
  - hruyro(25,:)* potential evapotranspiration in HRU during year (mm H<sub>2</sub>O)
  - hruyro(26,:)* annual amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)
  - hruyro(27,:)* annual amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)
  - hruyro(28,:)* annual amount of N (organic & mineral) auto-applied in HRU (kg N/ha)
  - hruyro(29,:)* annual amount of P (organic & mineral) auto-applied in HRU (kg P/ha)
  - hruyro(31,:)* water stress days in HRU during year (stress days)
  - hruyro(32,:)* temperature stress days in HRU during year (stress days)
  - hruyro(33,:)* nitrogen stress days in HRU during year (stress days)
  - hruyro(34,:)* phosphorus stress days in HRU during year (stress days)
  - hruyro(35,:)* organic nitrogen in surface runoff in HRU during year (kg N/ha)
  - hruyro(36,:)* organic phosphorus in surface runoff in HRU during year (kg P/ha)
  - hruyro(37,:)* nitrate in surface runoff in HRU during year (kg N/ha)
  - hruyro(38,:)* nitrate in lateral flow in HRU during year (kg N/ha)
  - hruyro(39,:)* soluble phosphorus in surface runoff in HRU during year (kg P/ha)
  - hruyro(40,:)* amount of nitrogen removed from soil by plant uptake in HRU during year (kg N/ha)
  - hruyro(41,:)* nitrate percolating past bottom of soil profile in HRU during year (kg N/ha)
  - hruyro(42,:)* amount of phosphorus removed from soil by plant uptake in HRU during year (kg P/ha)
  - hruyro(43,:)* amount of phosphorus moving from labile mineral to active mineral pool in HRU during year (kg P/ha)
  - hruyro(44,:)* amount of phosphorus moving from active mineral to stable mineral pool in HRU during year (kg P/ha)
  - hruyro(45,:)* amount of nitrogen applied to HRU in fertilizer and grazing operations during year (kg N/ha)
  - hruyro(46,:)* amount of phosphorus applied to HRU in fertilizer and grazing operations during year (kg P/ha)
  - hruyro(47,:)* amount of nitrogen added to soil by fixation in HRU during year (kg N/ha)
  - hruyro(48,:)* amount of nitrogen lost by denitrification in HRU during year (kg N/ha)
  - hruyro(49,:)* amount of nitrogen moving from active organic to nitrate pool in HRU during year (kg N/ha)
  - hruyro(50,:)* amount of nitrogen moving from active organic to stable organic pool in HRU during year (kg N/ha)
  - hruyro(51,:)* amount of phosphorus moving from organic to labile mineral pool in HRU during year (kg P/ha)
  - hruyro(52,:)* amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during year (kg N/ha)
  - hruyro(53,:)* amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during year (kg P/ha)
  - hruyro(54,:)* amount of nitrogen added to soil in rain during year (kg N/ha)
  - hruyro(61,:)* daily soil loss predicted with USLE equation (metric tons/ha)
  - hruyro(63,:)* less persistent bacteria transported to main channel from HRU during year (# bacteria/ha)
  - hruyro(64,:)* persistent bacteria transported to main channel from HRU during year (# bacteria/ha)
  - hruyro(65,:)* nitrate loading from groundwater in HRU to main channel during year (kg N/ha)
  - hruyro(66,:)* soluble P loading from groundwater in HRU to main channel during year (kg P/ha)
  - hruyro(67,:)* loading of mineral P attached to sediment in HRU to main channel during year (kg P/ha)
- real \*8, dimension(:,), allocatable **rchaao**
    - reach average annual output array (varies)*
    - rchaao(1,:)* flow into reach during simulation (m<sup>3</sup>/s)
    - rchaao(2,:)* flow out of reach during simulation (m<sup>3</sup>/s)
    - rchaao(3,:)* sediment transported into reach during simulation (metric tons)
    - rchaao(4,:)* sediment transported out of reach during simulation (metric tons)
    - rchaao(5,:)* sediment concentration in outflow during simulation (mg/L)
    - rchaao(6,:)* organic N transported into reach during simulation (kg N)
    - rchaao(7,:)* organic N transported out of reach during simulation (kg N)
    - rchaao(8,:)* organic P transported into reach during simulation (kg P)
    - rchaao(9,:)* organic P transported out of reach during simulation (kg P)
    - rchaao(10,:)* evaporation from reach during simulation (m<sup>3</sup>/s)
    - rchaao(11,:)* transmission losses from reach during simulation (m<sup>3</sup>/s)
    - rchaao(12,:)* conservative metal #1 transported out of reach during simulation (kg)
    - rchaao(13,:)* conservative metal #2 transported out of reach during simulation (kg)

- rchaao(14,:)* conservative metal #3 transported out of reach during simulation (kg)
- rchaao(15,:)* nitrate transported into reach during simulation (kg N)
- rchaao(16,:)* nitrate transported out of reach during simulation (kg N)
- rchaao(17,:)* soluble P transported into reach during simulation (kg P)
- rchaao(18,:)* soluble P transported out of reach during simulation (kg P)
- rchaao(19,:)* soluble pesticide transported into reach during simulation
- rchaao(20,:)* soluble pesticide transported out of reach during simulation
- rchaao(21,:)* sorbed pesticide transported into reach during simulation
- rchaao(22,:)* sorbed pesticide transported out of reach during simulation
- rchaao(23,:)* amount of pesticide lost through reactions in reach during simulation
- rchaao(24,:)* amount of pesticide lost through volatilization from reach during simulation
- rchaao(25,:)* amount of pesticide settling out of reach to bed sediment during simulation
- rchaao(26,:)* amount of pesticide resuspended from bed sediment to reach during simulation
- rchaao(27,:)* amount of pesticide diffusing from reach to bed sediment during simulation
- rchaao(28,:)* amount of pesticide in sediment layer lost through reactions during simulation
- rchaao(29,:)* amount of pesticide in sediment layer lost through burial during simulation
- rchaao(30,:)* chlorophyll-a transported into reach during simulation (kg chl<sub>a</sub>)
- rchaao(31,:)* chlorophyll-a transported out of reach during simulation (kg chl<sub>a</sub>)
- rchaao(32,:)* ammonia transported into reach during simulation (kg N)
- rchaao(33,:)* ammonia transported out of reach during simulation (kg N)
- rchaao(34,:)* nitrite transported into reach during simulation (kg N)
- rchaao(35,:)* nitrite transported out of reach during simulation (kg N)
- rchaao(36,:)* CBOD transported into reach during simulation (kg O<sub>2</sub>)
- rchaao(37,:)* CBOD transported out of reach during simulation (kg O<sub>2</sub>)
- rchaao(38,:)* dissolved oxygen transported into reach during simulation (kg O<sub>2</sub>)
- rchaao(39,:)* dissolved oxygen transported out of reach during simulation (kg O<sub>2</sub>)
- rchaao(40,:)* persistent bacteria transported out of reach during simulation (kg bact)
- rchaao(41,:)* less persistent bacteria transported out of reach during simulation (kg bact)
- rchaao(43,:)* Total N (org N + no<sub>3</sub> + no<sub>2</sub> + nh<sub>4</sub> outs) (kg)
- rchaao(44,:)* Total P (org P + sol p outs) (kg)
- real \*8, dimension(:,:), allocatable [submono](#)
  - subbasin* monthly output array (varies)
  - submono(1,:)* precipitation in subbasin for month (mm H<sub>2</sub>O)
  - submono(2,:)* snow melt in subbasin for month (mm H<sub>2</sub>O)
  - submono(3,:)* surface runoff loading in subbasin for month (mm H<sub>2</sub>O)
  - submono(4,:)* water yield from subbasin for month (mm H<sub>2</sub>O)
  - submono(5,:)* potential evapotranspiration in subbasin for month (mm H<sub>2</sub>O)
  - submono(6,:)* actual evapotranspiration in subbasin for month (mm H<sub>2</sub>O)
  - submono(7,:)* sediment yield from subbasin for month (metric tons/ha)
  - submono(8,:)* organic N loading from subbasin for month (kg N/ha)
  - submono(9,:)* organic P loading from subbasin for month (kg P/ha)
  - submono(10,:)* NO<sub>3</sub> loading from surface runoff in subbasin for month (kg N/ha)
  - submono(11,:)* soluble P loading from subbasin for month (kg P/ha)
  - submono(12,:)* groundwater loading from subbasin for month (mm H<sub>2</sub>O)
  - submono(13,:)* percolation out of soil profile in subbasin for month (mm H<sub>2</sub>O)
  - submono(14,:)* loading to reach of mineral P attached to sediment from subbasin for month (kg P/ha)
- real \*8, dimension(:,:), allocatable [subyro](#)
  - subbasin* annual output array (varies)
  - subyro(1,:)* precipitation in subbasin for year (mm H<sub>2</sub>O)
  - subyro(2,:)* snow melt in subbasin for year (mm H<sub>2</sub>O)
  - subyro(3,:)* surface runoff loading in subbasin for year (mm H<sub>2</sub>O)
  - subyro(4,:)* water yield from subbasin for year (mm H<sub>2</sub>O)
  - subyro(5,:)* potential evapotranspiration in subbasin for year (mm H<sub>2</sub>O)
  - subyro(6,:)* actual evapotranspiration in subbasin for year (mm H<sub>2</sub>O)
  - subyro(7,:)* sediment yield from subbasin for year (metric tons/ha)
  - subyro(8,:)* organic N loading from subbasin for year (kg N/ha)
  - subyro(9,:)* organic P loading from subbasin for year (kg P/ha)
  - subyro(10,:)* NO<sub>3</sub> loading from surface runoff in subbasin for year (kg N/ha)
  - subyro(11,:)* soluble P loading from subbasin for year (kg P/ha)
  - subyro(12,:)* groundwater loading from subbasin for year (mm H<sub>2</sub>O)
  - subyro(13,:)* percolation out of soil profile in subbasin for year (mm H<sub>2</sub>O)
  - subyro(14,:)* loading to reach of mineral P attached to sediment from subbasin for year (kg P/ha)
- real \*8, dimension(:,:), allocatable [hruaao](#)

HRU average annual output array (varies)

hruaao(1,:) precipitation in HRU during simulation (mm H2O)

hruaao(2,:) amount of precipitation falling as freezing rain/snow in HRU during simulation (mm H2O)

hruaao(3,:) amount of snow melt in HRU during simulation (mm H2O)

hruaao(4,:) amount of surface runoff to main channel from HRU during simulation (ignores impact of transmission losses) (mm H2O)

hruaao(5,:) amount of lateral flow contribution to main channel from HRU during simulation (mm H2O)

hruaao(6,:) amount of groundwater flow contribution to main channel from HRU during simulation (mm H2O)

hruaao(7,:) amount of water moving from shallow aquifer to plants or soil profile in HRU during simulation (mm H2O)

hruaao(8,:) amount of water recharging deep aquifer in HRU during simulation (mm H2O)

hruaao(9,:) total amount of water entering both aquifers from HRU during simulation (mm H2O)

hruaao(10,:) water yield (total amount of water entering main channel) from HRU during simulation (mm H2O)

hruaao(11,:) amount of water percolating out of the soil profile and into the vadose zone in HRU during simulation (mm H2O)

hruaao(12,:) actual evapotranspiration in HRU during simulation

hruaao(13,:) amount of transmission losses from tributary channels in HRU for simulation (mm H2O)

hruaao(14,:) sediment yield from HRU for simulation (metric tons/ha)

hruaao(17,:) amount of nitrogen applied in continuous fertilizer operation in HRU for simulation (kg N/ha)

hruaao(18,:) amount of phosphorus applied in continuous fertilizer operation in HRU for simulation (kg P/ha)

hruaao(23,:) amount of water removed from shallow aquifer in HRU for irrigation during simulation (mm H2O)

hruaao(24,:) amount of water removed from deep aquifer in HRU for irrigation during simulation (mm H2O)

hruaao(25,:) potential evapotranspiration in HRU during simulation (mm H2O)

hruaao(26,:) annual amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)

hruaao(27,:) annual amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)

hruaao(28,:) average annual amount of N (organic & mineral) auto-applied in HRU (kg N/ha)

hruaao(29,:) average annual amount of P (organic & mineral) auto-applied in HRU (kg P/ha)

hruaao(31,:) water stress days in HRU during simulation (stress days)

hruaao(32,:) temperature stress days in HRU during simulation (stress days)

hruaao(33,:) nitrogen stress days in HRU during simulation (stress days)

hruaao(34,:) phosphorus stress days in HRU during simulation (stress days)

hruaao(35,:) organic nitrogen in surface runoff in HRU during simulation (kg N/ha)

hruaao(36,:) organic phosphorus in surface runoff in HRU during simulation (kg P/ha)

hruaao(37,:) nitrate in surface runoff in HRU during simulation (kg N/ha)

hruaao(38,:) nitrate in lateral flow in HRU during simulation (kg N/ha)

hruaao(39,:) soluble phosphorus in surface runoff in HRU during simulation (kg P/ha)

hruaao(40,:) amount of nitrogen removed from soil by plant uptake in HRU during simulation (kg N/ha)

hruaao(41,:) nitrate percolating past bottom of soil profile in HRU during simulation (kg N/ha)

hruaao(42,:) amount of phosphorus removed from soil by plant uptake in HRU during simulation (kg P/ha)

hruaao(43,:) amount of phosphorus moving from labile mineral to active mineral pool in HRU during simulation (kg P/ha)

hruaao(44,:) amount of phosphorus moving from active mineral to stable mineral pool in HRU during simulation (kg P/ha)

hruaao(45,:) amount of nitrogen applied to HRU in fertilizer and grazing operations during simulation (kg N/ha)

hruaao(46,:) amount of phosphorus applied to HRU in fertilizer and grazing operations during simulation (kg P/ha)

hruaao(47,:) amount of nitrogen added to soil by fixation in HRU during simulation (kg N/ha)

hruaao(48,:) amount of nitrogen lost by denitrification in HRU during simulation (kg N/ha)

hruaao(49,:) amount of nitrogen moving from active organic to nitrate pool in HRU during simulation (kg N/ha)

hruaao(50,:) amount of nitrogen moving from active organic to stable organic pool in HRU during simulation (kg N/ha)

hruaao(51,:) amount of phosphorus moving from organic to labile mineral pool in HRU during simulation (kg P/ha)

hruaao(52,:) amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during simulation (kg N/ha)

hruaao(53,:) amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during simulation (kg P/ha)

hruaao(54,:) amount of nitrogen added to soil in rain during simulation (kg N/ha)

hruaao(61,:) daily soil loss predicted with USLE equation (metric tons/ha)

hruaao(63,:) less persistent bacteria transported to main channel from HRU during simulation (# bacteria/ha)

hruaao(64,:) persistent bacteria transported to main channel from HRU during simulation (# bacteria/ha)

hruaao(65,:) nitrate loading from groundwater in HRU to main channel during simulation (kg N/ha)

hruaao(66,:) soluble P loading from groundwater in HRU to main channel during simulation (kg P/ha)

hruaao(67,:) loading of mineral P attached to sediment in HRU to main channel during simulation (kg P/ha)

- real \*8, dimension(:,:), allocatable [subaao](#)

subbasin average annual output array (varies)

- real \*8, dimension(:,:), allocatable [resoutm](#)

reservoir monthly output array (varies)

resoutm(1,:) flow into reservoir during month ( $m^3/s$ )

resoutm(2,:) flow out of reservoir during month ( $m^3/s$ )

resoutm(3,:) sediment entering reservoir during month (metric tons)

resoutm(4,:) sediment leaving reservoir during month (metric tons)

resoutm(5,:) sediment concentration in reservoir during month (mg/L)

resoutm(6,:) pesticide entering reservoir during month (mg pst)

resoutm(7,:) pesticide lost from reservoir through reactions during month (mg pst)

resoutm(8,:) pesticide lost from reservoir through volatilization during month (mg pst)

resoutm(9,:) pesticide moving from water to sediment through settling during month (mg pst)

resoutm(10,:) pesticide moving from sediment to water through resuspension during month (mg pst)

resoutm(11,:) pesticide moving from water to sediment through diffusion during month (mg pst)

resoutm(12,:) pesticide lost from reservoir sediment layer through reactions during month (mg pst)

resoutm(13,:) pesticide lost from reservoir sediment layer through burial during month (mg pst)

resoutm(14,:) pesticide transported out of reservoir during month (mg pst)

resoutm(15,:) pesticide concentration in reservoir water during month (mg pst/ $m^3$ )

resoutm(16,:) pesticide concentration in reservoir sediment layer during month (mg pst/ $m^3$ )

resoutm(17,:) evaporation from reservoir during month ( $m^3 H_2O$ )

resoutm(18,:) seepage from reservoir during month ( $m^3 H_2O$ )

resoutm(19,:) precipitation on reservoir during month ( $m^3 H_2O$ )

resoutm(22,:) organic N entering reservoir during month (kg N)

resoutm(23,:) organic N leaving reservoir during month (kg N)

resoutm(24,:) organic P entering reservoir during month (kg P)

resoutm(25,:) organic P leaving reservoir during month (kg P)

resoutm(26,:) nitrate entering reservoir during month (kg N)

resoutm(27,:) nitrate leaving reservoir during month (kg N)

resoutm(28,:) nitrite entering reservoir during month (kg N)

resoutm(29,:) nitrite leaving reservoir during month (kg N)

resoutm(30,:) ammonia entering reservoir during month (kg N)

resoutm(31,:) ammonia leaving reservoir during month (kg N)

resoutm(32,:) mineral P entering reservoir during month (kg P)

resoutm(33,:) mineral P leaving reservoir during month (kg P)

resoutm(34,:) chlorophyll-a entering reservoir during month (kg chla)

resoutm(35,:) chlorophyll-a leaving reservoir during month (kg chla)

resoutm(36,:) organic P concentration in reservoir water during month (mg P/L)

resoutm(37,:) mineral P concentration in reservoir water during month (mg P/L)

resoutm(38,:) organic N concentration in reservoir water during month (mg N/L)

resoutm(39,:) nitrate concentration in reservoir water during month (mg N/L)

resoutm(40,:) nitrite concentration in reservoir water during month (mg N/L)

resoutm(41,:) ammonia concentration in reservoir water during month (mg N/L)

- real \*8, dimension(:,:), allocatable **resouty**

reservoir annual output array (varies)

resouty(1,:) flow into reservoir during year ( $m^3/s$ )

resouty(2,:) flow out of reservoir during year ( $m^3/s$ )

resouty(3,:) sediment entering reservoir during year (metric tons)

resouty(4,:) sediment leaving reservoir during year (metric tons)

resouty(5,:) sediment concentration in reservoir during year (mg/L)

resouty(6,:) pesticide entering reservoir during year (mg pst)

resouty(7,:) pesticide lost from reservoir through reactions during year (mg pst)

resouty(8,:) pesticide lost from reservoir through volatilization during year (mg pst)

resouty(9,:) pesticide moving from water to sediment through settling during year (mg pst)

resouty(10,:) pesticide moving from sediment to water through resuspension during year (mg pst)

resouty(11,:) pesticide moving from water to sediment through diffusion during year (mg pst)

resouty(12,:) pesticide lost from reservoir sediment layer through reactions during year (mg pst)

resouty(13,:) pesticide lost from reservoir sediment layer through burial during year (mg pst)

resouty(14,:) pesticide transported out of reservoir during year (mg pst)

resouty(15,:) pesticide concentration in reservoir water during year (mg pst/ $m^3$ )

resouty(16,:) pesticide concentration in reservoir sediment layer during year (mg pst/ $m^3$ )

resouty(17,:) evaporation from reservoir during year ( $m^3 H_2O$ )

resouty(18,:) seepage from reservoir during year ( $m^3 H_2O$ )

resouty(19,:) precipitation on reservoir during year ( $m^3 H_2O$ )

resouty(22,:) organic N entering reservoir during year (kg N)

resouty(23,:) organic N leaving reservoir during year (kg N)



- resouty(24,:) organic P entering reservoir during year (kg P)*
- resouty(25,:) organic P leaving reservoir during year (kg P)*
- resouty(26,:) nitrate entering reservoir during year (kg N)*
- resouty(27,:) nitrate leaving reservoir during year (kg N)*
- resouty(28,:) nitrite entering reservoir during year (kg N)*
- resouty(29,:) nitrite leaving reservoir during year (kg N)*
- resouty(30,:) ammonia entering reservoir during year (kg N)*
- resouty(31,:) ammonia leaving reservoir during year (kg N)*
- resouty(32,:) mineral P entering reservoir during year (kg P)*
- resouty(33,:) mineral P leaving reservoir during year (kg P)*
- resouty(34,:) chlorophyll-a entering reservoir during year (kg chla)*
- resouty(35,:) chlorophyll-a leaving reservoir during year (kg chla)*
- resouty(36,:) organic P concentration in reservoir water during year (mg P/L)*
- resouty(37,:) mineral P concentration in reservoir water during year (mg P/L)*
- resouty(38,:) organic N concentration in reservoir water during year (mg N/L)*
- resouty(39,:) nitrate concentration in reservoir water during year (mg N/L)*
- resouty(40,:) nitrite concentration in reservoir water during year (mg N/L)*
- resouty(41,:) ammonia concentration in reservoir water during year (mg N/L)*
- real \*8, dimension(:,:), allocatable **resouta**
  - reservoir average annual output array (varies)*
  - resouta(3,:) sediment entering reservoir during simulation (metric tons)*
  - resouta(4,:) sediment leaving reservoir during simulation (metric tons)*
  - resouta(17,:) evaporation from reservoir during simulation ( $m^3 H_2O$ )*
  - resouta(18,:) seepage from reservoir during simulation ( $m^3 H_2O$ )*
  - resouta(19,:) precipitation on reservoir during simulation ( $m^3 H_2O$ )*
  - resouta(20,:) water entering reservoir during simulation ( $m^3 H_2O$ )*
  - resouta(21,:) water leaving reservoir during simulation ( $m^3 H_2O$ )*
- real \*8, dimension(12, 8) **wshd\_aamon**
  - wshd\_aamon(:,1) average annual precipitation in watershed falling during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,2) average annual freezing rain in watershed falling during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,3) average annual surface runoff in watershed during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,4) average annual lateral flow in watershed during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,5) average annual water yield in watershed during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,6) average annual actual evapotranspiration in watershed during month (mm H<sub>2</sub>O)*
  - wshd\_aamon(:,7) average annual sediment yield in watershed during month (metric tons)*
  - wshd\_aamon(:,8) average annual potential evapotranspiration in watershed during month (mm H<sub>2</sub>O)*
- real \*8, dimension(:,:), allocatable **wtrmon**
  - HRU monthly output data array for impoundments (varies)*
  - wtrmon(1,:) evaporation from ponds in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(2,:) seepage from ponds in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(3,:) precipitation on ponds in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(4,:) amount of water entering ponds in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(5,:) sediment entering ponds in HRU for month (metric tons/ha)*
  - wtrmon(6,:) amount of water leaving ponds in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(7,:) sediment leaving ponds in HRU for month (metric tons/ha)*
  - wtrmon(8,:) precipitation on wetlands in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(9,:) volume of water entering wetlands from HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(10,:) sediment loading to wetlands for month from HRU (metric tons/ha)*
  - wtrmon(11,:) evaporation from wetlands in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(12,:) seepage from wetlands in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(13,:) volume of water leaving wetlands in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(14,:) sediment loading from wetlands in HRU to main channel during month (metric tons/ha)*
  - wtrmon(15,:) precipitation on potholes in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(16,:) evaporation from potholes in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(17,:) seepage from potholes in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(18,:) water leaving potholes in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(19,:) water entering potholes in HRU for month (mm H<sub>2</sub>O)*
  - wtrmon(20,:) sediment entering potholes in HRU for month (metric tons/ha)*
  - wtrmon(21,:) sediment leaving potholes in HRU for month (metric tons/ha)*
- real \*8, dimension(:,:), allocatable **wtryr**
  - HRU impoundment annual output array (varies)*
  - wtryr(1,:) evaporation from ponds in HRU for year (mm H<sub>2</sub>O)*

- wtryr(2,:) seepage from ponds in HRU for year (mm H2O)*
- wtryr(3,:) precipitation on ponds in HRU for year (mm H2O)*
- wtryr(4,:) amount of water entering ponds in HRU for year (mm H2O)*
- wtryr(5,:) sediment entering ponds in HRU for year (metric tons/ha)*
- wtryr(6,:) amount of water leaving ponds in HRU for year (mm H2O)*
- wtryr(7,:) sediment leaving ponds in HRU for year (metric tons/ha)*
- wtryr(8,:) precipitation on wetlands in HRU for year (mm H2O)*
- wtryr(9,:) volume of water entering wetlands from HRU for year (mm H2O)*
- wtryr(10,:) sediment loading to wetlands for year from HRU (metric tons/ha)*
- wtryr(11,:) evaporation from wetlands in HRU for year (mm H2O)*
- wtryr(12,:) seepage from wetlands in HRU for year (mm H2O)*
- wtryr(13,:) volume of water leaving wetlands in HRU for year (mm H2O)*
- wtryr(14,:) sediment loading from wetlands in HRU to main channel during year (metric tons/ha)*
- wtryr(15,:) precipitation on potholes in HRU during year (mm H2O)*
- wtryr(16,:) evaporation from potholes in HRU during year (mm H2O)*
- wtryr(17,:) seepage from potholes in HRU during year (mm H2O)*
- wtryr(18,:) water leaving potholes in HRU during year (mm H2O)*
- wtryr(19,:) water entering potholes in HRU during year (mm H2O)*
- wtryr(20,:) sediment entering potholes in HRU during year (metric tons/ha)*
- wtryr(21,:) sediment leaving potholes in HRU during year (metric tons/ha)*
- real \*8, dimension(:,:), allocatable [wtraa](#)  
*HRU impoundment average annual output array (varies)*
- real \*8, dimension(:,:), allocatable [sub\\_smfmx](#)  
*max melt rate for snow during year (June 21) for subbasin(:) where deg C refers to the air temperature. SUB\_SMFMX and SMFMN allow the rate of snow melt to vary through the year. These parameters are accounting for the impact of soil temperature on snow melt (range: -5.0/5.0) (mm/deg C/day)*
- real \*8, dimension(:,:), allocatable [sub\\_smfmn](#)  
*min melt rate for snow during year (Dec 21) for subbasin(:) (range: -5.0/5.0) where deg C refers to the air temperature (mm/deg C/day)*
- real \*8, dimension(:,:), allocatable [hrupstd](#)  
*hrupstd(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU on day (in solution) (mg pst)*  
*hrupstd(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU on day (sorbed to sediment) (mg pst)*
- real \*8, dimension(:,:), allocatable [hrupstm](#)  
*hrupstm(:,,:) HRU monthly pesticide output array (varies)*  
*hrupstm(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU during month (in solution) (mg pst)*  
*hrupstm(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU during month (sorbed to sediment) (mg pst)*  
*hrupstm(:,3,:) total pesticide loading to stream in surface runoff from HRU during month (mg pst)*
- real \*8, dimension(:,:), allocatable [hrupsta](#)  
*HRU average annual pesticide output array (varies)*
- real \*8, dimension(:,:), allocatable [hrupsty](#)  
*hrupsty(:,,:) HRU annual pesticide output array (varies)*  
*hrupsty(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU during year (in solution) (mg pst)*  
*hrupsty(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU during year (sorbed to sediment) (mg pst)*
- integer, dimension(:), allocatable [ifirstt](#)  
*temperature data search code (none)*  
*0 first day of temperature data located in file*  
*1 first day of temperature data not located in file*
- integer, dimension(:), allocatable [ifirstpcp](#)
- integer, dimension(:), allocatable [elevp](#)  
*elevation of precipitation gage station (m)*
- integer, dimension(:), allocatable [elevt](#)  
*elevation of temperature gage station (m)*
- real \*8, dimension(:,:), allocatable [ftmpmn](#)

- avg monthly minimum air temperature (deg C)*
  - real \*8, dimension(:,:), allocatable [ftmpmx](#)
  - avg monthly maximum air temperature (deg C)*
    - real \*8, dimension(:,:), allocatable [ftmpstdmn](#)
    - standard deviation for avg monthly minimum air temperature (deg C)*
      - real \*8, dimension(:,:), allocatable [ftmpstdmx](#)
      - standard deviation for avg monthly maximum air temperature (deg C)*
        - real \*8, dimension(:,:), allocatable [fpcp\\_stat](#)
        - fpcp\_stat(:,1,:): average amount of precipitation falling in one day for the month (mm/day)*
          - fpcp\_stat(:,2,:): standard deviation for the average daily precipitation (mm/day)*
          - fpcp\_stat(:,3,:): skew coefficient for the average daily precipitation (none)*
  - real \*8, dimension(:,:), allocatable [fpr\\_w1](#)
    - probability of wet day after dry day in month (none)*
  - real \*8, dimension(:,:), allocatable [fpr\\_w2](#)
    - probability of wet day after wet day in month (none)*
  - real \*8, dimension(:,:), allocatable [fpr\\_w3](#)
    - proportion of wet days in the month (none)*
  - real \*8, dimension(:), allocatable [ch\\_d](#)
    - average depth of main channel (m)*
  - real \*8, dimension(:), allocatable [flwin](#)
  - real \*8, dimension(:), allocatable [flwout](#)
  - real \*8, dimension(:), allocatable [bankst](#)
  - real \*8, dimension(:), allocatable [ch\\_wi](#)
  - real \*8, dimension(:), allocatable [ch\\_onco](#)
    - channel organic n concentration (ppm)*
  - real \*8, dimension(:), allocatable [ch\\_opco](#)
    - channel organic p concentration (ppm)*
  - real \*8, dimension(:), allocatable [ch\\_organ](#)
  - real \*8, dimension(:), allocatable [ch\\_organp](#)
  - real \*8, dimension(:), allocatable [drift](#)
    - amount of pesticide drifting onto main channel in subbasin (kg)*
  - real \*8, dimension(:), allocatable [rch\\_dox](#)
  - real \*8, dimension(:), allocatable [rch\\_bactp](#)
  - real \*8, dimension(:), allocatable [alpha\\_bnk](#)
    - alpha factor for bank storage recession curve (days)*
  - real \*8, dimension(:), allocatable [alpha\\_bnke](#)
    - $\exp(-\alpha_{bnk})$  (none)*
  - real \*8, dimension(:), allocatable [disolvp](#)
  - real \*8, dimension(:), allocatable [algae](#)
  - real \*8, dimension(:), allocatable [sedst](#)
  - real \*8, dimension(:), allocatable [rchstor](#)
  - real \*8, dimension(:), allocatable [organicon](#)
  - real \*8, dimension(:), allocatable [organiconp](#)
  - real \*8, dimension(:), allocatable [chlora](#)
  - real \*8, dimension(:), allocatable [ch\\_li](#)
    - initial length of main channel (km)*
  - real \*8, dimension(:), allocatable [ch\\_si](#)
    - initial slope of main channel (m/m)*
  - real \*8, dimension(:), allocatable [nitraten](#)
  - real \*8, dimension(:), allocatable [nitriten](#)
  - real \*8, dimension(:), allocatable [ch\\_bnk\\_san](#)
  - real \*8, dimension(:), allocatable [ch\\_bnk\\_sil](#)

- real \*8, dimension(:), allocatable **ch\_bnk\_cla**
- real \*8, dimension(:), allocatable **ch\_bnk\_gra**
- real \*8, dimension(:), allocatable **ch\_bed\_san**
- real \*8, dimension(:), allocatable **ch\_bed\_sil**
- real \*8, dimension(:), allocatable **ch\_bed\_cla**
- real \*8, dimension(:), allocatable **ch\_bed\_gra**
- real \*8, dimension(:), allocatable **depdfp**
- real \*8, dimension(:), allocatable **depsanfp**
- real \*8, dimension(:), allocatable **depsilfp**
- real \*8, dimension(:), allocatable **depclafp**
- real \*8, dimension(:), allocatable **depsagfp**
- real \*8, dimension(:), allocatable **deplagfp**
- real \*8, dimension(:), allocatable **depch**
- real \*8, dimension(:), allocatable **depsanch**
- real \*8, dimension(:), allocatable **depsilch**
- real \*8, dimension(:), allocatable **depclach**
- real \*8, dimension(:), allocatable **depsagch**
- real \*8, dimension(:), allocatable **deplagch**
- real \*8, dimension(:), allocatable **depgrach**
- real \*8, dimension(:), allocatable **depgrafp**
- real \*8, dimension(:), allocatable **grast**
- real \*8, dimension(:), allocatable **r2adj**  
*curve number retention parameter adjustment factor to adjust surface runoff for flat slopes (0.5 - 3.0) (dimensionless)*
- real \*8, dimension(:), allocatable **prf**  
*Reach peak rate adjustment factor for sediment routing in the channel. Allows impact of peak flow rate on sediment routing and channel reshaping to be taken into account (none)*
- real \*8, dimension(:), allocatable **depprch**
- real \*8, dimension(:), allocatable **depprfp**
- real \*8, dimension(:), allocatable **spcon**  
*linear parameter for calculating sediment reentrained in channel sediment routing*
- real \*8, dimension(:), allocatable **spexp**  
*exponent parameter for calculating sediment reentrained in channel sediment routing*
- real \*8, dimension(:), allocatable **sanst**
- real \*8, dimension(:), allocatable **silst**
- real \*8, dimension(:), allocatable **clast**
- real \*8, dimension(:), allocatable **sagst**
- real \*8, dimension(:), allocatable **lagst**
- real \*8, dimension(:), allocatable **pot\_san**
- real \*8, dimension(:), allocatable **pot\_sil**
- real \*8, dimension(:), allocatable **pot\_cla**
- real \*8, dimension(:), allocatable **pot\_sag**
- real \*8, dimension(:), allocatable **pot\_lag**
- real \*8, dimension(:), allocatable **potsani**
- real \*8, dimension(:), allocatable **potsili**
- real \*8, dimension(:), allocatable **potclai**
- real \*8, dimension(:), allocatable **potsagi**
- real \*8, dimension(:), allocatable **potlagi**
- real \*8, dimension(:), allocatable **sanyld**
- real \*8, dimension(:), allocatable **silyld**
- real \*8, dimension(:), allocatable **clayld**
- real \*8, dimension(:), allocatable **sagyld**
- real \*8, dimension(:), allocatable **lagyld**
- real \*8, dimension(:), allocatable **grayld**
- real \*8, dimension(:), allocatable **res\_san**



- real \*8, dimension(:), allocatable **res\_sil**
- real \*8, dimension(:), allocatable **res\_cla**
- real \*8, dimension(:), allocatable **res\_sag**
- real \*8, dimension(:), allocatable **res\_lag**
- real \*8, dimension(:), allocatable **res\_gra**
- real \*8, dimension(:), allocatable **pnd\_san**
- real \*8, dimension(:), allocatable **pnd\_sil**
- real \*8, dimension(:), allocatable **pnd\_cla**
- real \*8, dimension(:), allocatable **pnd\_sag**
- real \*8, dimension(:), allocatable **pnd\_lag**
- real \*8, dimension(:), allocatable **wet\_san**
- real \*8, dimension(:), allocatable **wet\_sil**
- real \*8, dimension(:), allocatable **wet\_cla**
- real \*8, dimension(:), allocatable **wet\_lag**
- real \*8, dimension(:), allocatable **wet\_sag**
- real \*8 **ressano**
- real \*8 **ressilo**
- real \*8 **resclao**
- real \*8 **ressago**
- real \*8 **reslago**
- real \*8 **resgrao**
- real \*8 **ressani**
- real \*8 **ressili**
- real \*8 **resclai**
- real \*8 **ressagi**
- real \*8 **reslagi**
- real \*8 **resgrai**
- real \*8 **potsano**
- real \*8 **potsilo**
- real \*8 **potclao**
- real \*8 **potsago**
- real \*8 **potlago**
- real \*8 **pndsanin**
- real \*8 **pndsilin**
- real \*8 **pndclain**
- real \*8 **pndsagin**
- real \*8 **pndlagin**
- real \*8 **pndsano**
- real \*8 **pndsilo**
- real \*8 **pndclao**
- real \*8 **pndsago**
- real \*8 **pndlago**
- real \*8, dimension(:), allocatable **ch\_di**  
*initial depth of main channel (m)*
- real \*8, dimension(:), allocatable **ch\_erod**  
*channel erodibility factor (0.0-1.0) (none)*  
*0 non-erosive channel*  
*1 no resistance to erosion*
- real \*8, dimension(:), allocatable **ch\_l2**  
*length of main channel (km)*
- real \*8, dimension(:), allocatable **ch\_cov**
- real \*8, dimension(:), allocatable **ch\_bnk\_bd**  
*bulk density of channel bank sediment (1.1-1.9) (g/cc)*
- real \*8, dimension(:), allocatable **ch\_bed\_bd**

- bulk density of channel bed sediment (1.1-1.9) (g/cc)*

  - real \*8, dimension(:), allocatable [ch\\_bnk\\_kd](#)
- erodibility of channel bank sediment by jet test (Peter Allen needs to give more info on this)*

  - real \*8, dimension(:), allocatable [ch\\_bed\\_kd](#)
- erodibility of channel bed sediment by jet test (Peter Allen needs to give more info on this)*

  - real \*8, dimension(:), allocatable [ch\\_bnk\\_d50](#)
- D50(median) particle size diameter of channel bank sediment (0.001 - 20)*

  - real \*8, dimension(:), allocatable [ch\\_bed\\_d50](#)
- D50(median) particle size diameter of channel bed sediment (micrometers) (0.001 - 20)*

  - real \*8, dimension(:), allocatable [ch\\_cov1](#)
- channel erodibility factor (0.0-1.0) (none)*  
*0 non-erosive channel*  
*1 no resistance to erosion*

  - real \*8, dimension(:), allocatable [ch\\_cov2](#)
- channel cover factor (0.0-1.0) (none)*  
*0 channel is completely protected from erosion by cover*  
*1 no vegetative cover on channel*

  - real \*8, dimension(:), allocatable [ch\\_cov2](#)
- critical shear stress of channel bed (N/m2)*

  - real \*8, dimension(:), allocatable [tc\\_bed](#)
- critical shear stress of channel bank (N/m2)*

  - real \*8, dimension(:), allocatable [tc\\_bnk](#)
- sediment routine methods (DAILY):*  
*0 = original SWAT method*  
*1 = Bagnold's*  
*2 = Kodatie*  
*3 = Molinas WU*  
*4 = Yang*

  - integer, dimension(:), allocatable [ch\\_eqn](#)
- pesticide reaction coefficient in reach (1/day)*

  - real \*8, dimension(:), allocatable [chpst\\_rea](#)
- pesticide volatilization coefficient in reach (m/day)*

  - real \*8, dimension(:), allocatable [chpst\\_vol](#)
- pesticide partition coefficient between water and sediment in reach ( $m^3/g$ )*

  - real \*8, dimension(:), allocatable [chpst\\_koc](#)
- resuspension velocity in reach for pesticide sorbed to sediment (m/day)*

  - real \*8, dimension(:), allocatable [chpst\\_rsp](#)
- settling velocity in reach for pesticide sorbed to sediment (m/day)*

  - real \*8, dimension(:), allocatable [chpst\\_stl](#)
- channel width to depth ratio (m/m)*

  - real \*8, dimension(:), allocatable [ch\\_wdr](#)
- mixing velocity (diffusion/dispersion) for pesticide in reach (m/day)*

  - real \*8, dimension(:), allocatable [chpst\\_mix](#)
- inital pesticide concentration in river bed sediment ( $mg/m^3$ )*

  - real \*8, dimension(:), allocatable [sedpst\\_conc](#)
- pesticide burial velocity in river bed sediment (m/day)*

  - real \*8, dimension(:), allocatable [sedpst\\_bry](#)
- pesticide reaction coefficient in river bed sediment (1/day)*

  - real \*8, dimension(:), allocatable [sedpst\\_rea](#)
- depth of active sediment layer in reach for pesticide (m)*

  - real \*8, dimension(:), allocatable [sedpst\\_act](#)
- channel width to depth ratio (m/m)*

  - real \*8, dimension(:), allocatable [rch\\_cbod](#)
- channel width to depth ratio (m/m)*

  - real \*8, dimension(:), allocatable [rch\\_bactlp](#)

- real \*8, dimension(:), allocatable [chside](#)  
*change in horizontal distance per unit vertical distance (0.0 - 5)*  
*0 = for vertical channel bank*  
*5 = for channel bank with gentl side slope*
- real \*8, dimension(:), allocatable [rs1](#)  
*local algal settling rate in reach at 20 deg C (m/day or m/hour)*
- real \*8, dimension(:), allocatable [rs2](#)  
*benthos source rate for dissolved phosphorus in reach at 20 deg C ((mg disP-P)/(m<sup>2</sup>\*day) or (mg disP-P)/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable [rs3](#)  
*benthos source rate for ammonia nitrogen in reach at 20 deg C ((mg NH4-N)/(m<sup>2</sup>\*day) or (mg NH4-N)/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable [rs4](#)  
*rate coefficient for organic nitrogen settling in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rs5](#)  
*organic phosphorus settling rate in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rk1](#)  
*CBOD deoxygenation rate coefficient in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rk2](#)  
*reaeration rate in accordance with Fickian diffusion in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rk3](#)  
*rate of loss of CBOD due to settling in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rk4](#)  
*sediment oxygen demand rate in reach at 20 deg C (mg O2/(m<sup>2</sup>\*day) or mg O2/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable [rk5](#)  
*coliform die-off rate in reach (1/day)*
- real \*8, dimension(:), allocatable [rs6](#)  
*rate coefficient for settling of arbitrary non-conservative constituent in reach (1/day)*
- real \*8, dimension(:), allocatable [rs7](#)  
*benthos source rate for arbitrary non-conservative constituent in reach ((mg ANC)/(m<sup>2</sup>\*day))*
- real \*8, dimension(:), allocatable [bc1](#)  
*rate constant for biological oxidation of NH3 to NO2 in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [bc2](#)  
*rate constant for biological oxidation of NO2 to NO3 in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [bc3](#)  
*rate constant for hydrolysis of organic N to ammonia in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [bc4](#)  
*rate constant for the decay of organic P to dissolved P in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable [rk6](#)  
*decay rate for arbitrary non-conservative constituent in reach (1/day)*
- real \*8, dimension(:), allocatable **ammonian**
- real \*8, dimension(:), allocatable **orig\_sedpstconc**
- real \*8, dimension(:,:), allocatable [wurch](#)  
*average daily water removal from the reach for the month (10<sup>4</sup> m<sup>3</sup>/day)*
- integer, dimension(:), allocatable **icanal**
- integer, dimension(:), allocatable **itb**
- real \*8, dimension(:), allocatable [ch\\_revap](#)  
*revap coeff: this variable controls the amount of water moving from bank storage to the root zone as a result of soil moisture depletion(none)*
- real \*8, dimension(:), allocatable **dep\_chan**
- real \*8, dimension(:), allocatable [harg\\_petco](#)  
*coefficient related to radiation used in hargreaves eq (range: 0.0019 - 0.0032)*
- real \*8, dimension(:), allocatable **subfr\_nowtr**

- real \*8, dimension(:), allocatable [cncoef\\_sub](#)  
*soil water depletion coefficient used in the new (modified curve number method) same as soil index coeff used in APEX range: 0.5 - 2.0*
- real \*8, dimension(:), allocatable [dr\\_sub](#)
- real \*8, dimension(:), allocatable [sub\\_fr](#)  
*fraction of total watershed area contained in subbasin (km2/km2)*
- real \*8, dimension(:), allocatable [sub\\_sw](#)  
*water in soil profile in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable [wcklsp](#)
- real \*8, dimension(:), allocatable [sub\\_minp](#)
- real \*8, dimension(:), allocatable [sub\\_sumfc](#)
- real \*8, dimension(:), allocatable [sub\\_gwno3](#)
- real \*8, dimension(:), allocatable [sub\\_gwsolp](#)
- real \*8, dimension(:), allocatable [co2](#)  
*CO2 concentration (ppmv)*
- real \*8, dimension(:), allocatable [sub\\_km](#)  
*area of subbasin in square kilometers (km<sup>2</sup>)*
- real \*8, dimension(:), allocatable [wlat](#)  
*latitude of weather station used to compile data (degrees)*
- real \*8, dimension(:), allocatable [sub\\_tc](#)  
*time of concentration for subbasin (hour)*
- real \*8, dimension(:), allocatable [sub\\_pet](#)
- real \*8, dimension(:), allocatable [welev](#)  
*elevation of weather station used to compile weather generator data (m)*
- real \*8, dimension(:), allocatable [sub\\_bd](#)  
*bulk density in subbasin first soil layer (Mg/m<sup>3</sup>)*
- real \*8, dimension(:), allocatable [sub\\_orgn](#)  
*amount of nitrogen stored in all organic pools (kg N/ha)*
- real \*8, dimension(:), allocatable [sub\\_orgp](#)  
*amount of phosphorus stored in all organic pools (kg P/ha)*
- real \*8, dimension(:), allocatable [sub\\_wtmp](#)
- real \*8, dimension(:), allocatable [sub\\_sedpa](#)
- real \*8, dimension(:), allocatable [sub\\_sedps](#)
- real \*8, dimension(:), allocatable [daylmn](#)  
*shortest daylength occurring during the year (hour)*
- real \*8, dimension(:), allocatable [sub\\_minpa](#)  
*amount of phosphorus stored in active mineral pools sorbed to sediment (kg P/ha)*
- real \*8, dimension(:), allocatable [sub\\_minps](#)  
*amount of phosphorus stored in stable mineral pools sorbed to sediment (kg P/ha)*
- real \*8, dimension(:), allocatable [latcos](#)  
*cos(latitude) (none)*
- real \*8, dimension(:), allocatable [latsin](#)  
*sin(latitude) (none)*
- real \*8, dimension(:), allocatable [phutot](#)  
*total potential heat units for year (used when no crop is growing) (heat unit)*
- real \*8, dimension(:), allocatable [plaps](#)  
*precipitation lapse rate: precipitation change due to change in elevation (mm H2O/km)*
- real \*8, dimension(:), allocatable [tlaps](#)  
*temperature lapse rate: temperature change due to change in elevation (deg C/km)*
- real \*8, dimension(:), allocatable [tmp\\_an](#)  
*average annual air temperature (deg C)*
- real \*8, dimension(:), allocatable [sub\\_precip](#)

- amount of water reaching soil surface in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable [rammo\\_sub](#)
- atmospheric deposition of ammonium values for entire watershed (mg/l)*
- real \*8, dimension(:), allocatable [rcn\\_sub](#)
- atmospheric deposition of nitrate for entire watershed (mg/l)*
- real \*8, dimension(:), allocatable [pcpdays](#)
- real \*8, dimension(:), allocatable [atmo\\_day](#)
- real \*8, dimension(:), allocatable [sub\\_snom](#)
- amount of snow melt in subbasin on day (mm H2O)*
- real \*8, dimension(:), allocatable [sub\\_qd](#)
- surface runoff that reaches main channel during day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable [sub\\_sedy](#)
- real \*8, dimension(:), allocatable [sub\\_tran](#)
- transmission losses on day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable [sub\\_no3](#)
- real \*8, dimension(:), allocatable [sub\\_latno3](#)
- real \*8, dimension(:,:), allocatable [sub\\_sftmp](#)
- snowfall temperature for subbasin(:). Mean air temperature at which precip is equally likely to be rain as snow/freezing rain (range: -5.0/5.0) (deg C)*
- real \*8, dimension(:,:), allocatable [sub\\_smtmp](#)
- snow melt base temperature for subbasin(:) mean air temperature at which snow melt will occur (range: -5.0/5.0) (deg C)*
- real \*8, dimension(:,:), allocatable [sub\\_timp](#)
- snow pack temperature lag factor (0-1) (none)*  
*1 = no lag (snow pack temp=current day air temp) as the lag factor goes to zero, the snow pack's temperature will be less influenced by the current day's air temperature*
- real \*8, dimension(:), allocatable [sub\\_tileno3](#)
- real \*8, dimension(:), allocatable [sub\\_solp](#)
- real \*8, dimension(:), allocatable [sub\\_subp](#)
- real \*8, dimension(:), allocatable [sub\\_etday](#)
- real \*8, dimension(:), allocatable [sub\\_elev](#)
- average elevation of HRU (m)*
- real \*8, dimension(:), allocatable [sub\\_surfq](#)
- surface runoff generated on day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable [sub\\_wyld](#)
- real \*8, dimension(:), allocatable [qird](#)
- real \*8, dimension(:), allocatable [sub\\_gwq](#)
- real \*8, dimension(:), allocatable [sub\\_sep](#)
- real \*8, dimension(:), allocatable [sub\\_chl](#)
- real \*8, dimension(:), allocatable [sub\\_cbod](#)
- real \*8, dimension(:), allocatable [sub\\_dox](#)
- real \*8, dimension(:), allocatable [sub\\_solpst](#)
- real \*8, dimension(:), allocatable [sub\\_sorpst](#)
- real \*8, dimension(:), allocatable [sub\\_yorgn](#)
- real \*8, dimension(:), allocatable [sub\\_yorgp](#)
- real \*8, dimension(:), allocatable [sub\\_lat](#)
- latitude of HRU/subbasin (degrees)*
- real \*8, dimension(:), allocatable [sub\\_bactp](#)
- real \*8, dimension(:), allocatable [sub\\_bactlp](#)
- real \*8, dimension(:), allocatable [sub\\_latq](#)
- real \*8, dimension(:), allocatable [sub\\_gwq\\_d](#)
- real \*8, dimension(:), allocatable [sub\\_tileq](#)
- real \*8, dimension(:), allocatable [sub\\_vaptile](#)

- real \*8, dimension(:), allocatable **sub\_dsan**
- real \*8, dimension(:), allocatable **sub\_dsil**
- real \*8, dimension(:), allocatable **sub\_dcla**
- real \*8, dimension(:), allocatable **sub\_dsag**
- real \*8, dimension(:), allocatable **sub\_dlag**
- real \*8 **vap\_tile**
- real \*8, dimension(:), allocatable **wnan**
- real \*8, dimension(:,:), allocatable **sol\_stpwt**
- real \*8, dimension(:,:), allocatable **sub\_pst**  
*amount of pesticide in layer in subbasin (kg/ha)*
- real \*8, dimension(:,:), allocatable **sub\_hhqd**
- real \*8, dimension(:,:), allocatable **sub\_hhwtmp**
- real \*8, dimension(:,:), allocatable **huminc**  
*monthly humidity adjustment. Daily values for relative humidity within the month are rasied or lowered by the specified amount (used in climate change studies) (none)*
- real \*8, dimension(:,:), allocatable **radinc**  
*monthly solar radiation adjustment. Daily radiation within the month is raised or lowered by the specified amount (used in climate change studies) (MJ/m<sup>2</sup>)*
- real \*8, dimension(:,:), allocatable **rfinc**  
*monthly rainfall adjustment. Daily rainfall within the month is adjusted to the specified percentage of the original value (used in climate change studies)(%)*
- real \*8, dimension(:,:), allocatable **tmpinc**  
*monthly temperature adjustment. Daily maximum and minimum temperatures within the month are raised or lowered by the specified amount (used in climate change studies) (deg C)*
- real \*8, dimension(:), allocatable **ch\_k1**  
*effective hydraulic conductivity of tributary channel alluvium (mm/hr)*
- real \*8, dimension(:), allocatable **ch\_k2**  
*effective hydraulic conductivity of main channel alluvium (mm/hr)*
- real \*8, dimension(:,:), allocatable **elevb**  
*elevation at the center of the band in subbasin (m)*
- real \*8, dimension(:,:), allocatable **elevb\_fr**  
*fraction of subbasin area within elevation band (the same fractions should be listed for all HRUs within the subbasin) (none)*
- real \*8, dimension(:,:), allocatable **wndav**  
*average wind speed for the month (m/s)*
- real \*8, dimension(:), allocatable **ch\_n1**  
*Manning's "n" value for the tributary channels (none)*
- real \*8, dimension(:), allocatable **ch\_n2**  
*Manning's "n" value for the main channel (none)*
- real \*8, dimension(:), allocatable **ch\_s1**  
*average slope of tributary channels (m/m)*
- real \*8, dimension(:), allocatable **ch\_s2**  
*average slope of main channel (m/m)*
- real \*8, dimension(:), allocatable **ch\_w1**  
*average width of tributary channels (m)*
- real \*8, dimension(:), allocatable **ch\_w2**  
*average width of main channel (m)*
- real \*8, dimension(:,:), allocatable **dewpt**  
*average dew point temperature for the month (deg C)*
- real \*8, dimension(:,:), allocatable **amp\_r**  
*average fraction of total daily rainfall occuring in maximum half-hour period for month (none)*
- real \*8, dimension(:,:), allocatable **solarav**

- average daily solar radiation for the month (MJ/m<sup>2</sup>/day)*
- real \*8, dimension(:,:), allocatable [tmpstdmx](#)
- standard deviation for avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [pcf](#)
- normalization coefficient for precipitation generated from skewed distribution (none)*
- real \*8, dimension(:,:), allocatable [tmpmn](#)
- avg monthly minimum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [tmpmx](#)
- avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [tmpstdmn](#)
- standard deviation for avg monthly minimum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [otmpstdmn](#)
- real \*8, dimension(:,:), allocatable [otmpmn](#)
- real \*8, dimension(:,:), allocatable [otmpmx](#)
- real \*8, dimension(:,:), allocatable [otmpstdmx](#)
- real \*8, dimension(:,:), allocatable [ch\\_erodmo](#)
- real \*8, dimension(:,:), allocatable [uh](#)
- real \*8, dimension(:,:), allocatable [hqdsave](#)
- real \*8, dimension(:,:), allocatable [hsdsave](#)
- real \*8, dimension(:,:), allocatable [pr\\_w1](#)
- probability of wet day after dry day in month (none)*
- real \*8, dimension(:,:), allocatable [pr\\_w2](#)
- probability of wet day after wet day in month (none)*
- real \*8, dimension(:,:), allocatable [pr\\_w3](#)
- proportion of wet days in the month (none)*
- real \*8, dimension(:,:), allocatable [pcp\\_stat](#)
- real \*8, dimension(:,:), allocatable [opr\\_w1](#)
- real \*8, dimension(:,:), allocatable [opr\\_w2](#)
- real \*8, dimension(:,:), allocatable [opr\\_w3](#)
- real \*8, dimension(:,:), allocatable [opcp\\_stat](#)
- integer, dimension(:), allocatable [ireg](#)
- precipitation category (none):*
  - 1 precipitation <= 508 mm/yr*
  - 2 precipitation > 508 and <= 1016 mm/yr*
  - 3 precipitation > 1016 mm/yr*
- integer, dimension(:), allocatable [hrutot](#)
- number of HRUs in subbasin (none)*
- integer, dimension(:), allocatable [hru1](#)
- integer, dimension(:), allocatable [ihgage](#)
- HRU relative humidity data code (gage # for relative humidity data used in as HRU) (none)*
- integer, dimension(:), allocatable [isgage](#)
- HRU solar radiation data code (record # for solar radiation used in HRU) (none)*
- integer, dimension(:), allocatable [iwgage](#)
- HRU wind speed gage data code (gage # for wind speed data used in HRU) (none)*
- integer, dimension(:), allocatable [subgis](#)
- GIS code printed to output files (output.sub, .rch) (none)*
- integer, dimension(:), allocatable [irgage](#)
- subbasin rain gage data code (gage # for rainfall data used in HRU) (none)*
- integer, dimension(:), allocatable [itgage](#)
- subbasin temp gage data code (gage # for temperature data used in HRU) (none)*
- integer, dimension(:), allocatable [irelh](#)



- (none) *irelh* = 0 (dewpoint)  
*irelh* = 1 (relative humidity)  
 note: *inputs* > 1.0 (dewpoint)  
*inputs* < 1.0 (relative hum)
- integer, dimension(:), allocatable **fcst\_reg**
  - real \*8, dimension(:,:), allocatable **sol\_aorgn**  
*amount of nitrogen stored in the active organic (humic) nitrogen pool in soil layer (kg N/ha)*
  - real \*8, dimension(:,:), allocatable **sol\_fon**  
*amount of nitrogen stored in the fresh organic (residue) pool in soil layer (kg N/ha)*
  - real \*8, dimension(:,:), allocatable **sol\_tmp**  
*average temperature of soil layer on previous day or  
 daily average temperature of soil layer (deg C)*
  - real \*8, dimension(:,:), allocatable **sol\_awc**  
*available water capacity of soil layer (mm H2O/mm soil)*
  - real \*8, dimension(:,:), allocatable **volcr**  
*crack volume for soil layer (mm)*
  - real \*8, dimension(:,:), allocatable **sol\_prk**  
*percolation storage from soil layer on current day (mm H2O)*
  - real \*8, dimension(:,:), allocatable **pperco\_sub**  
*subbasin phosphorus percolation coefficient. Ratio of soluble phosphorus in surface to soluble phosphorus in percolate*
  - real \*8, dimension(:,:), allocatable **sol\_stap**  
*amount of phosphorus in the soil layer stored in the stable mineral phosphorus pool (kg P/ha)*
  - real \*8, dimension(:,:), allocatable **conv\_wt**  
*factor which converts kg/kg soil to kg/ha (none)*
  - real \*8, dimension(:,:), allocatable **sol\_actp**  
*amount of phosphorus stored in the active mineral phosphorus pool (kg P/ha)*
  - real \*8, dimension(:,:), allocatable **sol\_solp**  
*soluble P concentration in top soil layer (mg P/kg soil) or  
 amount of inorganic phosphorus stored in solution in soil layer. NOTE UNIT CHANGE! (kg P/ha)*
  - real \*8, dimension(:,:), allocatable **crdep**  
*maximum or potential crack volume (mm)*
  - real \*8, dimension(:,:), allocatable **sol\_fc**  
*amount of water available to plants in soil layer at field capacity (fc - wp water) (mm H2O)*
  - real \*8, dimension(:,:), allocatable **sol\_ul**  
*amount of water held in the soil layer at saturation (sat - wp water) (mm H2O)*
  - real \*8, dimension(:,:), allocatable **sol\_bd**  
*bulk density of the soil layer in HRU (Mg/m<sup>3</sup>)*
  - real \*8, dimension(:,:), allocatable **sol\_z**  
*depth to bottom of each soil profile layer in a given HRU (mm)*
  - real \*8, dimension(:,:), allocatable **sol\_st**  
*amount of water stored in the soil layer on any given day (less wilting point water) (mm H2O)*
  - real \*8, dimension(:,:), allocatable **sol\_up**  
*water content of soil at -0.033 MPa (field capacity) (mm H2O/mm soil)*
  - real \*8, dimension(:,:), allocatable **sol\_clay**  
*percent clay content in soil layer in HRU (UNIT CHANGE!) (% or none)*
  - real \*8, dimension(:,:), allocatable **sol\_hk**  
*beta coefficient to calculate hydraulic conductivity (none)*
  - real \*8, dimension(:,:), allocatable **flat**  
*lateral flow storage in soil layer on current day (mm H2O)*
  - real \*8, dimension(:,:), allocatable **sol\_nh3**  
*amount of nitrogen stored in the ammonium pool in soil layer (kg N/ha)*



- real \*8, dimension(:,:), allocatable **sol\_ec**  
*electrical conductivity of soil layer (dS/m)*
- real \*8, dimension(:,:), allocatable **sol\_orgn**  
*amount of nitrogen stored in the stable organic N pool. NOTE UNIT CHANGE! (mg N/kg soil or kg N/ha)*
- real \*8, dimension(:,:), allocatable **sol\_por**  
*total porosity of soil layer expressed as a fraction of the total volume (none)*
- real \*8, dimension(:,:), allocatable **sol\_wp**  
*water content of soil at -1.5 MPa (wilting point) (mm H2O/mm soil)*
- real \*8, dimension(:,:), allocatable **sol\_orgp**  
*amount of phosphorus stored in the organic P pool in soil layer. NOTE UNIT CHANGE! (mg P/kg soil or kg P/ha)*
- real \*8, dimension(:,:), allocatable **sol\_hum**  
*amount of organic matter in the soil layer classified as humic substances (kg humus/ha)*
- real \*8, dimension(:,:), allocatable **sol\_wpmm**  
*water content of soil at -1.5 MPa (wilting point) (mm H2O)*
- real \*8, dimension(:,:), allocatable **sol\_no3**  
*amount of nitrogen stored in the nitrate pool in the soil layer. This variable is read in as a concentration and converted to kg/ha (this value is read from the .sol file in units of mg/kg) (kg N/ha)*
- real \*8, dimension(:,:), allocatable **sol\_cbn**  
*percent organic carbon in soil layer (%)*
- real \*8, dimension(:,:), allocatable **sol\_k**  
*saturated hydraulic conductivity of soil layer (mm/hour)*
- real \*8, dimension(:,:), allocatable **sol\_rsd**  
*amount of organic matter in the soil layer classified as residue (kg/ha)*
- real \*8, dimension(:,:), allocatable **sol\_fop**  
*amount of phosphorus stored in the fresh organic (residue) pool in soil layer (kg P/ha)*
- real \*8, dimension(:,:), allocatable **sol\_rock**  
*percent of rock fragments in soil layer (%)*
- real \*8, dimension(:,:), allocatable **sol\_silt**  
*percent silt content in soil material (UNIT CHANGE!) (%) or none)*
- real \*8, dimension(:,:), allocatable **sol\_sand**  
*percent sand content of soil material (%)*
- real \*8, dimension(:,:), allocatable **orig\_solno3**
- real \*8, dimension(:,:), allocatable **orig\_solorgn**
- real \*8, dimension(:,:), allocatable **orig\_solsolp**
- real \*8, dimension(:,:), allocatable **orig\_solorgp**
- real \*8, dimension(:,:), allocatable **orig\_soltmp**
- real \*8, dimension(:,:), allocatable **orig\_solrsd**
- real \*8, dimension(:,:), allocatable **orig\_solfop**
- real \*8, dimension(:,:), allocatable **orig\_solfon**
- real \*8, dimension(:,:), allocatable **orig\_solaorgn**
- real \*8, dimension(:,:), allocatable **orig\_solst**
- real \*8, dimension(:,:), allocatable **orig\_solactp**
- real \*8, dimension(:,:), allocatable **orig\_solstap**
- real \*8, dimension(:,:), allocatable **orig\_volcr**
- real \*8, dimension(:,:), allocatable **conk**  
*lateral saturated hydraulic conductivity for each profile layer in a give HRU. For example (conk(2,1) is conductivity of layer from sol\_z(1,1) to sol\_z(2,1) in HRU1 (mm/hr)*
- real \*8, dimension(:,:), allocatable **sol\_pst**  
*sol\_pst(:,1) initial amount of pesticide in first layer read in from .chm file (mg/kg)*  
*sol\_pst(:,,:) amount of pesticide in soil layer. NOTE UNIT CHANGE! (kg/ha)*
- real \*8, dimension(:,:), allocatable **sol\_kp**

- pesticide sorption coefficient,  $K_p$ ; the ratio of the concentration in the solid phase to the concentration in solution ((mg/kg)/(mg/L) or  $m^3/ton$ )*
- real \*8, dimension(:,:), allocatable **orig\_solpst**
  - real \*8, dimension(:), allocatable **velsetlr**
  - real \*8, dimension(:), allocatable **velsetlp**
  - real \*8, dimension(:), allocatable **br1**
- 1st shape parameter for reservoir surface area equation (none)*
- real \*8, dimension(:), allocatable **evrsv**
- lake evaporation coefficient (none)*
- real \*8, dimension(:), allocatable **res\_k**
- hydraulic conductivity of the reservoir bottom (mm/hr)*
- real \*8, dimension(:), allocatable **lkpst\_conc**
- pesticide concentration in lake water (mg/ $m^3$ )*
- real \*8, dimension(:), allocatable **res\_evol**
- volume of water needed to fill the reservoir to the emergency spillway (read in as  $10^4 m^3$  and converted to  $m^3$ )*
- real \*8, dimension(:), allocatable **res\_pvol**
- volume of water needed to fill the reservoir to the principal spillway (read in as  $10^4 m^3$  and converted to  $m^3$ )*
- real \*8, dimension(:), allocatable **res\_vol**
- reservoir volume (read in as  $10^4 m^3$  and converted to  $m^3$ ) ( $m^3$ )*
- real \*8, dimension(:), allocatable **res\_psa**
- reservoir surface area when reservoir is filled to principal spillway (ha)*
- real \*8, dimension(:), allocatable **lkpst\_rea**
- pesticide reaction coefficient in lake water (1/day)*
- real \*8, dimension(:), allocatable **lkpst\_vol**
- pesticide volatilization coefficient in lake water (m/day)*
- real \*8, dimension(:), allocatable **br2**
- 2nd shape parameter for reservoir surface area equation (none)*
- real \*8, dimension(:), allocatable **res\_rr**
- average daily principal spillway release volume (read in as a release rate in  $m^3/s$  and converted to  $m^3/day$ )*
- real \*8, dimension(:), allocatable **res\_sed**
- amount of sediment in reservoir (read in as mg/L and converted to kg/L) (kg/L)*
- real \*8, dimension(:), allocatable **lkpst\_koc**
- pesticide partition coefficient between water and sediment in lake water ( $m^3/g$ )*
- real \*8, dimension(:), allocatable **lkpst\_mix**
- mixing velocity (diffusion/dispersion) in lake water for pesticide (m/day)*
- real \*8, dimension(:), allocatable **lkpst\_rsp**
- resuspension velocity in lake water for pesticide sorbed to sediment (m/day)*
- real \*8, dimension(:), allocatable **lkpst\_stl**
- settling velocity in lake water for pesticide sorbed to sediment (m/day)*
- real \*8, dimension(:), allocatable **lkspst\_conc**
- pesticide concentration in lake bed sediment (mg/ $m^3$ )*
- real \*8, dimension(:), allocatable **lkspst\_rea**
- pesticide reaction coefficient in lake bed sediment (1/day)*
- real \*8, dimension(:), allocatable **theta\_n**
  - real \*8, dimension(:), allocatable **theta\_p**
  - real \*8, dimension(:), allocatable **con\_nirr**
  - real \*8, dimension(:), allocatable **con\_pirr**
  - real \*8, dimension(:), allocatable **lkspst\_act**
- depth of active sediment layer in lake for for pesticide (m)*

- real \*8, dimension(:), allocatable **lkspst\_bry**  
*pesticide burial velocity in lake bed sediment (m/day)*
- real \*8, dimension(:), allocatable **sed\_stlr**
- real \*8, dimension(7) **resdata**
- real \*8, dimension(:), allocatable **res\_nsed**  
*normal amount of sediment in reservoir (read in as mg/L and convert to kg/L) (kg/L)*
- real \*8, dimension(:), allocatable **wurtmf**  
*fraction of water removed from the reservoir via WURESN which is returned and becomes flow from the reservoir outlet (none)*
- real \*8, dimension(:), allocatable **chlar**  
*chlorophyll-a production coefficient for reservoir (none)*
- real \*8, dimension(:), allocatable **res\_no3**  
*amount of nitrate in reservoir (kg N)*
- real \*8, dimension(:), allocatable **res\_orgn**  
*amount of organic N in reservoir (kg N)*
- real \*8, dimension(:), allocatable **res\_orgp**  
*amount of organic P in reservoir (kg P)*
- real \*8, dimension(:), allocatable **res\_solp**  
*amount of soluble P in reservoir (kg P)*
- real \*8, dimension(:), allocatable **res\_seci**  
*secchi-disk depth (m)*
- real \*8, dimension(:), allocatable **res\_chla**
- real \*8, dimension(:), allocatable **res\_esa**  
*reservoir surface area when reservoir is filled to emergency spillway (ha)*
- real \*8, dimension(:), allocatable **res\_nh3**  
*amount of ammonia in reservoir (kg N)*
- real \*8, dimension(:), allocatable **res\_no2**  
*amount of nitrite in reservoir (kg N)*
- real \*8, dimension(:), allocatable **seccir**  
*water clarity coefficient for reservoir (none)*
- real \*8, dimension(:), allocatable **res\_bactp**
- real \*8, dimension(:), allocatable **res\_bactlp**
- real \*8, dimension(:), allocatable **oflowmn\_fps**  
*minimum reservoir outflow as a fraction of the principal spillway volume (0-1) (fraction)*
- real \*8, dimension(:), allocatable **starg\_fps**  
*target volume as a fraction of the principal spillway volume (.1-5) (fraction)*
- real \*8, dimension(:), allocatable **weirc**
- real \*8, dimension(:), allocatable **weirk**
- real \*8, dimension(:), allocatable **weirw**
- real \*8, dimension(:), allocatable **acoef**
- real \*8, dimension(:), allocatable **bcoef**
- real \*8, dimension(:), allocatable **ccoef**
- real \*8, dimension(:), allocatable **orig\_resvol**
- real \*8, dimension(:), allocatable **orig\_rested**
- real \*8, dimension(:), allocatable **orig\_lkpstconc**
- real \*8, dimension(:), allocatable **orig\_lkpstconc**
- real \*8, dimension(:), allocatable **orig\_ressolp**
- real \*8, dimension(:), allocatable **orig\_resorgp**
- real \*8, dimension(:), allocatable **orig\_resno3**
- real \*8, dimension(:), allocatable **orig\_resno2**
- real \*8, dimension(:), allocatable **orig\_resnh3**
- real \*8, dimension(:), allocatable **orig\_resorgn**

- real \*8, dimension(:,:), allocatable **oflowmn**  
*minimum daily outflow for the month (read in as  $m^3/s$  and converted to  $m^3/day$ ) ( $m^3/day$ )*
- real \*8, dimension(:,:), allocatable **oflowmx**  
*maximum daily outflow for the month (read in as  $m^3/s$  and converted to  $m^3/day$ ) ( $m^3/day$ )*
- real \*8, dimension(:,:), allocatable **starg**  
*monthly target reservoir storage (needed if IRESKO=2) (read in as  $10^4 m^3$  and converted to  $m^3$ ) ( $m^3$ )*
- real \*8, dimension(:), allocatable **psetlr1**  
*phosphorus settling rate for mid-year period (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*
- real \*8, dimension(:), allocatable **psetlr2**  
*phosphorus settling rate for remainder of year (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*
- real \*8, dimension(:), allocatable **nsetlr1**  
*nitrogen settling rate for mid-year period (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*
- real \*8, dimension(:), allocatable **nsetlr2**  
*nitrogen settling rate for remainder of year (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*
- real \*8, dimension(:,:), allocatable **wuresn**  
*average amount of water withdrawn from reservoir each month for consumptive water use (read in as  $10^4 m^3$  and converted to  $m^3$ ) ( $m^3$ )*
- real \*8, dimension(:,:,:), allocatable **res\_out**  
*measured average daily outflow from the reservoir for the month (needed if IRESKO=1) (read in as  $m^3/s$  and converted to  $m^3/day$ ) ( $m^3/day$ )*
- integer, dimension(:), allocatable **res\_sub**  
*number of subbasin reservoir is in (weather for the subbasin is used for the reservoir) (none)*
- integer, dimension(:), allocatable **ires1**  
*beginning of mid-year nutrient settling "season" (none)*
- integer, dimension(:), allocatable **ires2**  
*end of mid-year nutrient settling "season" (none)*
- integer, dimension(:), allocatable **iresco**  
*outflow simulation code (none):*  
*0 compute outflow for uncontrolled reservoir with average annual release rate*  
*1 measured monthly outflow*  
*2 simulated controlled outflow-target release*  
*3 measured daily outflow*  
*4 stage/volume/outflow relationship*
- integer, dimension(:), allocatable **iyres**  
*year of the simulation that the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **mores**  
*month the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **iflod1r**  
*beginning month of non-flood season (needed if IRESKO=2) (none)*
- integer, dimension(:), allocatable **iflod2r**  
*ending month of non-flood season (needed if IRESKO=2) (none)*
- integer, dimension(:), allocatable **ndtargr**  
*number of days to reach target storage from current reservoir storage (needed if IRESKO=2) (days)*
- real \*8, dimension(:), allocatable **ap\_ef**  
*application efficiency (0-1) (none)*
- real \*8, dimension(:), allocatable **decay\_f**  
*exponential of the rate constant for degradation of the pesticide on foliage (none)*
- real \*8, dimension(:), allocatable **skoc**  
*soil adsorption coefficient normalized for soil organic carbon content  $((mg/kg)/(mg/L))$*
- real \*8, dimension(:), allocatable **decay\_s**  
*exponential of the rate constant for degradation of the pesticide in soil (none)*
- real \*8, dimension(:), allocatable **hlife\_f**

- half-life of pesticide on foliage (days)*
  - real \*8, dimension(:), allocatable [hlife\\_s](#)
  - half-life of pesticide in soil (days)*
    - real \*8, dimension(:), allocatable [pst\\_wof](#)
    - fraction of pesticide on foliage which is washed-off by a rainfall event (none)*
      - real \*8, dimension(:), allocatable [pst\\_wsol](#)
      - solubility of chemical in water (mg/L (ppm))*
        - real \*8, dimension(:), allocatable [irramt](#)
        - depth of irrigation water applied to HRU (mm H2O)*
          - real \*8, dimension(:), allocatable **phusw**
          - real \*8, dimension(:), allocatable **phusw\_nocrop**
          - integer, dimension(:), allocatable [pstflg](#)
          - flag for types of pesticide used in watershed. Array location is pesticide ID number*
            - 0: pesticide not used
            - 1: pesticide used
          - integer, dimension(:), allocatable [nope](#)
          - sequence number of pesticide in NPNO(:) (none)*
            - integer, dimension(:), allocatable **nop**
            - integer, dimension(:), allocatable **yr\_skip**
            - integer, dimension(:), allocatable **isweep**
            - integer, dimension(:), allocatable **icrmx**
            - integer, dimension(:), allocatable **nopmx**
            - integer, dimension(:, :), allocatable **mgtop**
            - integer, dimension(:, :), allocatable **idop**
            - integer, dimension(:, :), allocatable **mgt1iop**
            - integer, dimension(:, :), allocatable **mgt2iop**
            - integer, dimension(:, :), allocatable **mgt3iop**
            - real \*8, dimension(:, :), allocatable **mgt4op**
            - real \*8, dimension(:, :), allocatable **mgt5op**
            - real \*8, dimension(:, :), allocatable **mgt6op**
            - real \*8, dimension(:, :), allocatable **mgt7op**
            - real \*8, dimension(:, :), allocatable **mgt8op**
            - real \*8, dimension(:, :), allocatable **mgt9op**
            - real \*8, dimension(:, :), allocatable **mgt10iop**
            - real \*8, dimension(:, :), allocatable **phu\_op**
            - real \*8, dimension(:), allocatable [cnyld](#)
            - fraction of nitrogen in yield (kg N/kg yield)*
              - real \*8, dimension(:), allocatable [rsdco\\_pl](#)
              - plant residue decomposition coefficient. The fraction of residue which will decompose in a day assuming optimal moisture, temperature, C:N ratio, and C:P ratio (none)*
                - real \*8, dimension(:), allocatable [wac21](#)
                - 1st shape parameter for radiation use efficiency equation (none)*
                  - real \*8, dimension(:), allocatable [wac22](#)
                  - 2nd shape parameter for radiation use efficiency equation (none)*
                    - real \*8, dimension(:), allocatable [alai\\_min](#)
                    - minimum LAI during winter dormant period ( $m^2/m^2$ )*
                      - real \*8, dimension(:), allocatable [leaf1](#)
                      - 1st shape parameter for leaf area development equation (none)*
                        - real \*8, dimension(:), allocatable [leaf2](#)
                        - 2nd shape parameter for leaf area development equation (none)*
                          - real \*8, dimension(:), allocatable [wsyf](#)
                          - Value of harvest index between 0 and HVSTI which represents the lowest value expected due to water stress ((kg/ha)/(kg/ha))*

- real \*8, dimension(:), allocatable **bio\_e**  
*biomass-energy ratio. The potential (unstressed) growth rate per unit of intercepted photosynthetically active radiation ((kg/ha)/(MJ/m\*\*2))*
- real \*8, dimension(:), allocatable **hvsti**  
*harvest index: crop yield/aboveground biomass ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable **t\_base**  
*minimum temperature for plant growth (deg C)*
- real \*8, dimension(:), allocatable **t\_opt**  
*optimal temperature for plant growth (deg C)*
- real \*8, dimension(:), allocatable **chtmx**  
*maximum canopy height (m)*
- real \*8, dimension(:), allocatable **cvm**  
*natural log of USLE\_C (the minimum value of the USLE C factor for the land cover) (none)*
- real \*8, dimension(:), allocatable **gsi**  
*maximum stomatal conductance (m/s)*
- real \*8, dimension(:), allocatable **vpd2**  
*rate of decline in stomatal conductance per unit increase in vapor pressure deficit ((m/s)\*(1/kPa))*
- real \*8, dimension(:), allocatable **wavp**  
*rate of decline in radiation use efficiency as a function of vapor pressure deficit (none)*
- real \*8, dimension(:), allocatable **bio\_leaf**  
*fraction of leaf/needle biomass that drops during dormancy (for trees only) (none)*
- real \*8, dimension(:), allocatable **blai**  
*maximum (potential) leaf area index (none)*
- real \*8, dimension(:), allocatable **cpyld**  
*fraction of phosphorus in yield (kg P/kg yield)*
- real \*8, dimension(:), allocatable **dlai**  
*fraction of growing season when leaf area declines (none)*
- real \*8, dimension(:), allocatable **rdmx**  
*maximum root depth of plant (m)*
- real \*8, dimension(:), allocatable **bio\_n1**  
*1st shape parameter for plant N uptake equation (none)*
- real \*8, dimension(:), allocatable **bio\_n2**  
*2nd shape parameter for plant N uptake equation (none)*
- real \*8, dimension(:), allocatable **bio\_p1**  
*1st shape parameter for plant P uptake equation (none)*
- real \*8, dimension(:), allocatable **bio\_p2**  
*2st shape parameter for plant P uptake equation (none)*
- real \*8, dimension(:), allocatable **bm\_dieoff**  
*fraction above ground biomass that dies off at dormancy (fraction)*
- real \*8, dimension(:), allocatable **bm\_x\_trees**
- real \*8, dimension(:), allocatable **ext\_coef**
- real \*8, dimension(:), allocatable **rsr1**  
*initial root to shoot ratio at the beg of growing season*
- real \*8, dimension(:), allocatable **rsr2**  
*root to shoot ratio at the end of the growing season*
- real \*8, dimension(:), allocatable **pltnfr1**  
*nitrogen uptake parameter #1: normal fraction of N in crop biomass at emergence (kg N/kg biomass)*
- real \*8, dimension(:), allocatable **pltnfr2**  
*nitrogen uptake parameter #2: normal fraction of N in crop biomass at 0.5 maturity (kg N/kg biomass)*
- real \*8, dimension(:), allocatable **pltnfr3**  
*nitrogen uptake parameter #3: normal fraction of N in crop biomass at maturity (kg N/kg biomass)*

- real \*8, dimension(:), allocatable [pltpfr1](#)  
*phosphorus uptake parameter #1: normal fraction of P in crop biomass at emergence (kg P/kg biomass)*
- real \*8, dimension(:), allocatable [pltpfr2](#)  
*phosphorus uptake parameter #2: normal fraction of P in crop biomass at 0.5 maturity (kg P/kg biomass)*
- real \*8, dimension(:), allocatable [pltpfr3](#)  
*phosphorus uptake parameter #3: normal fraction of P in crop biomass at maturity (kg P/kg biomass)*
- integer, dimension(:), allocatable [idc](#)  
*crop/landcover category (none):*  
1 warm season annual legume  
2 cold season annual legume  
3 perennial legume  
4 warm season annual  
5 cold season annual  
6 perennial  
7 trees
- integer, dimension(:), allocatable [mat\\_yrs](#)
- real \*8, dimension(:), allocatable [bactpdb](#)  
*concentration of persistent bacteria in manure (fertilizer) (cfu/g manure)*
- real \*8, dimension(:), allocatable [fminn](#)  
*fraction of fertilize/manure that is mineral nitrogen (NO<sub>3</sub> + NH<sub>3</sub>) (kg minN/kg fert)*
- real \*8, dimension(:), allocatable [forgn](#)  
*fraction of organic nitrogen in fertilizer/manure (kg orgN/kg fert)*
- real \*8, dimension(:), allocatable [forgp](#)  
*fraction of fertilizer/manure that is organic phosphorus (kg orgP/kg fert)*
- real \*8, dimension(:), allocatable [bactkddb](#)  
*fraction of bacteria in solution (the remaining fraction is sorbed to soil particles) (none):*  
1: all bacteria in solution  
0: all bacteria sorbed to soil particles
- real \*8, dimension(:), allocatable [bactlpdb](#)  
*concentration of less persistent bacteria in manure (fertilizer) (cfu/g manure)*
- real \*8, dimension(:), allocatable [fminp](#)  
*fraction of fertilizer that is mineral phosphorus in fertilizer/manure (kg minP/kg fert)*
- real \*8, dimension(:), allocatable [fnh3n](#)  
*fraction of mineral N content that is NH<sub>3</sub>-N in fertilizer/manure (kg NH<sub>3</sub>-N/kg minN)*
- character(len=8), dimension(200) [fertnm](#)  
*name of fertilizer*
- real \*8, dimension(:), allocatable [curbden](#)  
*curb length density in HRU (km/ha)*
- real \*8, dimension(:), allocatable [dirtmx](#)  
*maximum amount of solids allowed to build up on impervious surfaces (kg/curb km)*
- real \*8, dimension(:), allocatable [fimp](#)  
*fraction of HRU area that is impervious (both directly and indirectly connected)(fraction)*
- real \*8, dimension(:), allocatable [urbcoef](#)  
*wash-off coefficient for removal of constituents from an impervious surface (1/mm)*
- real \*8, dimension(:), allocatable [thalf](#)  
*time for the amount of solids on impervious areas to build up to 1/2 the maximum level (days)*
- real \*8, dimension(:), allocatable [tnconc](#)  
*concentration of total nitrogen in suspended solid load from impervious areas (mg N/kg sed)*
- real \*8, dimension(:), allocatable [tno3conc](#)  
*concentration of NO<sub>3</sub>-N in suspended solid load from impervious areas (mg NO<sub>3</sub>-N/kg sed)*
- real \*8, dimension(:), allocatable [tpconc](#)  
*concentration of total phosphorus in suspended solid load from impervious areas (mg P/kg sed)*



- real \*8, dimension(:), allocatable **fcimp**  
*fraction of HRU area that is classified as directly connected impervious (fraction)*
- real \*8, dimension(:), allocatable **urbcn2**  
*SCS curve number for moisture condition II in impervious areas (none)*
- real \*8 **fr\_curb**  
*availability factor, the fraction of the curb length that is sweepable (none)*
- real \*8 **frt\_kg**  
*amount of fertilizer applied to HRU (kg/ha)*
- real \*8 **pst\_dep**  
*depth of pesticide in the soil (mm)*
- real \*8 **sweepeff**
- real \*8, dimension(:), allocatable **ranrns\_hru**  
*random roughness for a given HRU (mm)*
- integer, dimension(:), allocatable **itill**
- real \*8, dimension(:), allocatable **deptil**  
*depth of mixing caused by tillage operation (mm)*
- real \*8, dimension(:), allocatable **effmix**  
*mixing efficiency of tillage operation (none)*
- real \*8, dimension(:), allocatable **ranrns**  
*random roughness of a given tillage operation (mm)*
- character(len=8), dimension(550) **tillnm**  
*8-character name for the tillage operation*
- real \*8, dimension(:), allocatable **rnum1s**  
*For ICODES equal to (none)*  
*0,1,3,5,9: not used*  
*2: Fraction of flow in channel*  
*4: amount of water transferred (as defined by INUM4S)*  
*7,8,10,11: drainage area in square kilometers associated with the record file*  
*12: reparation coefficient.*
- real \*8, dimension(:), allocatable **hyd\_dakm**  
*total drainage area of hydrograph in square kilometers ( $\text{km}^2$ )*
- real \*8, dimension(:,:), allocatable **varoute**
- real \*8, dimension(:,:), allocatable **shyd**
- real \*8, dimension(:,:), allocatable **vartran**
- real \*8, dimension(:,:), allocatable **hhvaroute**
- integer, dimension(:), allocatable **icodes**  
*routing command code (none):*  
*0 = finish*  
*1 = subbasin*  
*2 = route*  
*3 = routres*  
*4 = transfer*  
*5 = add*  
*6 = rechour*  
*7 = recmon*  
*8 = recyear*  
*9 = save*  
*10 = recday*  
*11 = recnst*  
*12 = structure*  
*13 = apex*  
*14 = saveconc*  
*15 =*
- integer, dimension(:), allocatable **ihouts**



*For ICODES equal to (none)*

*0: not used*

*1,2,3,5,7,8,10,11: hydrograph storage location number*

*4: departure type (1=reach, 2=reservoir)*

*9: hydrograph storage location of data to be printed to event file*

*14: hydrograph storage location of data to be printed to saveconc file.*

- integer, dimension(:), allocatable **inum1s**

*For ICODES equal to (none)*

*0: not used*

*1: subbasin number*

*2: reach number*

*3: reservoir number*

*4: reach or res # flow is diverted from*

*5: hydrograph storage location of 1st dataset to be added*

*7,8,9,10,11,14: file number.*

- integer, dimension(:), allocatable **inum2s**

*For ICODES equal to (none)*

*0,1,7,8,10,11: not used*

*2,3: inflow hydrograph storage location*

*4: destination type (1=reach, 2=reservoir)*

*5: hydrograph storage location of 2nd dataset to be added*

*9,14: print frequency (0=daily, 1=hourly)*

- integer, dimension(:), allocatable **inum3s**

*For ICODES equal to (none)*

*0,1,5,7,8,10,11: not used*

*2,3: subbasin number 4: destination number. Reach or reservoir receiving water*

*9: print format (0=normal, fixed format; 1=txt format for AV interface, recday)*

- integer, dimension(:), allocatable **inum4s**

*For ICODES equal to (none)*

*0,2,3,5,7,8,9,10,11: not used*

*1: GIS code printed to output file (optional)*

*4: rule code governing transfer of water (1=fraction transferred out, 2=min volume or flow left, 3=exact amount transferred)*

- integer, dimension(:), allocatable **inum5s**
- integer, dimension(:), allocatable **inum6s**
- integer, dimension(:), allocatable **inum7s**
- integer, dimension(:), allocatable **inum8s**
- integer, dimension(:), allocatable **subed**
- character(len=10), dimension(:), allocatable **recmonps**
- character(len=10), dimension(:), allocatable **reccnsteps**
- character(len=5), dimension(:), allocatable **subnum**
- character(len=4), dimension(:), allocatable **hruno**
- real \*8, dimension(:), allocatable **grwat\_n**  
*Mannings's n for grassed waterway (none)*
- integer, dimension(:), allocatable **grwat\_i**  
*flag for the simulation of grass waterways (none)*  
*= 0 inactive*  
*= 1 active*
- real \*8, dimension(:), allocatable **grwat\_l**  
*length of grass waterway (km)*
- real \*8, dimension(:), allocatable **grwat\_w**  
*average width of grassed waterway (m)*
- real \*8, dimension(:), allocatable **grwat\_d**  
*depth of grassed waterway from top of bank to bottom (m)*
- real \*8, dimension(:), allocatable **grwat\_s**  
*average slope of grassed waterway channel (m)*
- real \*8, dimension(:), allocatable **grwat\_spcon**

- linear parameter for calculating sediment in grassed waterways (none)*
- real \*8, dimension(:), allocatable **tc\_gwat**
- real \*8, dimension(:), allocatable **pot\_volmm**
- real \*8, dimension(:), allocatable **pot\_tilemm**
- real \*8, dimension(:), allocatable **pot\_volxmm**
- real \*8, dimension(:), allocatable **pot\_fr**
- fraction of HRU area that drains into pothole ( $\text{km}^2/\text{km}^2$ )*
- real \*8, dimension(:), allocatable **pot\_tile**
- average daily outflow to main channel from tile flow if drainage tiles are installed in pothole (needed only if current HRU is IPOT) ( $\text{m}^3/\text{s}$ )*
- real \*8, dimension(:), allocatable **pot\_vol**
- initial or current volume of water stored in the depression/impounded area (read in as mm and converted to  $\text{m}^3$ ) (needed only if current HRU is IPOT) (mm or  $\text{m}^3 \text{H2O}$ )*
- real \*8, dimension(:), allocatable **potsa**
- surface area of impounded water body (ha)*
- real \*8, dimension(:), allocatable **pot\_volx**
- maximum volume of water stored in the depression/impounded area (read in as mm and converted to  $\text{m}^3$ ) (needed only if current HRU is IPOT) (mm)*
- real \*8, dimension(:), allocatable **wfsh**
- wetting front matric potential (average capillary suction at wetting front) (mm)*
- real \*8, dimension(:), allocatable **potflwi**
- real \*8, dimension(:), allocatable **potsedi**
- real \*8, dimension(:), allocatable **pot\_no3l**
- nitrate decay rate in impounded area (1/day)*
- real \*8, dimension(:), allocatable **pot\_nsed**
- normal sediment concentration in impounded water (needed only if current HRU is IPOT)(mg/L)*
- real \*8, dimension(:), allocatable **gwno3**
- nitrate-N concentration in groundwater loading to reach (mg N/L)*
- real \*8, dimension(:), allocatable **newrti**
- infiltration rate for last time step from the previous day (mm/hr)*
- real \*8, dimension(:), allocatable **fsred**
- reduction in bacteria loading from filter strip (none)*
- real \*8, dimension(:), allocatable **pot\_sed**
- real \*8, dimension(:), allocatable **pot\_no3**
- real \*8, dimension(:), allocatable **tmpavp**
- real \*8, dimension(:), allocatable **dis\_stream**
- average distance to stream (m)*
- real \*8, dimension(:), allocatable **evpot**
- pothole evaporation coefficient (none)*
- real \*8, dimension(:), allocatable **pot\_solpl**
- real \*8, dimension(:), allocatable **sed\_con**
- real \*8, dimension(:), allocatable **orgn\_con**
- real \*8, dimension(:), allocatable **orgp\_con**
- real \*8, dimension(:), allocatable **pot\_k**
- hydraulic conductivity of soil surface of pothole defaults to conductivity of upper soil (0.↵ 01-10.) layer*
- real \*8, dimension(:), allocatable **soln\_con**
- real \*8, dimension(:), allocatable **solp\_con**
- real \*8, dimension(:), allocatable **n\_reduc**
- nitrogen uptake reduction factor (not currently used; defaulted 300.)*
- real \*8, dimension(:), allocatable **n\_lag**
- lag coefficient for calculating nitrate concentration in subsurface drains (0.001 - 1.0) (dimensionless)*
- real \*8, dimension(:), allocatable **n\_ln**

- power function exponent for calculating nitrate concentration in subsurface drains (1.0 - 3.0) (dimensionless)*

  - real \*8, dimension(:), allocatable [n\\_lnco](#)
- coefficient for power function for calculating nitrate concentration in subsurface drains (0.5 - 4.0) (dimensionless)*

  - integer, dimension(:), allocatable [ioper](#)
- integer, dimension(:), allocatable*

  - integer, dimension(:), allocatable [ngrwat](#)
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [usle\\_ls](#)

*USLE equation length slope (LS) factor (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [filterw](#)

*filter strip width for bacteria transport (m)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [phuacc](#)

*fraction of plant heat units accumulated (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [sumix](#)

*sum of all tillage mixing efficiencies for HRU operation (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [epco](#)

*plant water uptake compensation factor (0-1) (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [esco](#)

*soil evaporation compensation factor (0-1) (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [hru\\_slp](#)

*average slope steepness in HRU (m/m)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [slsubbsn](#)

*average slope length for subbasin (m)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [erorgn](#)

*organic N enrichment ratio, if left blank the model will calculate for every event (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [erorgp](#)

*organic P enrichment ratio, if left blank the model will calculate for every event (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [biomix](#)

*biological mixing efficiency. Mixing of soil due to activity of earthworms and other soil biota. Mixing is performed at the end of every calendar year (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [pnd\\_seci](#)

*secchi-disk depth of pond (m)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [canmx](#)

*maximum canopy storage (mm H2O)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [divmax](#)

*maximum daily irrigation diversion from the reach (when IRRSC=1 or IRR=3): when value is positive the units are mm H2O; when the value is negative, the units are (10<sup>4</sup> m<sup>3</sup> H2O) (mm H2O or 10<sup>4</sup> m<sup>3</sup> H2O)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [flowmin](#)

*minimum instream flow for irrigation diversions when IRRSC=1, irrigation water will be diverted only when streamflow is at or above FLOWMIN (m<sup>3</sup>/s)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [usle\\_p](#)

*USLE equation support practice (P) factor (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [lat\\_sed](#)

*sediment concentration in lateral flow (g/L)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [rch\\_dakm](#)

*total drainage area contributing to flow at the outlet (pour point) of the reach in square kilometers (km<sup>2</sup>)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [cn1](#)

*SCS runoff curve number for moisture condition I (none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [pnd\\_no3s](#)

*amount of nitrate originating from lateral flow in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [lat\\_ttime](#)

*lateral flow travel time or exponential of the lateral flow travel time (days or none)*
- real \*8, dimension(:), allocatable*

  - real \*8, dimension(:), allocatable [cn2](#)

- SCS runoff curve number for moisture condition II (none)*

  - real \*8, dimension(:), allocatable [flowfr](#)  
*fraction of available flow in reach that is allowed to be applied to the HRU (none)*
- real \*8, dimension(:), allocatable [sol\\_zmx](#)  
*maximum rooting depth (mm)*
- real \*8, dimension(:), allocatable [tile\\_ttime](#)  
*exponential of the tile flow travel time (none)*
- real \*8, dimension(:), allocatable [slsoil](#)  
*slope length for lateral subsurface flow (m)*
- real \*8, dimension(:), allocatable [gwmimp](#)  
*soluble P concentration in groundwater loading to reach (mg P/L)*
- real \*8, dimension(:), allocatable [sol\\_cov](#)  
*amount of residue on soil surface (kg/ha)*
- real \*8, dimension(:), allocatable [sed\\_stl](#)  
*fraction of sediment remaining suspended in impoundment after settling for one day (kg/kg)*
- real \*8, dimension(:), allocatable [ov\\_n](#)  
*Manning's "n" value for overland flow (none)*
- real \*8, dimension(:), allocatable [pnd\\_no3](#)  
*amount of nitrate originating from surface runoff in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [pnd\\_solp](#)  
*amount of soluble P originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [yldanu](#)  
*annual yield (dry weight) in the HRU (metric tons/ha)*
- real \*8, dimension(:), allocatable [driftco](#)  
*coefficient for pesticide drift directly onto stream (none)*
- real \*8, dimension(:), allocatable [pnd\\_orgn](#)  
*amount of organic N originating from surface runoff in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [pnd\\_orgp](#)  
*amount of organic P originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [cn3](#)  
*SCS runoff curve number for moisture condition III (none)*
- real \*8, dimension(:), allocatable [twlpnd](#)  
*water lost through seepage from ponds on day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [twlwet](#)  
*water lost through seepage from wetlands on day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [hru\\_fr](#)  
*fraction of subbasin area contained in HRU ( $\text{km}^2/\text{km}^2$ )*
- real \*8, dimension(:), allocatable [sol\\_sumul](#)  
*amount of water held in soil profile at saturation (mm H2O)*
- real \*8, dimension(:), allocatable [pnd\\_chla](#)  
*amount of chlorophyll-a in pond at end of day (kg chl\_a)*
- real \*8, dimension(:), allocatable [hru\\_km](#)  
*area of HRU in square kilometers ( $\text{km}^2$ )*
- real \*8, dimension(:), allocatable [bio\\_ms](#)  
*land cover/crop biomass (dry weight) (kg/ha)*
- real \*8, dimension(:), allocatable [sol\\_alb](#)  
*albedo when soil is moist (none)*
- real \*8, dimension(:), allocatable [strsw](#)  
*fraction of potential plant growth achieved on the day where the reduction is caused by water stress (none)*
- real \*8, dimension(:), allocatable [pnd\\_fr](#)  
*fraction of HRU/subbasin area that drains into ponds (none)*

- real \*8, dimension(:), allocatable [pnd\\_k](#)  
*hydraulic conductivity through bottom of ponds (mm/hr)*
- real \*8, dimension(:), allocatable [pnd\\_psa](#)  
*surface area of ponds when filled to principal spillway (ha)*
- real \*8, dimension(:), allocatable [pnd\\_pvol](#)  
*runoff volume from catchment area needed to fill the ponds to the principal spillway (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [pnd\\_esa](#)  
*surface area of ponds when filled to emergency spillway (ha)*
- real \*8, dimension(:), allocatable [pnd\\_evol](#)  
*runoff volume from catchment area needed to fill the ponds to the emergency spillway (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [pnd\\_vol](#)  
*volume of water in ponds (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [yldaa](#)  
*average annual yield (dry weight) in the HRU (metric tons)*
- real \*8, dimension(:), allocatable [pnd\\_nsed](#)  
*normal sediment concentration in pond water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [pnd\\_sed](#)  
*sediment concentration in pond water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [dep\\_imp](#)  
*depth to impervious layer (mm)*
- real \*8, dimension(:), allocatable **strsa**
- real \*8, dimension(:), allocatable **evpnd**
- real \*8, dimension(:), allocatable **evwet**
- real \*8, dimension(:), allocatable [wet\\_fr](#)  
*fraction of HRU/subbasin area that drains into wetlands (none)*
- real \*8, dimension(:), allocatable [wet\\_k](#)  
*hydraulic conductivity of bottom of wetlands (mm/hr)*
- real \*8, dimension(:), allocatable [wet\\_nsa](#)  
*surface area of wetlands in subbasin at normal water level (ha)*
- real \*8, dimension(:), allocatable [wet\\_nvol](#)  
*runoff volume from catchment area needed to fill wetlands to normal water level (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- integer, dimension(:), allocatable **iwetgw**
- integer, dimension(:), allocatable **iwetile**
- real \*8, dimension(:), allocatable [wet\\_mxsa](#)  
*surface area of wetlands at maximum water level (ha)*
- real \*8, dimension(:), allocatable [wet\\_mxvol](#)  
*runoff volume from catchment area needed to fill wetlands to maximum water level (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [wet\\_vol](#)  
*volume of water in wetlands (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [wet\\_nsed](#)  
*normal sediment concentration in wetland water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [wet\\_sed](#)  
*sediment concentration in wetland water (UNIT CHANGE!) (mg/L or kg/L)*
- real \*8, dimension(:), allocatable [bp1](#)  
*1st shape parameter for pond surface area equation (none)*
- real \*8, dimension(:), allocatable [bp2](#)  
*2nd shape parameter for the pond surface area equation (none)*
- real \*8, dimension(:), allocatable [sci](#)

- retention coefficient for CN method based on plant ET (none)*
- real \*8, dimension(:), allocatable [smx](#)
- retention coefficient for CN method based on soil moisture (none)*
- real \*8, dimension(:), allocatable [bw1](#)
- 1st shape parameter for the wetland surface area equation (none)*
- real \*8, dimension(:), allocatable [bw2](#)
- 2nd shape parameter for the wetland surface area equation (none)*
- real \*8, dimension(:), allocatable [bactpq](#)
- persistent bacteria in soil solution (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [cnday](#)
- curve number for current day, HRU and at current soil moisture (none)*
- real \*8, dimension(:), allocatable [bactlp\\_plt](#)
- less persistent bacteria on foliage (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [bactp\\_plt](#)
- persistent bacteria on foliage (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [auto\\_eff](#)
- fertilizer application efficiency calculated as the amount of N applied divided by the amount of N removed at harvest (none)*
- real \*8, dimension(:), allocatable [secciw](#)
- water clarity coefficient for wetland (none)*
- real \*8, dimension(:), allocatable [sol\\_sw](#)
- amount of water stored in soil profile on current day (mm H2O)*
- real \*8, dimension(:), allocatable [bactlpq](#)
- less persistent bacteria in soil solution (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [chlaw](#)
- chlorophyll-a production coefficient for wetland (none)*
- real \*8, dimension(:), allocatable [tmpav](#)
- average air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [bactlps](#)
- less persistent bacteria attached to soil particles (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [bactps](#)
- persistent bacteria attached to soil particles (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [sno\\_hru](#)
- amount of water stored as snow in HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable [wet\\_orgn](#)
- amount of organic N originating from surface runoff in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [hru\\_ra](#)
- solar radiation for the day in HRU (MJ/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [subp](#)
- precipitation for the day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [rsdin](#)
- initial residue cover (kg/ha)*
- real \*8, dimension(:), allocatable [tmn](#)
- minimum air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [tmx](#)
- maximum air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [tmp\\_hi](#)
- real \*8, dimension(:), allocatable [tmp\\_lo](#)
- real \*8, dimension(:), allocatable [usle\\_k](#)
- USLE equation soil erodibility (K) factor (none)*
- real \*8, dimension(:), allocatable [tconc](#)

- time of concentration for HRU (hour)*
- real \*8, dimension(:), allocatable [hru\\_rmx](#)
- maximum possible solar radiation for the day in HRU (MJ/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [rwt](#)
- fraction of total plant biomass that is in roots (none)*
- real \*8, dimension(:), allocatable [olai](#)
- real \*8, dimension(:), allocatable [usle\\_cfacs](#)
- real \*8, dimension(:), allocatable [usle\\_eifacs](#)
- real \*8, dimension(:), allocatable [sol\\_sumfc](#)
- amount of water held in soil profile at field capacity (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [t\\_ov](#)
- time for flow from farthest point in subbasin to enter a channel (hour)*
- real \*8, dimension(:), allocatable [anano3](#)
- total amount of NO<sub>3</sub> applied during the year in auto-fertilization (kg N/ha)*
- real \*8, dimension(:), allocatable [aird](#)
- amount of water applied to HRU on current day (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [wet\\_orgp](#)
- amount of organic P originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [sol\\_avpor](#)
- average porosity for entire soil profile (none)*
- real \*8, dimension(:), allocatable [usle\\_mult](#)
- product of USLE K,P,LS,exp(rock) (none)*
- real \*8, dimension(:), allocatable [rhd](#)
- relative humidity for the day in HRU (none)*
- real \*8, dimension(:), allocatable [u10](#)
- wind speed (measured at 10 meters above surface) for the day in HRU (m/s)*
- real \*8, dimension(:), allocatable [cht](#)
- canopy height (m)*
- real \*8, dimension(:), allocatable [aairr](#)
- average annual amount of irrigation water applied to HRU (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [lai\\_aamx](#)
- maximum leaf area index for the entire period of simulation in the HRU (none)*
- real \*8, dimension(:), allocatable [deepirr](#)
- amount of water removed from deep aquifer for irrigation (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [shallirr](#)
- amount of water removed from shallow aquifer for irrigation (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [ch\\_l1](#)
- longest tributary channel length in subbasin (km)*
- real \*8, dimension(:), allocatable [wet\\_no3](#)
- amount of nitrate originating from surface runoff in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [ovrlnd](#)
- overland flow onto HRU from upstream routing unit (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [canstor](#)
- amount of water held in canopy storage (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [irr\\_mx](#)
- maximum irrigation amount per auto application (mm)*
- real \*8, dimension(:), allocatable [auto\\_wstr](#)
- water stress factor which triggers auto irrigation (none or mm)*
- real \*8, dimension(:), allocatable [cfrt\\_id](#)
- fertilizer/manure identification number from database (fert.dat) (none)*
- real \*8, dimension(:), allocatable [cfrt\\_kg](#)



- amount of fertilzier/manure applied to HRU on a given day ((kg/ha)/day)*
- real \*8, dimension(:), allocatable **cpst\_id**
- real \*8, dimension(:), allocatable **cpst\_kg**
- real \*8, dimension(:), allocatable **irr\_asq**
- surface runoff ratio*
- real \*8, dimension(:), allocatable **irr\_eff**
- real \*8, dimension(:), allocatable **irrsq**
- surface runoff ratio (0-1) .1 is 10% surface runoff (frac)*
- real \*8, dimension(:), allocatable **irrsalt**
- concentration of salt in irrigation water (mg/kg)*
- real \*8, dimension(:), allocatable **irrefm**
- real \*8, dimension(:), allocatable **bio\_eat**
- dry weight of biomass removed by grazing daily ((kg/ha)/day)*
- real \*8, dimension(:), allocatable **bio\_trmp**
- dry weight of biomass removed by trampling daily ((kg/ha)/day)*
- integer, dimension(:), allocatable **ipst\_freq**
- number of days between applications (days)*
- integer, dimension(:), allocatable **ifrt\_freq**
- number of days between applications in continuous fertilizer operation (days)*
- integer, dimension(:), allocatable **irr\_noa**
- integer, dimension(:), allocatable **irr\_sc**
- integer, dimension(:), allocatable **irr\_no**
- integer, dimension(:), allocatable **imp\_trig**
- release/impound action code (none):*
  - 0 begin impounding water*
  - 1 release impounded water*
- integer, dimension(:), allocatable **fert\_days**
- number of days continuous fertilization will be simulated (none)*
- integer, dimension(:), allocatable **irr\_sca**
- integer, dimension(:), allocatable **idplt**
- land cover/crop identification code for first crop grown in HRU (the only crop if there is no rotation) (from crop.dat)*
  - (none)*
- integer, dimension(:), allocatable **wstrs\_id**
- water stress identifier (none):*
  - 1 plant water demand*
  - 2 soil water deficit*
- integer, dimension(:), allocatable **pest\_days**
- real \*8, dimension(:,:), allocatable **bio\_aahv**
- real \*8, dimension(:), allocatable **cumei**
- real \*8, dimension(:), allocatable **cumeira**
- real \*8, dimension(:), allocatable **cumrt**
- real \*8, dimension(:), allocatable **cumrai**
- real \*8, dimension(:), allocatable **wet\_solp**
- amount of soluble P originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable **wet\_chla**
- amount of chlorophyll-a in wetland at end of day (kg chla)*
- real \*8, dimension(:), allocatable **wet\_no3s**
- amount of nitrate originating from lateral flow in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable **pstsol**
- amount of soluble pesticide leached from bottom of soil profile on current day (kg pst/ha)*
- real \*8, dimension(:), allocatable **pnd\_no3g**
- amount of nitrate originating from groundwater in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable **wet\_seci**



- secchi-disk depth in wetland at end of day (m)*
- real \*8, dimension(:), allocatable [delay](#)  
*groundwater delay: time required for water leaving the bottom of the root zone to reach the shallow aquifer (days)*
- real \*8, dimension(:), allocatable [gwht](#)  
*groundwater height (m)*
- real \*8, dimension(:), allocatable [gw\\_q](#)  
*groundwater contribution to streamflow from HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable [pnd\\_solpg](#)  
*amount of soluble P originating from groundwater in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [alpha\\_bf](#)  
*alpha factor for groundwater recession curve (1/days)*
- real \*8, dimension(:), allocatable [alpha\\_bfe](#)  
 $\exp(-\alpha_{bf})$  (none)
- real \*8, dimension(:), allocatable [gw\\_spyld](#)  
*specific yield for shallow aquifer ( $m^3/m^3$ )*
- real \*8, dimension(:), allocatable [alpha\\_bf\\_d](#)  
*alpha factor for groundwater recession curve of the deep aquifer (1/days)*
- real \*8, dimension(:), allocatable [alpha\\_bfe\\_d](#)  
 $\exp(-\alpha_{bf_d})$  for deep aquifer (none)
- real \*8, dimension(:), allocatable [gw\\_qdeep](#)  
*groundwater contribution to streamflow from deep aquifer from HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable [gw\\_delaye](#)  
 $\exp(-1/delay)$  where *delay(:)* is the groundwater delay (time required for water leaving the bottom of the root zone to reach the shallow aquifer; units-days) (none)
- real \*8, dimension(:), allocatable [gw\\_revap](#)  
*revap coeff: this variable controls the amount of water moving from the shallow aquifer to the root zone as a result of soil moisture depletion (none)*
- real \*8, dimension(:), allocatable [rchrg\\_dp](#)  
*recharge to deep aquifer: the fraction of root zone percolation that reaches the deep aquifer (none)*
- real \*8, dimension(:), allocatable [anion\\_excl](#)  
*fraction of porosity from which anions are excluded*
- real \*8, dimension(:), allocatable [revapmn](#)  
*threshold depth of water in shallow aquifer required to allow revap to occur (mm H2O)*
- real \*8, dimension(:), allocatable [rchrg](#)  
*amount of water recharging both aquifers on current day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [bio\\_min](#)  
*minimum plant biomass for grazing (kg/ha)*
- real \*8, dimension(:), allocatable [ffc](#)  
*initial HRU soil water content expressed as fraction of field capacity (none)*
- real \*8, dimension(:), allocatable [surqsolp](#)  
*amount of soluble phosphorus in surface runoff in HRU for the day (kg P/ha)*
- real \*8, dimension(:), allocatable [deepst](#)  
*depth of water in deep aquifer (mm H2O)*
- real \*8, dimension(:), allocatable [shallst](#)  
*depth of water in shallow aquifer in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [wet\\_solpg](#)  
*amount of soluble P originating from groundwater in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [cklsp](#)
- real \*8, dimension(:), allocatable [rchrg\\_src](#)
- real \*8, dimension(:), allocatable [trapeff](#)  
*filter strip trapping efficiency (used for everything but bacteria) (none)*

- real \*8, dimension(:), allocatable [sol\\_avbd](#)  
*average bulk density for soil profile (Mg/m<sup>3</sup>)*
- real \*8, dimension(:), allocatable [wet\\_no3g](#)  
*amount of nitrate originating from groundwater in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [tdrain](#)  
*time to drain soil to field capacity yield used in autofertilization (hours)*
- real \*8, dimension(:), allocatable [gwqmn](#)  
*threshold depth of water in shallow aquifer required before groundwater flow will occur (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [snotmp](#)  
*temperature of snow pack in HRU (deg C)*
- real \*8, dimension(:), allocatable [pplnt](#)  
*plant uptake of phosphorus in HRU for the day (kg P/ha)*
- real \*8, dimension(:), allocatable [gdrain](#)  
*drain tile lag time: the amount of time between the transfer of water from the soil to the drain tile and the release of the water from the drain tile to the reach (hours)*
- real \*8, dimension(:), allocatable [ddrain](#)  
*depth of drain tube from the soil surface (mm)*
- real \*8, dimension(:), allocatable [sol\\_crk](#)  
*crack volume potential of soil (none)*
- real \*8, dimension(:), allocatable [brt](#)  
*fraction of surface runoff within the subbasin which takes 1 day or less to reach the subbasin outlet (none)*
- real \*8, dimension(:), allocatable [dayl](#)  
*length of the current day (hours)*
- real \*8, dimension(:), allocatable [sstmaxd](#)  
*static maximum depressional storage; read from .sdr (mm)*
- real \*8, dimension(:), allocatable [re](#)  
*effective radius of drains (mm)*
- real \*8, dimension(:), allocatable [sdrain](#)  
*distance between two drain tubes or tiles (mm)*
- real \*8, dimension(:), allocatable [ddrain\\_hru](#)
- real \*8, dimension(:), allocatable [drain\\_co](#)  
*drainage coefficient (mm/day)*
- real \*8, dimension(:), allocatable [latksatf](#)  
*multiplication factor to determine conk(j1,j) from sol\_k(j1,j) for HRU (none)*
- real \*8, dimension(:), allocatable [pc](#)  
*pump capacity (default pump capacity = 1.042mm/hr or 25mm/day) (mm/hr)*
- real \*8, dimension(:), allocatable [stmaxd](#)  
*maximum surface depressional storage for day in a given HRU (mm)*
- real \*8, dimension(:), allocatable [rnd3](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [rnd2](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [twash](#)
- real \*8, dimension(:), allocatable [sol\\_cnsu](#)
- real \*8, dimension(:), allocatable [doxq](#)
- real \*8, dimension(:), allocatable [rnd8](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [rnd9](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [percn](#)  
*amount of nitrate percolating past bottom of soil profile (kg N/ha)*

- real \*8, dimension(:), allocatable **sol\_sumwp**
- real \*8, dimension(:), allocatable **qdr**  
*total amount of water entering main channel for day from HRU (mm H2O)*
- real \*8, dimension(:), allocatable **tauton**  
*amount of N applied in autofert operation in year (kg N/ha)*
- real \*8, dimension(:), allocatable **tautop**  
*amount of P applied in autofert operation in year (kg N/ha)*
- real \*8, dimension(:), allocatable **cbodu**
- real \*8, dimension(:), allocatable **chl\_a**
- real \*8, dimension(:), allocatable **tfertn**
- real \*8, dimension(:), allocatable **tfertp**
- real \*8, dimension(:), allocatable **tgrazn**
- real \*8, dimension(:), allocatable **tgrazp**
- real \*8, dimension(:), allocatable **latq**  
*total lateral flow in soil profile for the day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable **npInt**  
*plant uptake of nitrogen in HRU for the day (kg N/ha)*
- real \*8, dimension(:), allocatable **latno3**  
*amount of nitrate transported with lateral flow (kg N/ha)*
- real \*8, dimension(:), allocatable **minpgw**
- real \*8, dimension(:), allocatable **no3gw**
- real \*8, dimension(:), allocatable **tileq**
- real \*8, dimension(:), allocatable **tileno3**
- real \*8, dimension(:), allocatable **sedorgn**  
*amount of organic nitrogen in surface runoff in HRU for the day (kg N/ha)*
- real \*8, dimension(:), allocatable **sedminpa**  
*amount of active mineral phosphorus sorbed to sediment in surface runoff in HRU for day (kg P/ha)*
- real \*8, dimension(:), allocatable **sedminps**  
*amount of stable mineral phosphorus sorbed to sediment in surface runoff in HRU for day (kg P/ha)*
- real \*8, dimension(:), allocatable **sedyld**  
*soil loss caused by water erosion for day in HRU (metric tons)*
- real \*8, dimension(:), allocatable **sepbtm**  
*percolation from bottom of soil profile for the day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable **strsn**  
*fraction of potential plant growth achieved on the day where the reduction is caused by nitrogen stress (none)*
- real \*8, dimension(:), allocatable **sedorgp**  
*amount of organic phosphorus in surface runoff in HRU for the day (kg P/ha)*
- real \*8, dimension(:), allocatable **surfq**  
*surface runoff generated in HRU on the current day (mm H2O)*
- real \*8, dimension(:), allocatable **strstmp**  
*fraction of potential plant growth achieved on the day in HRU where the reduction is caused by temperature stress (none)*
- real \*8, dimension(:), allocatable **strsp**  
*fraction of potential plant growth achieved on the day where the reduction is caused by phosphorus stress (none)*
- real \*8, dimension(:), allocatable **surqno3**  
*amount of nitrate transported with surface runoff (kg N/ha)*
- real \*8, dimension(:), allocatable **hru\_ha**  
*area of HRU in hectares (ha)*
- real \*8, dimension(:), allocatable **hru\_dafr**  
*fraction of total watershed area contained in HRU (km2/km2)*
- real \*8, dimension(:), allocatable **tcfrtn**
- real \*8, dimension(:), allocatable **tcfrtp**

- real \*8, dimension(:), allocatable [drydep\\_no3](#)  
*atmospheric dry deposition of nitrates (kg/ha/yr)*
- real \*8, dimension(:), allocatable [drydep\\_nh4](#)  
*atmospheric dry deposition of ammonia (kg/ha/yr)*
- real \*8, dimension(:), allocatable [bio\\_yrms](#)  
*annual biomass (dry weight) in the HRU (metric tons/ha)*
- real \*8, dimension(:), allocatable [phubase](#)  
*base zero total heat units (used when no land cover is growing) (heat units)*
- real \*8, dimension(:), allocatable [hvstiadj](#)  
*optimal harvest index adjusted for water stress for current time during growing season ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable [laiday](#)  
*leaf area index for HRU ( $m^2/m^2$ )*
- real \*8, dimension(:), allocatable [chlap](#)  
*chlorophyll-a production coefficient for pond (none)*
- real \*8, dimension(:), allocatable [pnd\\_psed](#)  
*amount of mineral P attached to sediment originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [laimxfr](#)
- real \*8, dimension(:), allocatable [seccip](#)  
*water clarity coefficient for pond (none)*
- real \*8, dimension(:), allocatable [plantn](#)  
*amount of nitrogen in plant biomass (kg N/ha)*
- real \*8, dimension(:), allocatable [plt\\_et](#)  
*actual ET simulated during life of plant (mm H2O)*
- real \*8, dimension(:), allocatable [wet\\_psed](#)  
*amount of mineral P attached to sediment originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [bio\\_aams](#)  
*average annual biomass (dry weight) in the HRU (metric tons)*
- real \*8, dimension(:), allocatable [plantp](#)  
*amount of phosphorus stored in plant biomass (kg P/ha)*
- real \*8, dimension(:), allocatable [plt\\_pet](#)  
*potential ET simulated during life of plant (mm H2O)*
- real \*8, dimension(:), allocatable [dormhr](#)  
*time threshold used to define dormant period for plant (when daylength is within the time specified by dl from the minimum daylength for the area, the plant will go dormant) (hour)*
- real \*8, dimension(:), allocatable [lai\\_yrmx](#)  
*maximum leaf area index for the year in the HRU (none)*
- real \*8, dimension(:), allocatable [bio\\_aamx](#)
- real \*8, dimension(:), allocatable [lat\\_pst](#)  
*amount of pesticide in lateral flow in HRU for the day (kg pst/ha)*
- real \*8, dimension(:), allocatable [fld\\_fr](#)  
*fraction of HRU area that drains into floodplain ( $km^2/km^2$ )*
- real \*8, dimension(:), allocatable [orig\\_snohru](#)
- real \*8, dimension(:), allocatable [orig\\_potvol](#)
- real \*8, dimension(:), allocatable [pltfr\\_n](#)  
*fraction of plant biomass that is nitrogen (none)*
- real \*8, dimension(:), allocatable [orig\\_alai](#)
- real \*8, dimension(:), allocatable [orig\\_bioms](#)
- real \*8, dimension(:), allocatable [pltfr\\_p](#)  
*fraction of plant biomass that is phosphorus (none)*
- real \*8, dimension(:), allocatable [orig\\_phuacc](#)
- real \*8, dimension(:), allocatable [orig\\_sumix](#)

- real \*8, dimension(:), allocatable [phu\\_plt](#)  
*total number of heat units to bring plant to maturity (heat units)*
- real \*8, dimension(:), allocatable **orig\_phu**
- real \*8, dimension(:), allocatable **orig\_shallst**
- real \*8, dimension(:), allocatable **orig\_deepst**
- real \*8, dimension(:), allocatable [rip\\_fr](#)  
*fraction of HRU area that drains into riparian zone ( $\text{km}^2/\text{km}^2$ )*
- real \*8, dimension(:), allocatable **orig\_pndvol**
- real \*8, dimension(:), allocatable **orig\_pndsed**
- real \*8, dimension(:), allocatable **orig\_pndno3**
- real \*8, dimension(:), allocatable **orig\_pndsolp**
- real \*8, dimension(:), allocatable **orig\_pndorgn**
- real \*8, dimension(:), allocatable **orig\_pndorgp**
- real \*8, dimension(:), allocatable **orig\_wetvol**
- real \*8, dimension(:), allocatable **orig\_wetsed**
- real \*8, dimension(:), allocatable **orig\_wetno3**
- real \*8, dimension(:), allocatable **orig\_wetsolp**
- real \*8, dimension(:), allocatable **orig\_wetorgn**
- real \*8, dimension(:), allocatable **orig\_wetorgp**
- real \*8, dimension(:), allocatable **orig\_solcov**
- real \*8, dimension(:), allocatable **orig\_solsw**
- real \*8, dimension(:), allocatable **orig\_potno3**
- real \*8, dimension(:), allocatable **orig\_potsed**
- real \*8, dimension(:), allocatable [wtab](#)  
*water table based on 30 day antecedent climate (precip,et) (mm)*
- real \*8, dimension(:), allocatable **wtab\_mn**
- real \*8, dimension(:), allocatable **wtab\_mx**
- real \*8, dimension(:), allocatable [shallst\\_n](#)  
*nitrate concentration in shallow aquifer converted to kg/ha (ppm NO3-N)*
- real \*8, dimension(:), allocatable **gw\_nloss**
- real \*8, dimension(:), allocatable **rchrg\_n**
- real \*8, dimension(:), allocatable **det\_san**
- real \*8, dimension(:), allocatable **det\_sil**
- real \*8, dimension(:), allocatable **det\_cla**
- real \*8, dimension(:), allocatable **det\_sag**
- real \*8, dimension(:), allocatable **det\_lag**
- real \*8, dimension(:), allocatable [afrt\\_surface](#)  
*fraction of fertilizer which is applied to top 10 mm of soil (the remaining fraction is applied to first soil layer) (none)*
- real \*8, dimension(:), allocatable [tnylida](#)  
*estimated/target nitrogen content of yield used in autofertilization (kg N/kg yield)*
- real \*8 [firt\\_surface](#)  
*fraction of fertilizer which is applied to the top 10 mm of soil (the remaining fraction is applied to the first soil layer) (none)*
- real \*8, dimension(:), allocatable [auto\\_nyr](#)  
*maximum NO3-N content allowed to be applied in one year by auto-fertilization (kg NO3-N/ha)*
- real \*8, dimension(:), allocatable [auto\\_napp](#)  
*maximum NO3-N content allowed in one fertilizer application (kg NO3-N/ha)*
- real \*8, dimension(:), allocatable [auto\\_nstrs](#)  
*nitrogen stress factor which triggers auto fertilization (none)*
- real \*8, dimension(:), allocatable [manure\\_kg](#)  
*dry weight of manure deposited on HRU daily ((kg/ha)/day)*
- real \*8, dimension(:,:), allocatable **rcn\_mo**
- real \*8, dimension(:,:), allocatable **rammo\_mo**

- real \*8, dimension(:,:), allocatable **drydep\_no3\_mo**
- real \*8, dimension(:,:), allocatable **drydep\_nh4\_mo**
- real \*8, dimension(:), allocatable **rcn\_d**
- real \*8, dimension(:), allocatable **rammo\_d**
- real \*8, dimension(:), allocatable **drydep\_no3\_d**
- real \*8, dimension(:), allocatable **drydep\_nh4\_d**
- real \*8, dimension(:,:), allocatable **yldn**
- integer, dimension(:,:), allocatable **gwati**
- real \*8, dimension(:,:), allocatable **gwatn**
- real \*8, dimension(:,:), allocatable **gwatl**
- real \*8, dimension(:,:), allocatable **gwatw**
- real \*8, dimension(:,:), allocatable **gwatd**
- real \*8, dimension(:,:), allocatable **gwatveg**
- real \*8, dimension(:,:), allocatable **gwata**
- real \*8, dimension(:,:), allocatable **gwats**
- real \*8, dimension(:,:), allocatable **gwatspcon**
- real \*8, dimension(:,:), allocatable **rfqeo\_30d**
- real \*8, dimension(:,:), allocatable **eo\_30d**
- real \*8, dimension(:), allocatable **psetlp1**  
*phosphorus settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **psetlp2**  
*phosphorus settling rate for 2nd season (m/day)*
- real \*8, dimension(:,:), allocatable **wgnold**  
*previous value of wgncur(:,) (none)*
- real \*8, dimension(:,:), allocatable **wgncur**  
*parameter to predict the impact of precip on other weather attributes (none)*  
*wgncur(1,:) parameter which predicts impact of precip on daily maximum air temperature*  
*wgncur(2,:) parameter which predicts impact of precip on daily minimum air temperature*  
*wgncur(3,:) parameter which predicts impact of precip on daily solar radiation*
- real \*8, dimension(:), allocatable **wrt1**  
*1st shape parameter for calculation of water retention (none)*
- real \*8, dimension(:), allocatable **wrt2**  
*2nd shape parameter for calculation of water retention (none)*
- real \*8, dimension(:,:), allocatable **pst\_enr**  
*pesticide enrichment ratio (none)*
- real \*8, dimension(:,:), allocatable **pst\_surq**  
*amount of pesticide type lost in surface runoff on current day in HRU (kg/ha)*
- real \*8, dimension(:,:), allocatable **zdb**  
*division term from net pesticide equation (mm)*
- real \*8, dimension(:,:), allocatable **plt\_pst**  
*pesticide on plant foliage (kg/ha)*
- real \*8, dimension(:), allocatable **psetlw1**  
*phosphorus settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **psetlw2**  
*phosphorus settling rate for 2nd season (m/day)*
- real \*8, dimension(:,:), allocatable **pst\_sed**  
*pesticide loading from HRU sorbed onto sediment (kg/ha)*
- real \*8, dimension(:,:), allocatable **wupnd**  
*average daily water removal from the pond for the month ( $10^4 \text{ m}^3/\text{day}$ )*
- real \*8, dimension(:,:), allocatable **phi**

- phi(1,:)* cross-sectional area of flow at bankfull depth ( $m^2$ ) *phi(2,:)* (none) *phi(3,:)* (none) *phi(4,:)* (none) *phi(5,:)* (none) *phi(6,:)* bottom width of main channel (m) *phi(7,:)* depth of water when reach is at bankfull depth (m) *phi(8,:)* average velocity when reach is at bankfull depth (m/s) *phi(9,:)* wave celerity when reach is at bankfull depth (m/s) *phi(10,:)* storage time constant for reach at bankfull depth (ratio of storage to discharge) (hour) *phi(11,:)* average velocity when reach is at 0.1 bankfull depth (low flow) (m/s) *phi(12,:)* wave celerity when reach is at 0.1 bankfull depth (low flow) (m/s) *phi(13,:)* storage time constant for reach at 0.1 bankfull depth (low flow) (ratio of storage to discharge) (hour)
- real \*8, dimension(:,:), allocatable [pcpband](#)  
precipitation for the day in band in HRU (mm H2O)
  - real \*8, dimension(:,:), allocatable [tavband](#)  
average temperature for the day in band in HRU (deg C)
  - real \*8, dimension(:), allocatable [wat\\_phi1](#)  
cross-sectional area of flow at bankfull depth ( $m^2$ )
  - real \*8, dimension(:), allocatable [wat\\_phi5](#)  
flow rate when reach is at bankfull depth ( $m^3/s$ )
  - real \*8, dimension(:), allocatable [wat\\_phi6](#)  
bottom width of main channel (m)
  - real \*8, dimension(:), allocatable [wat\\_phi7](#)  
depth of water when reach is at bankfull (m)
  - real \*8, dimension(:), allocatable [wat\\_phi8](#)  
average velocity when reach is at bankfull depth (m/s)
  - real \*8, dimension(:), allocatable [wat\\_phi9](#)  
wave celerity when reach is at bankfull depth (m/s)
  - real \*8, dimension(:), allocatable [wat\\_phi10](#)  
storage time constant for reach at bankfull depth (ratio of storage to discharge) (hour)
  - real \*8, dimension(:), allocatable [wat\\_phi11](#)  
average velocity when reach is at 0.1 bankfull depth (low flow) (m/s)
  - real \*8, dimension(:), allocatable [wat\\_phi12](#)  
wave celerity when reach is at 0.1 bankfull depth (low flow) (m/s)
  - real \*8, dimension(:), allocatable [wat\\_phi13](#)  
storage time constant for reach at 0.1 bankfull depth (low flow) (ratio of storage to discharge) (hour)
  - real \*8, dimension(:,:), allocatable [snoeb](#)  
snow water content in elevation band on current day (mm H2O)
  - real \*8, dimension(:,:), allocatable [wudeep](#)  
average daily water removal from the deep aquifer for the month ( $10^4 m^3/day$ )
  - real \*8, dimension(:,:), allocatable [wushal](#)  
average daily water removal from the shallow aquifer for the month ( $10^4 m^3/day$ )
  - real \*8, dimension(:,:), allocatable [tmnband](#)  
minimum temperature for the day in band in HRU (deg C)
  - real \*8, dimension(:), allocatable [bss1](#)
  - real \*8, dimension(:), allocatable [bss2](#)
  - real \*8, dimension(:), allocatable [bss3](#)
  - real \*8, dimension(:), allocatable [bss4](#)
  - real \*8, dimension(:), allocatable [nsetlw1](#)  
nitrogen settling rate for 1st season (m/day)
  - real \*8, dimension(:), allocatable [nsetlw2](#)  
nitrogen settling rate for 2nd season (m/day)
  - real \*8, dimension(:,:), allocatable [snotmpeb](#)  
temperature of snow pack in elevation band (deg C)
  - real \*8, dimension(:), allocatable [surf\\_bs1](#)  
amount of surface runoff lagged over one day (mm H2O)
  - real \*8, dimension(:), allocatable [surf\\_bs2](#)



- real \*8, dimension(:), allocatable **surf\_bs3**
- real \*8, dimension(:), allocatable **surf\_bs4**
- real \*8, dimension(:), allocatable **surf\_bs5**
- real \*8, dimension(:), allocatable **surf\_bs6**
- real \*8, dimension(:), allocatable **surf\_bs7**
- real \*8, dimension(:), allocatable **surf\_bs8**
- real \*8, dimension(:), allocatable **surf\_bs9**
- real \*8, dimension(:), allocatable **surf\_bs10**
- real \*8, dimension(:), allocatable **surf\_bs11**
- real \*8, dimension(:), allocatable **surf\_bs12**
- real \*8, dimension(:), allocatable **surf\_bs13**
- real \*8, dimension(:), allocatable **surf\_bs14**
- real \*8, dimension(:), allocatable **surf\_bs15**
- real \*8, dimension(:), allocatable **surf\_bs16**
- real \*8, dimension(:), allocatable **surf\_bs17**
- real \*8, dimension(:), allocatable **nsetlp1**  
*nitrogen settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **nsetlp2**  
*nitrogen settling rate for 2nd season (m/day)*
- real \*8, dimension(:,:), allocatable **tmxband**  
*maximum temperature for the day in band in HRU (deg C)*
- real \*8, dimension(:,:), allocatable **frad**  
*fraction of solar radiation occuring during hour in day in HRU (none)*
- real \*8, dimension(:,:), allocatable **rainsub**  
*precipitation for the time step during the day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable **rstpbsb**
- real \*8, dimension(:,:), allocatable **orig\_snoeb**
- real \*8, dimension(:,:), allocatable **orig\_pltpst**
- real \*8, dimension(:,:), allocatable **terr\_p**
- real \*8, dimension(:,:), allocatable **terr\_cn**
- real \*8, dimension(:,:), allocatable **terr\_sl**
- real \*8, dimension(:,:), allocatable **drain\_d**
- real \*8, dimension(:,:), allocatable **drain\_t**
- real \*8, dimension(:,:), allocatable **drain\_g**
- real \*8, dimension(:,:), allocatable **drain\_idep**
- real \*8, dimension(:,:), allocatable **cont\_cn**
- real \*8, dimension(:,:), allocatable **cont\_p**
- real \*8, dimension(:,:), allocatable **filt\_w**
- real \*8, dimension(:,:), allocatable **strip\_n**
- real \*8, dimension(:,:), allocatable **strip\_cn**
- real \*8, dimension(:,:), allocatable **strip\_c**
- real \*8, dimension(:,:), allocatable **strip\_p**
- real \*8, dimension(:,:), allocatable **fire\_cn**
- real \*8, dimension(:,:), allocatable **cropno\_upd**
- real \*8, dimension(:,:), allocatable **hi\_upd**
- real \*8, dimension(:,:), allocatable **laimx\_upd**
- real \*8, dimension(:,:), allocatable **phug**  
*fraction of plant heat units at which grazing begins (none)*
- real \*8, dimension(:,:), allocatable **pst\_lag**
- integer, dimension(:), allocatable **hrupest**  
*pesticide use flag (none)*  
*0: no pesticides used in HRU*  
*1: pesticides used in HRU*
- integer, dimension(:), allocatable **nrelease**



- sequence number of impound/release operation within the year (none)*
- integer, dimension(:), allocatable [swtrg](#)
  - rainfall event flag (none):*
  - 0: no rainfall event over midnight*
  - 1: rainfall event over midnight*
- integer, dimension(:), allocatable [nrot](#)
  - number of years of rotation (none)*
- integer, dimension(:), allocatable [nfert](#)
  - sequence number of fertilizer application within the year (none)*
- integer, dimension(:), allocatable [nro](#)
  - sequence number of year in rotation (none)*
- integer, dimension(:), allocatable [igro](#)
  - land cover status code (none). This code informs the model whether or not a land cover is growing at the beginning of the simulation*
  - 0 no land cover currently growing*
  - 1 land cover growing*
- integer, dimension(:), allocatable [ipnd1](#)
  - beginning month of nutrient settling season (none)*
- integer, dimension(:), allocatable [ipnd2](#)
  - ending month of nutrient settling season (none)*
- integer, dimension(:), allocatable [nair](#)
  - sequence number of auto-irrigation application within the year (none)*
- integer, dimension(:), allocatable [iflod1](#)
  - beginning month of non-flood season (none)*
- integer, dimension(:), allocatable [iflod2](#)
  - ending month of non-flood season (none)*
- integer, dimension(:), allocatable [ndtarg](#)
  - number of days required to reach target storage from current pond storage (none)*
- integer, dimension(:), allocatable [nirr](#)
  - sequence number of irrigation application within the year (none)*
- integer, dimension(:), allocatable [nstress](#)
  - code for approach used to determine amount of nitrogen to HRU (none):*
  - 0 nitrogen target approach*
  - 1 annual max approach*
- integer, dimension(:), allocatable [iaftrtyp](#)
- integer, dimension(:), allocatable [igrotree](#)
- integer, dimension(:), allocatable [grz\\_days](#)
  - number of days grazing will be simulated (none)*
- integer, dimension(:), allocatable [nmgt](#)
  - management code (for GIS output only) (none)*
- integer, dimension(:), allocatable [nafert](#)
  - sequence number of auto-fert application within the year (none)*
- integer, dimension(:), allocatable [nsweep](#)
  - sequence number of street sweeping operation within the year (none)*
- integer, dimension(:), allocatable [icr](#)
  - sequence number of crop grown within the current year (none)*
- integer, dimension(:), allocatable [ncut](#)
  - sequence number of harvest operation within a year (none)*
- integer, dimension(:), allocatable [irrho](#)

- irrigation source location (none)*
- if IRRSC=1, IRRNO is the number of the reach*
- if IRRSC=2, IRRNO is the number of the reservoir*
- if IRRSC=3, IRRNO is the number of the subbasin*
- if IRRSC=4, IRRNO is the number of the subbasin*
- if IRRSC=5, not used*
- integer, dimension(:), allocatable **sol\_nly**  
*number of soil layers in HRU (none)*
- integer, dimension(:), allocatable **npcp**  
*prior day category (none)*  
*1 dry day*  
*2 wet day*
- integer, dimension(:), allocatable **irn**  
*average annual number of irrigation applications in HRU (none)*
- integer, dimension(:), allocatable **ncf**  
*sequence number of continuous fertilization operation within the year (none)*
- integer, dimension(:), allocatable **ngr**  
*sequence number of grazing operation within the year (none)*
- integer, dimension(:), allocatable **igrz**  
*grazing flag for HRU (none):*  
*0 HRU currently not grazed*  
*1 HRU currently grazed*
- integer, dimension(:), allocatable **ndeat**  
*number of days HRU has been grazed (days)*
- integer, dimension(:), allocatable **hru\_sub**  
*subbasin in which HRU is located (none)*
- integer, dimension(:), allocatable **urblu**  
*urban land type identification number from urban.dat (none)*
- integer, dimension(:), allocatable **ldrain**  
*soil layer where drainage tile is located (none)*
- integer, dimension(:), allocatable **idorm**  
*dormancy status code (none):*  
*0 land cover growing (not dormant)*  
*1 land cover dormant*
- integer, dimension(:), allocatable **hru\_seq**
- integer, dimension(:), allocatable **iurban**  
*urban simulation code (none):*  
*0 no urban sections in HRU*  
*1 urban sections in HRU, simulate using USGS regression equations*  
*2 urban sections in HRU, simulate using build up/wash off algorithm*
- integer, dimension(:), allocatable **icftr**  
*continuous fertilizer flag for HRU (none):*  
*0 HRU currently not continuously fertilized*  
*1 HRU currently continuously fertilized*
- integer, dimension(:), allocatable **iday\_fert**
- integer, dimension(:), allocatable **ifld**  
*number of HRU (in subbasin) that is a floodplain (none)*
- integer, dimension(:), allocatable **irip**  
*number of HRU (in subbasin) that is a riparian zone (none)*
- integer, dimension(:), allocatable **hrgis**  
*GIS code printed to output files (output.hru, .rch) (none)*
- integer, dimension(:), allocatable **ndcftr**  
*number of days HRU has been continuously fertilized (days)*
- integer, dimension(:), allocatable **irrsp**

- irrigation source code (none):*
  - 1 divert water from reach
  - 2 divert water from reservoir
  - 3 divert water from shallow aquifer
  - 4 divert water from deep aquifer
  - 5 divert water from source outside watershed
- integer, dimension(:), allocatable **ntil**
  - sequence number of tillage operation within current year (none)*
- integer, dimension(:), allocatable **orig\_igro**
- integer, dimension(:), allocatable **iwatable**
  - high water table code (none):*
    - 0 no high water table
    - 1 high water table
- integer, dimension(:), allocatable **curyr\_mat**
- integer, dimension(:), allocatable **icpst**
  - icpst = 0 do not apply*
  - icpst = 1 application period*
- integer, dimension(:), allocatable **ndcpst**
  - current day within the application period (day)*
- integer, dimension(:), allocatable **ncpest**
- integer, dimension(:), allocatable **iday\_pest**
  - current day between applications (day)*
- integer, dimension(:), allocatable **irr\_flag**
- integer, dimension(:), allocatable **irra\_flag**
- integer, dimension(:,:), allocatable **rndseed**
  - random number generator seeds array. The seeds in the array are used to generate random numbers for the following purposes (none):*
    - (1) wet/dry day probability
    - (2) solar radiation
    - (3) precipitation
    - (4) USLE rainfall erosion index
    - (5) wind speed
    - (6) 0.5 hr rainfall fraction
    - (7) relative humidity
    - (8) maximum temperature
    - (9) minimum temperature
    - (10) generate new random numbers
- integer, dimension(:,:), allocatable **iterr**
- integer, dimension(:,:), allocatable **iyterr**
- integer, dimension(:,:), allocatable **itdrain**
- integer, dimension(:,:), allocatable **iydrain**
- integer, dimension(:,:), allocatable **ncrops**
- integer, dimension(:), allocatable **manure\_id**
  - manure (fertilizer) identification number from fert.dat (none)*
- integer, dimension(:,:), allocatable **mgt\_sdr**
- integer, dimension(:,:), allocatable **idplot**
- integer, dimension(:,:), allocatable **icont**
- integer, dimension(:,:), allocatable **iycont**
- integer, dimension(:,:), allocatable **ifilt**
- integer, dimension(:,:), allocatable **iyfilt**
- integer, dimension(:,:), allocatable **istrip**
- integer, dimension(:,:), allocatable **iystrip**
- integer, dimension(:,:), allocatable **iopday**
- integer, dimension(:,:), allocatable **iopyr**
- integer, dimension(:,:), allocatable **mgt\_ops**
- real \*8, dimension(:), allocatable **wshd\_pstap**

- total amount of pesticide type applied in watershed during simulation (kg/ha)*

  - real \*8, dimension(:), allocatable [wshd\\_pstdg](#)
- amount of pesticide lost through degradation in watershed (kg pst/ha)*

  - integer, dimension(12) **ndmo**
- integer, dimension(:), allocatable* [npno](#)

*array of unique pesticides used in watershed (none)*

  - integer, dimension(:), allocatable **mcrhru**
- character(len=13), dimension(18) [rfile](#)

*rainfall file names (.pcp)*

  - character(len=13), dimension(18) [tfile](#)
- temperature file names (.tmp)*

  - character(len=4), dimension(1000) [urbname](#)
- name of urban land use*

  - character(len=1), dimension(:), allocatable [kirr](#)
- irrigation in HRU*

  - character(len=1), dimension(:), allocatable **hydgrp**
- character(len=16), dimension(:), allocatable [snam](#)

*soil series name*

  - character(len=17), dimension(300) [pname](#)
- name of pesticide/toxin*

  - character(len=4), dimension(60) [title](#)
- description lines in file.cio (1st 3 lines)*

  - character(len=4), dimension(5000) [cpnm](#)
- four character code to represent crop name*

  - character(len=17), dimension(50) **fname**
- real \*8, dimension(:, :, :), allocatable [flomon](#)

*average daily water loading for month ( $m^3/day$ )*

  - real \*8, dimension(:, :, :), allocatable [solpstmon](#)
- average daily soluble pesticide loading for month (mg pst/day)*

  - real \*8, dimension(:, :, :), allocatable [srbpstmon](#)
- average daily sorbed pesticide loading for month (mg pst/day)*

  - real \*8, dimension(:, :, :), allocatable [orgnmon](#)
- average daily organic N loading for month (kg N/day)*

  - real \*8, dimension(:, :, :), allocatable [orgpmon](#)
- average daily organic P loading for month (kg P/day)*

  - real \*8, dimension(:, :, :), allocatable [sedmon](#)
- average daily sediment loading for month (metric tons/day)*

  - real \*8, dimension(:, :, :), allocatable [minpmon](#)
- average daily mineral P loading for month (kg P/day)*

  - real \*8, dimension(:, :, :), allocatable [nh3mon](#)
- average amount of NH3-N loaded to stream on a given day in the month (kg N/day)*

  - real \*8, dimension(:, :, :), allocatable [no3mon](#)
- average daily NO3-N loading for month (kg N/day)*

  - real \*8, dimension(:, :, :), allocatable [bactlpmon](#)
- average amount of less persistent bacteria loaded to stream on a given day in the month (# bact/day)*

  - real \*8, dimension(:, :, :), allocatable [bactpmon](#)
- average amount of persistent bacteria loaded to stream on a given day in the month (# bact/day)*

  - real \*8, dimension(:, :, :), allocatable [no2mon](#)
- average amount of NO2-N loaded to stream on a given day in the month (kg N/day)*

  - real \*8, dimension(:, :, :), allocatable [cmtl1mon](#)
- average amount of conservative metal #1 loaded to stream on a given day in the month (# bact/day)*

- real \*8, dimension(:,:), allocatable [cmtl2mon](#)  
*average amount of conservative metal #2 loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:,:), allocatable [cmtl3mon](#)  
*average amount of conservative metal #3 loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:,:), allocatable [cbodmon](#)  
*average daily loading of CBOD in month (kg/day)*
- real \*8, dimension(:,:), allocatable [chlamon](#)  
*average daily loading of chlorophyll-a in month (kg/day)*
- real \*8, dimension(:,:), allocatable [disoxmon](#)  
*average daily loading of dissolved O2 in month (kg/day)*
- real \*8, dimension(:,:), allocatable [floyr](#)  
*average daily water loading for year ( $m^3$ /day)*
- real \*8, dimension(:,:), allocatable [orgnyr](#)  
*average daily organic N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [orgpyr](#)  
*average daily organic P loading for year (kg P/day)*
- real \*8, dimension(:,:), allocatable [sedyr](#)  
*average daily sediment loading for year (metric tons/day)*
- real \*8, dimension(:,:), allocatable [minpyr](#)  
*average daily mineral P loading for year (kg P/day)*
- real \*8, dimension(:,:), allocatable [nh3yr](#)  
*average daily NH3-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [no2yr](#)  
*average daily NO2-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [no3yr](#)  
*average daily NO3-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [bactlpyr](#)  
*average daily loading of less persistent bacteria for year (# bact/day)*
- real \*8, dimension(:,:), allocatable [bactpyr](#)  
*average daily loading of persistent bacteria for year (# bact/day)*
- real \*8, dimension(:,:), allocatable [cmtl1yr](#)  
*average daily loading of conservative metal #1 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [chlayr](#)  
*average daily loading of chlorophyll-a in year (kg/day)*
- real \*8, dimension(:,:), allocatable [cmtl2yr](#)  
*average daily loading of conservative metal #2 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [cmtl3yr](#)  
*average daily loading of conservative metal #3 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [cbodyr](#)  
*average daily loading of CBOD in year (kg/day)*
- real \*8, dimension(:,:), allocatable [disoxyr](#)  
*average daily loading of dissolved O2 in year (kg/day)*
- real \*8, dimension(:,:), allocatable [solpstyr](#)  
*average daily soluble pesticide loading for year (mg pst/day)*
- real \*8, dimension(:,:), allocatable [srbspstyr](#)  
*average daily sorbed pesticide loading for year (mg pst/day)*
- real \*8, dimension(:,:), allocatable [sol\\_mc](#)
- real \*8, dimension(:,:), allocatable [sol\\_mn](#)
- real \*8, dimension(:,:), allocatable [sol\\_mp](#)
- real \*8, dimension(:), allocatable [flocnst](#)
- real \*8, dimension(:), allocatable [orgncnst](#)



- real \*8, dimension(:), allocatable **hsedyld**
- real \*8, dimension(:), allocatable **hsedst**
- real \*8, dimension(:), allocatable **hharea**
- real \*8, dimension(:), allocatable **hsolpst**
- real \*8, dimension(:), allocatable **hsorpst**
- real \*8, dimension(:), allocatable **hhqday**  
*surface runoff generated each timestep of day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable **precipdt**  
*precipitation, or effective precipitation reaching soil surface, in time step for HRU (mm H2O)*
- real \*8, dimension(:), allocatable **hhtime**
- real \*8, dimension(:), allocatable **hbactp**
- real \*8, dimension(:), allocatable **hbactlp**
- integer, dimension(10) **ivar\_orig**
- real \*8, dimension(10) **rvar\_orig**
- integer **nsave**  
*number of save commands in .fig file*
- integer **nauto**
- integer **iatmodep**
- real \*8, dimension(:), allocatable **wattemp**
- real \*8, dimension(:), allocatable **lkpst\_mass**
- real \*8, dimension(:), allocatable **lkspst\_mass**
- real \*8, dimension(:), allocatable **vel\_chan**
- real \*8, dimension(:), allocatable **vfscn**  
*fraction of the total runoff from the entire field entering the most concentrated 10% of the VFS (none)*
- real \*8, dimension(:), allocatable **vfstratio**  
*field area/VFS area ratio (none)*
- real \*8, dimension(:), allocatable **vfsch**  
*fraction of flow entering the most concentrated 10% of the VFS which is fully channelized (none)*
- real \*8, dimension(:), allocatable **vfsi**
- real \*8, dimension(:,:), allocatable **filter\_i**
- real \*8, dimension(:,:), allocatable **filter\_ratio**
- real \*8, dimension(:,:), allocatable **filter\_con**
- real \*8, dimension(:,:), allocatable **filter\_ch**
- real \*8, dimension(:,:), allocatable **sol\_n**
- integer **cswat**  
*= 0 Static soil carbon (old mineralization routines)*  
*= 1 C-FARM one carbon pool model*  
*= 2 Century model*
- real \*8, dimension(:,:), allocatable **sol\_bdp**
- real \*8, dimension(:,:), allocatable **tillagef**
- real \*8, dimension(:), allocatable **rtfr**
- real \*8, dimension(:), allocatable **stsol\_rd**  
*storing last soil root depth for use in harvestkilop/kilop (mm)*
- integer **urban\_flag**
- integer **dorm\_flag**
- real \*8 **bf\_flg**
- real \*8 **iabstr**
- real \*8, dimension(:), allocatable **ubnrunoff**
- real \*8, dimension(:), allocatable **ubntss**
- real \*8, dimension(:,:), allocatable **sub\_ubnrunoff**
- real \*8, dimension(:,:), allocatable **sub\_ubntss**
- real \*8, dimension(:,:), allocatable **ovrlnd\_dt**
- real \*8, dimension(:,:), allocatable **hhsurf\_bs1**
- real \*8, dimension(:,:), allocatable **hhsurf\_bs2**

- integer **iuh**  
*unit hydrograph method: 1=triangular UH; 2=gamma funtion UH;*
- integer **sed\_ch**  
*channel routing for HOURLY; 0=Bagnold; 2=Brownlie; 3=Yang;*
- real \*8 **eros\_expo**  
*an exponent in the overland flow erosion equation ranges 1.5-3.0*
- real \*8 **eros\_spl**  
*coefficient of splash erosion varing 0.9-3.1*
- real \*8 **rill\_mult**  
*Multiplier to USLE\_K for soil susceptible to rill erosion, range 0.5-2.0.*
- real \*8 **sedprev**
- real \*8 **c\_factor**
- real \*8 **ch\_d50**  
*median particle diameter of channel bed (mm)*
- real \*8 **sig\_g**  
*geometric standard deviation of particle sizes for the main channel. Mean air temperature at which precipitation is equally likely to be rain as snow/freezing rain.*
- real \*8 **uhalpha**  
*alpha coefficient for estimating unit hydrograph using a gamma function (\*.bsn)*
- real \*8 **abstinit**
- real \*8 **abstmax**
- real \*8, dimension(:,:), allocatable **hhsedy**  
*sediment yield from HRU drung a time step applied to HRU (tons)*
- real \*8, dimension(:,:), allocatable **sub\_subp\_dt**
- real \*8, dimension(:,:), allocatable **sub\_hhsedy**
- real \*8, dimension(:,:), allocatable **sub\_atmp**
- real \*8, dimension(:), allocatable **rhy**
- real \*8, dimension(:), allocatable **init\_abstrc**
- real \*8, dimension(:), allocatable **dratio**
- real \*8, dimension(:), allocatable **hrtevp**
- real \*8, dimension(:), allocatable **hrttlc**
- real \*8, dimension(:,:), allocatable **rchhr**
- real \*8, dimension(:), allocatable **hhresflwi**
- real \*8, dimension(:), allocatable **hhresflwo**
- real \*8, dimension(:), allocatable **hhressedi**
- real \*8, dimension(:), allocatable **hhressedo**
- character(len=4), dimension(:), allocatable **lu\_nodrain**
- integer, dimension(:), allocatable **bmpdrain**
- real \*8, dimension(:), allocatable **sub\_cn2**
- real \*8, dimension(:), allocatable **sub\_ha\_urb**
- real \*8, dimension(:), allocatable **bmp\_recharge**
- real \*8, dimension(:), allocatable **sub\_ha\_imp**
- real \*8, dimension(:), allocatable **subdr\_km**
- real \*8, dimension(:), allocatable **subdr\_ickm**
- real \*8, dimension(:,:), allocatable **sf\_im**
- real \*8, dimension(:,:), allocatable **sf\_iy**
- real \*8, dimension(:,:), allocatable **sp\_sa**
- real \*8, dimension(:,:), allocatable **sp\_pvol**
- real \*8, dimension(:,:), allocatable **sp\_pd**
- real \*8, dimension(:,:), allocatable **sp\_sedi**
- real \*8, dimension(:,:), allocatable **sp\_sede**
- real \*8, dimension(:,:), allocatable **ft\_sa**
- real \*8, dimension(:,:), allocatable **ft\_fsa**



- real \*8, dimension(:,:), allocatable **ft\_dep**
- real \*8, dimension(:,:), allocatable **ft\_h**
- real \*8, dimension(:,:), allocatable **ft\_pd**
- real \*8, dimension(:,:), allocatable **ft\_k**
- real \*8, dimension(:,:), allocatable **ft\_dp**
- real \*8, dimension(:,:), allocatable **ft\_dc**
- real \*8, dimension(:,:), allocatable **ft\_por**
- real \*8, dimension(:,:), allocatable **tss\_den**
- real \*8, dimension(:,:), allocatable **ft\_alp**
- real \*8, dimension(:,:), allocatable **sf\_fr**
- real \*8, dimension(:,:), allocatable **sp\_qi**
- real \*8, dimension(:,:), allocatable **sp\_k**
- real \*8, dimension(:,:), allocatable **ft\_qpnd**
- real \*8, dimension(:,:), allocatable **sp\_dp**
- real \*8, dimension(:,:), allocatable **ft\_qsw**
- real \*8, dimension(:,:), allocatable **ft\_qin**
- real \*8, dimension(:,:), allocatable **ft\_qout**
- real \*8, dimension(:,:), allocatable **ft\_sedpnd**
- real \*8, dimension(:,:), allocatable **sp\_bpw**
- real \*8, dimension(:,:), allocatable **ft\_bpw**
- real \*8, dimension(:,:), allocatable **ft\_sed\_cumul**
- real \*8, dimension(:,:), allocatable **sp\_sed\_cumul**
- integer, dimension(:), allocatable **num\_sf**
- integer, dimension(:,:), allocatable **sf\_typ**
- integer, dimension(:,:), allocatable **sf\_dim**
- integer, dimension(:,:), allocatable **ft\_qfg**
- integer, dimension(:,:), allocatable **sp\_qfg**
- integer, dimension(:,:), allocatable **sf\_ptp**
- integer, dimension(:,:), allocatable **ft\_fc**
- real \*8 **sfsedmean**
- real \*8 **sfsedstdev**
- integer, dimension(:), allocatable **ntp\_imo**  
*month the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **ntp\_iyr**  
*year of the simulation that the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **ntp\_numstage**  
*total number of stages in the weir (none)*
- integer, dimension(:), allocatable **ntp\_numweir**  
*total number of weirs in the BMP (none)*
- integer, dimension(:), allocatable **ntp\_onoff**  
*sub-basin detention pond is associated with (none)*
- integer, dimension(:), allocatable **ntp\_reltype**  
*equations for stage-discharge relationship (none):*  
*1=exponential function,*  
*2=linear,*  
*3=logarithmic,*  
*4=cubic,*  
*5=power*
- integer, dimension(:), allocatable **ntp\_stagdis**  
*(none):*  
*0=use weir/orifice discharge equation to calculate outflow,*  
*1=use stage-dicharge relationship*
- integer, dimension(:), allocatable **ntp\_subnum**
- real \*8, dimension(:), allocatable **cf**

- this parameter controls the response of decomposition to the combined effect of soil temperature and moisture.*
- real \*8, dimension(:), allocatable [cfh](#)  
*maximum humification rate*
  - real \*8, dimension(:), allocatable [cfdec](#)  
*the undisturbed soil turnover rate under optimum soil water and temperature. Increasing it will increase carbon and organic N decomp.*
  - real \*8, dimension(:), allocatable [lat\\_orgn](#)
  - real \*8, dimension(:), allocatable [lat\\_orgp](#)
  - integer, dimension(:,:), allocatable [dtp\\_weirdim](#)  
*weir dimensions (none),*  
*1=read user input,*  
*0=use model calculation*
  - integer, dimension(:,:), allocatable [dtp\\_weirtype](#)  
*type of weir (none):*  
*1=rectangular and*  
*2=circular*
  - real \*8, dimension(:), allocatable [dtp\\_coef1](#)  
*coefficient of 3rd degree in the polynomial equation (none)*
  - real \*8, dimension(:), allocatable [dtp\\_coef2](#)  
*coefficient of 2nd degree in the polynomial equation (none)*
  - real \*8, dimension(:), allocatable [dtp\\_coef3](#)  
*coefficient of 1st degree in the polynomial equation (none)*
  - real \*8, dimension(:), allocatable [dtp\\_evrsv](#)  
*detention pond evaporation coefficient (none)*
  - real \*8, dimension(:), allocatable [dtp\\_expont](#)  
*exponent used in the exponential equation (none)*
  - real \*8, dimension(:), allocatable [dtp\\_intcept](#)  
*intercept used in regression equations (none)*
  - real \*8, dimension(:), allocatable [dtp\\_lwratio](#)  
*ratio of length to width of water back up (none)*
  - real \*8, dimension(:), allocatable [dtp\\_totwrwid](#)  
*total constructed width of the detention wall across the creek (m)*
  - real \*8, dimension(:), allocatable [dtp\\_inflvol](#)
  - real \*8, dimension(:), allocatable [dtp\\_wdep](#)
  - real \*8, dimension(:), allocatable [dtp\\_totdep](#)
  - real \*8, dimension(:), allocatable [dtp\\_watdepact](#)
  - real \*8, dimension(:), allocatable [dtp\\_outflow](#)
  - real \*8, dimension(:), allocatable [dtp\\_totrel](#)
  - real \*8, dimension(:), allocatable [dtp\\_backoff](#)
  - real \*8, dimension(:), allocatable [dtp\\_seep\\_sa](#)
  - real \*8, dimension(:), allocatable [dtp\\_evap\\_sa](#)
  - real \*8, dimension(:), allocatable [dtp\\_pet\\_day](#)
  - real \*8, dimension(:), allocatable [dtp\\_pcpvol](#)
  - real \*8, dimension(:), allocatable [dtp\\_seepvol](#)
  - real \*8, dimension(:), allocatable [dtp\\_evapvol](#)
  - real \*8, dimension(:), allocatable [dtp\\_flowin](#)
  - real \*8, dimension(:), allocatable [dtp\\_backup\\_length](#)
  - real \*8, dimension(:), allocatable [dtp\\_ivol](#)
  - real \*8, dimension(:), allocatable [dtp\\_ised](#)
  - integer, dimension(:,:), allocatable [so\\_res\\_flag](#)
  - integer, dimension(:,:), allocatable [ro\\_bmp\\_flag](#)
  - real \*8, dimension(:,:), allocatable [sol\\_watp](#)
  - real \*8, dimension(:,:), allocatable [sol\\_solp\\_pre](#)

- real \*8, dimension(:,:), allocatable **psp\_store**
- real \*8, dimension(:,:), allocatable **ssp\_store**
- real \*8, dimension(:,:), allocatable **so\_res**
- real \*8, dimension(:,:), allocatable **sol\_cal**
- real \*8, dimension(:,:), allocatable **sol\_ph**
- integer **sol\_p\_model**
- integer, dimension(:,:), allocatable **a\_days**
- integer, dimension(:,:), allocatable **b\_days**
- real \*8, dimension(:), allocatable **min\_res**

*minimum residue allowed due to implementation of residue managment in the OPS file (kg/ha)*

- real \*8, dimension(:), allocatable **harv\_min**
- real \*8, dimension(:), allocatable **fstap**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_flo**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sed**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_bac**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_pp**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sp**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_pn**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sn**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_flos**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_seds**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_bacs**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_pps**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sps**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_pns**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sns**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_flot**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_sedt**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_bact**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_ppt**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_spt**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_pnt**
- real \*8, dimension(:,:), allocatable **ro\_bmp\_snt**
- real \*8, dimension(:), allocatable **bmp\_flo**
- real \*8, dimension(:), allocatable **bmp\_sed**
- real \*8, dimension(:), allocatable **bmp\_bac**
- real \*8, dimension(:), allocatable **bmp\_pp**
- real \*8, dimension(:), allocatable **bmp\_sp**
- real \*8, dimension(:), allocatable **bmp\_pn**
- real \*8, dimension(:), allocatable **bmp\_sn**
- real \*8, dimension(:), allocatable **bmp\_flag**
- real \*8, dimension(:), allocatable **bmp\_flos**
- real \*8, dimension(:), allocatable **bmp\_seds**
- real \*8, dimension(:), allocatable **bmp\_bacs**
- real \*8, dimension(:), allocatable **bmp\_pps**
- real \*8, dimension(:), allocatable **bmp\_sps**
- real \*8, dimension(:), allocatable **bmp\_pns**
- real \*8, dimension(:), allocatable **bmp\_sns**
- real \*8, dimension(:), allocatable **bmp\_flot**
- real \*8, dimension(:), allocatable **bmp\_sedt**
- real \*8, dimension(:), allocatable **bmp\_bact**
- real \*8, dimension(:), allocatable **bmp\_ppt**
- real \*8, dimension(:), allocatable **bmp\_spt**
- real \*8, dimension(:), allocatable **bmp\_pnt**

- real \*8, dimension(:), allocatable **bmp\_snt**
- real \*8, dimension(:,:), allocatable **ntp\_addon**  
*the distance between spillway levels (m)*
- real \*8, dimension(:,:), allocatable **ntp\_cdis**  
*discharge coefficient for weir/orifice flow (none)*
- real \*8, dimension(:,:), allocatable **ntp\_depweir**  
*depth of rectangular weir at different stages (m)*
- real \*8, dimension(:,:), allocatable **ntp\_diaweir**  
*diameter of orifice hole at different stages (m)*
- real \*8, dimension(:,:), allocatable **ntp\_flowrate**  
*maximum discharge from each stage of the weir/hole ( $m^3/s$ )*
- real \*8, dimension(:,:), allocatable **ntp\_pcp**  
*precipitation for different return periods (not used) (mm)*
- real \*8, dimension(:,:), allocatable **ntp\_retperd**  
*return period at different stages (years)*
- real \*8, dimension(:,:), allocatable **ntp\_wdratio**  
*width depth ratio of rectangular weirs (none)*
- real \*8, dimension(:,:), allocatable **ntp\_wrwid**
- real \*8, dimension(:), allocatable **ri\_subkm**
- real \*8, dimension(:), allocatable **ri\_totpv**
- real \*8, dimension(:), allocatable **irmmdt**
- real \*8, dimension(:,:), allocatable **ri\_sed**
- real \*8, dimension(:,:), allocatable **ri\_fr**
- real \*8, dimension(:,:), allocatable **ri\_dim**
- real \*8, dimension(:,:), allocatable **ri\_im**
- real \*8, dimension(:,:), allocatable **ri\_iy**
- real \*8, dimension(:,:), allocatable **ri\_sa**
- real \*8, dimension(:,:), allocatable **ri\_vol**
- real \*8, dimension(:,:), allocatable **ri\_qi**
- real \*8, dimension(:,:), allocatable **ri\_k**
- real \*8, dimension(:,:), allocatable **ri\_dd**
- real \*8, dimension(:,:), allocatable **ri\_evrsv**
- real \*8, dimension(:,:), allocatable **ri\_dep**
- real \*8, dimension(:,:), allocatable **ri\_ndt**
- real \*8, dimension(:,:), allocatable **ri\_pmpvol**
- real \*8, dimension(:,:), allocatable **ri\_sed\_cumul**
- real \*8, dimension(:,:), allocatable **hrnopcp**
- real \*8, dimension(:,:), allocatable **ri\_qloss**
- real \*8, dimension(:,:), allocatable **ri\_pumpv**
- real \*8, dimension(:,:), allocatable **ri\_sedi**
- character(len=4), dimension(:,:), allocatable **ri\_nirr**
- integer, dimension(:), allocatable **num\_ri**
- integer, dimension(:), allocatable **ri\_luflg**
- integer, dimension(:), allocatable **num\_noirr**
- integer, dimension(:), allocatable **wtp\_subnum**
- integer, dimension(:), allocatable **wtp\_onoff**
- integer, dimension(:), allocatable **wtp\_imo**
- integer, dimension(:), allocatable **wtp\_iyr**
- integer, dimension(:), allocatable **wtp\_dim**
- integer, dimension(:), allocatable **wtp\_stagdis**
- integer, dimension(:), allocatable **wtp\_sdtype**
- real \*8, dimension(:), allocatable **wtp\_pvol**
- real \*8, dimension(:), allocatable **wtp\_pdepth**

- real \*8, dimension(:), allocatable **wtp\_sdslope**
- real \*8, dimension(:), allocatable **wtp\_lenwidth**
- real \*8, dimension(:), allocatable **wtp\_extdepth**
- real \*8, dimension(:), allocatable **wtp\_hydeff**
- real \*8, dimension(:), allocatable **wtp\_evrsv**
- real \*8, dimension(:), allocatable **wtp\_sdintc**
- real \*8, dimension(:), allocatable **wtp\_sdexp**
- real \*8, dimension(:), allocatable **wtp\_sdc1**
- real \*8, dimension(:), allocatable **wtp\_sdc2**
- real \*8, dimension(:), allocatable **wtp\_sdc3**
- real \*8, dimension(:), allocatable **wtp\_pdia**
- real \*8, dimension(:), allocatable **wtp\_plen**
- real \*8, dimension(:), allocatable **wtp\_pmann**
- real \*8, dimension(:), allocatable **wtp\_ploss**
- real \*8, dimension(:), allocatable **wtp\_k**
- real \*8, dimension(:), allocatable **wtp\_dp**
- real \*8, dimension(:), allocatable **wtp\_sedi**
- real \*8, dimension(:), allocatable **wtp\_sede**
- real \*8, dimension(:), allocatable **wtp\_qi**
- real \*8 **lai\_init**  
*initial leaf area index of transplants*
- real \*8 **bio\_init**  
*initial biomass of transplants (kg/ha)*
- real \*8 **cnop**  
*SCS runoff curve number for moisture condition II (none)*
- real \*8 **harveff**  
*harvest efficiency: fraction of harvested yield that is removed from HRU; the remainder becomes residue on the soil surface (none)*
- real \*8 **hi\_ovr**  
*harvest index target specified at harvest ((kg/ha)/(kg/ha))*
- real \*8 **frac\_harvk**
- real \*8 **lid\_vgcl**  
*van Genuchten equation's coefficient, l (none)*
- real \*8 **lid\_vgcm**  
*van Genuchten equation's coefficient, m (none)*
- real \*8 **lid\_qsurf\_total**
- real \*8 **lid\_farea\_sum**
- real \*8, dimension(:,:), allocatable **lid\_cuminf\_last**  
*cumulative amount of water infiltrated into the amended soil layer at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **lid\_cumr\_last**  
*cumulative amount of rainfall at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **lid\_excum\_last**  
*cumulative amount of excess rainfall at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **lid\_f\_last**  
*potential infiltration rate of the amended soil layer at the last time step in a day (mm/mm H2O)*
- real \*8, dimension(:,:), allocatable **lid\_sw\_last**  
*soil water content of the amended soil layer at the last time step in a day (mm/mm H2O)*
- real \*8, dimension(:,:), allocatable **lid\_qsurf**  
*depth of runoff generated on a LID in a given time interval (mm H2O)*
- real \*8, dimension(:,:), allocatable **interval\_last**
- real \*8, dimension(:,:), allocatable **lid\_str\_last**
- real \*8, dimension(:,:), allocatable **lid\_farea**
- real \*8, dimension(:,:), allocatable **lid\_sw\_add**

- real \*8, dimension(:,:), allocatable **lid\_cumqperc\_last**
- real \*8, dimension(:,:), allocatable **lid\_cumirr\_last**
- integer, dimension(:,:), allocatable **gr\_onoff**
- integer, dimension(:,:), allocatable **gr\_imo**
- integer, dimension(:,:), allocatable **gr\_iyr**
- real \*8, dimension(:,:), allocatable **gr\_farea**

*fractional area of a green roof to the HRU (none)*

- real \*8, dimension(:,:), allocatable **gr\_solop**
- real \*8, dimension(:,:), allocatable **gr\_etcoef**
- real \*8, dimension(:,:), allocatable **gr\_fc**
- real \*8, dimension(:,:), allocatable **gr\_wp**
- real \*8, dimension(:,:), allocatable **gr\_ksat**
- real \*8, dimension(:,:), allocatable **gr\_por**
- real \*8, dimension(:,:), allocatable **gr\_hydeff**
- real \*8, dimension(:,:), allocatable **gr\_soldpt**
- integer, dimension(:,:), allocatable **rg\_onoff**
- integer, dimension(:,:), allocatable **rg\_imo**
- integer, dimension(:,:), allocatable **rg\_iyr**
- real \*8, dimension(:,:), allocatable **rg\_farea**
- real \*8, dimension(:,:), allocatable **rg\_solop**
- real \*8, dimension(:,:), allocatable **rg\_etcoef**
- real \*8, dimension(:,:), allocatable **rg\_fc**
- real \*8, dimension(:,:), allocatable **rg\_wp**
- real \*8, dimension(:,:), allocatable **rg\_ksat**
- real \*8, dimension(:,:), allocatable **rg\_por**
- real \*8, dimension(:,:), allocatable **rg\_hydeff**
- real \*8, dimension(:,:), allocatable **rg\_soldpt**
- real \*8, dimension(:,:), allocatable **rg\_dimop**
- real \*8, dimension(:,:), allocatable **rg\_sarea**
- real \*8, dimension(:,:), allocatable **rg\_vol**
- real \*8, dimension(:,:), allocatable **rg\_sth**
- real \*8, dimension(:,:), allocatable **rg\_sdia**
- real \*8, dimension(:,:), allocatable **rg\_bdia**
- real \*8, dimension(:,:), allocatable **rg\_sts**
- real \*8, dimension(:,:), allocatable **rg\_orifice**
- real \*8, dimension(:,:), allocatable **rg\_oheight**
- real \*8, dimension(:,:), allocatable **rg\_odia**
- integer, dimension(:,:), allocatable **cs\_onoff**
- integer, dimension(:,:), allocatable **cs\_imo**
- integer, dimension(:,:), allocatable **cs\_iyr**
- integer, dimension(:,:), allocatable **cs\_grcon**
- real \*8, dimension(:,:), allocatable **cs\_farea**
- real \*8, dimension(:,:), allocatable **cs\_vol**
- real \*8, dimension(:,:), allocatable **cs\_rdepth**
- integer, dimension(:,:), allocatable **pv\_onoff**
- integer, dimension(:,:), allocatable **pv\_imo**
- integer, dimension(:,:), allocatable **pv\_iyr**
- integer, dimension(:,:), allocatable **pv\_solop**
- real \*8, dimension(:,:), allocatable **pv\_grvdep**
- real \*8, dimension(:,:), allocatable **pv\_grvpor**
- real \*8, dimension(:,:), allocatable **pv\_farea**
- real \*8, dimension(:,:), allocatable **pv\_drcoef**
- real \*8, dimension(:,:), allocatable **pv\_fc**
- real \*8, dimension(:,:), allocatable **pv\_wp**

- real \*8, dimension(:,:), allocatable **pv\_ksat**
- real \*8, dimension(:,:), allocatable **pv\_por**
- real \*8, dimension(:,:), allocatable **pv\_hydeff**
- real \*8, dimension(:,:), allocatable **pv\_soldpt**
- integer, dimension(:,:), allocatable **lid\_onoff**
- real \*8, dimension(:,:), allocatable **sol\_hsc**  
*mass of C present in slow humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_hsn**  
*mass of N present in slow humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_hpc**  
*mass of C present in passive humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_hpn**  
*mass of N present in passive humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lm**  
*mass of metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lmc**  
*mass of C in metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lmn**  
*mass of N in metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_ls**  
*mass of structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lsc**  
*mass of C in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lsl**  
*mass of lignin in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_lsn**  
*mass of N in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **sol\_bmc**
- real \*8, dimension(:,:), allocatable **sol\_bmn**
- real \*8, dimension(:,:), allocatable **sol\_rnmn**
- real \*8, dimension(:,:), allocatable **sol\_lslc**
- real \*8, dimension(:,:), allocatable **sol\_lslnc**
- real \*8, dimension(:,:), allocatable **sol\_rspc**
- real \*8, dimension(:,:), allocatable **sol\_woc**
- real \*8, dimension(:,:), allocatable **sol\_won**
- real \*8, dimension(:,:), allocatable **sol\_hp**
- real \*8, dimension(:,:), allocatable **sol\_hs**
- real \*8, dimension(:,:), allocatable **sol\_bm**
- real \*8, dimension(:,:), allocatable **sol\_cac**
- real \*8, dimension(:,:), allocatable **sol\_cec**
- real \*8, dimension(:,:), allocatable **sol\_percc**
- real \*8, dimension(:,:), allocatable **sol\_latc**
- real \*8, dimension(:), allocatable **sedc\_d**  
*amount of C lost with sediment pools (kg C/ha)*
- real \*8, dimension(:), allocatable **surfqc\_d**
- real \*8, dimension(:), allocatable **latc\_d**
- real \*8, dimension(:), allocatable **percc\_d**
- real \*8, dimension(:), allocatable **foc\_d**
- real \*8, dimension(:), allocatable **nppc\_d**
- real \*8, dimension(:), allocatable **rsdc\_d**
- real \*8, dimension(:), allocatable **grainc\_d**
- real \*8, dimension(:), allocatable **stoverc\_d**
- real \*8, dimension(:), allocatable **soc\_d**

- real \*8, dimension(:), allocatable **rspc\_d**
- real \*8, dimension(:), allocatable **emitc\_d**
- real \*8, dimension(:), allocatable **sub\_sedc\_d**
- real \*8, dimension(:), allocatable **sub\_surfqc\_d**
- real \*8, dimension(:), allocatable **sub\_latc\_d**
- real \*8, dimension(:), allocatable **sub\_percc\_d**
- real \*8, dimension(:), allocatable **sub\_foc\_d**
- real \*8, dimension(:), allocatable **sub\_nppc\_d**
- real \*8, dimension(:), allocatable **sub\_rsd\_c\_d**
- real \*8, dimension(:), allocatable **sub\_grainc\_d**
- real \*8, dimension(:), allocatable **sub\_stoverc\_d**
- real \*8, dimension(:), allocatable **sub\_emitc\_d**
- real \*8, dimension(:), allocatable **sub\_soc\_d**
- real \*8, dimension(:), allocatable **sub\_rspc\_d**
- real \*8, dimension(:), allocatable **sedc\_m**
- real \*8, dimension(:), allocatable **surfqc\_m**
- real \*8, dimension(:), allocatable **latc\_m**
- real \*8, dimension(:), allocatable **percc\_m**
- real \*8, dimension(:), allocatable **foc\_m**
- real \*8, dimension(:), allocatable **nppc\_m**
- real \*8, dimension(:), allocatable **rsdc\_m**
- real \*8, dimension(:), allocatable **grainc\_m**
- real \*8, dimension(:), allocatable **stoverc\_m**
- real \*8, dimension(:), allocatable **emitc\_m**
- real \*8, dimension(:), allocatable **soc\_m**
- real \*8, dimension(:), allocatable **rspc\_m**
- real \*8, dimension(:), allocatable **sedc\_a**
- real \*8, dimension(:), allocatable **surfqc\_a**
- real \*8, dimension(:), allocatable **latc\_a**
- real \*8, dimension(:), allocatable **percc\_a**
- real \*8, dimension(:), allocatable **foc\_a**
- real \*8, dimension(:), allocatable **nppc\_a**
- real \*8, dimension(:), allocatable **rsdc\_a**
- real \*8, dimension(:), allocatable **grainc\_a**
- real \*8, dimension(:), allocatable **stoverc\_a**
- real \*8, dimension(:), allocatable **emitc\_a**
- real \*8, dimension(:), allocatable **soc\_a**
- real \*8, dimension(:), allocatable **rspc\_a**
- integer, dimension(:), allocatable **tillage\_switch**
- real \*8, dimension(:), allocatable **tillage\_depth**
- integer, dimension(:), allocatable **tillage\_days**
- real \*8, dimension(:), allocatable **tillage\_factor**
- real \*8 **dthy**  
*time interval for subdaily routing*
- integer, dimension(4) **ihx**
- integer, dimension(:), allocatable **nhy**
- real \*8, dimension(:), allocatable **rchx**
- real \*8, dimension(:), allocatable **rcss**
- real \*8, dimension(:), allocatable **qcap**
- real \*8, dimension(:), allocatable **chxa**
- real \*8, dimension(:), allocatable **chxp**
- real \*8, dimension(:, :, :), allocatable **qhy**
- real \*8 **ff1**
- real \*8 **ff2**



### 5.1.1 Detailed Description

main module containing the global variables

### 5.1.2 Variable Documentation

#### 5.1.2.1 igropt

`integer parm::igropt`

Qual2E option for calculating the local specific growth rate of algae  
1: multiplicative.

$$u = \text{mumax} \text{ fll } f_{nn} \text{ fpp}$$

2: limiting nutrient

$$u = \text{mumax} \text{ fll } \min(f_{nn}, f_{pp})$$

3: harmonic mean

$$u = \text{mumax} \text{ fll } \frac{2}{\frac{1}{f_{nn}} + \frac{1}{f_{pp}}}$$



## Chapter 6

# Data Type Documentation

### 6.1 parm::fcgd Interface Reference

#### Public Member Functions

- real \*8 function **fcgd** (xx)

The documentation for this interface was generated from the following file:

- [modparm.f90](#)

### 6.2 parm::regres Interface Reference

#### Public Member Functions

- real \*8 function **regres** (k, j)

The documentation for this interface was generated from the following file:

- [modparm.f90](#)

### 6.3 parm::tair Interface Reference

#### Public Member Functions

- real \*8 function **tair** (hr, jj)

The documentation for this interface was generated from the following file:

- [modparm.f90](#)

### 6.4 parm::theta Interface Reference

#### Public Member Functions

- real \*8 function **theta** (r20, thk, tmp)

The documentation for this interface was generated from the following file:

- [modparm.f90](#)



# Chapter 7

## File Documentation

### 7.1 albedo.f90 File Reference

#### Functions/Subroutines

- subroutine `albedo` (*j*)  
*this subroutine calculates albedo in the HRU for the day*

#### 7.1.1 Detailed Description

file containing the subroutine `albedo`

##### Author

modified by Javier Burguete

#### 7.1.2 Function/Subroutine Documentation

##### 7.1.2.1 `albedo()`

```
subroutine albedo (  
    integer, intent(in) j )
```

this subroutine calculates albedo in the HRU for the day

##### Parameters

|           |          |            |
|-----------|----------|------------|
| <i>in</i> | <i>j</i> | HRU number |
|-----------|----------|------------|

## 7.2 allocate\_parms.f90 File Reference

### Functions/Subroutines

- subroutine [allocate\\_parms](#)  
*this subroutine allocates array sizes*

#### 7.2.1 Detailed Description

file containing the subroutine allocate\_parms

##### Author

modified by Javier Burguete

## 7.3 alph.f90 File Reference

### Functions/Subroutines

- subroutine [alph](#) (iwave, j)  
*this subroutine computes alpha, a dimensionless parameter that expresses the fraction of total rainfall that occurs during 0.5h*

#### 7.3.1 Detailed Description

file containing the subroutine alph

##### Author

modified by Javier Burguete

#### 7.3.2 Function/Subroutine Documentation

##### 7.3.2.1 alph()

```
subroutine alph (  
    integer, intent(in) iwave,  
    integer, intent(in) j )
```

this subroutine computes alpha, a dimensionless parameter that expresses the fraction of total rainfall that occurs during 0.5h

## Parameters

|    |              |  |
|----|--------------|--|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU MUSLE(sedyld) each hru is calculated independently using hru area and adjusted channel length<br>iwave = 1 subbasin # for subbasin MUSLE is computed for entire subbasin using hru weighted KLSCP |
| in | <i>j</i>     | HRU number   |

## 7.4 anfert.f90 File Reference

### Functions/Subroutines

- subroutine [anfert](#) (j)

*this subroutine automatically applies Nitrogen and Phosphorus when Nitrogen stress exceeds a user input threshold*

#### 7.4.1 Detailed Description

file containing the subroutine anfert

#### Author

modified by Javier Burguete

#### 7.4.2 Function/Subroutine Documentation

##### 7.4.2.1 anfert()

```
subroutine anfert (
    integer, intent(in) j )
```

this subroutine automatically applies Nitrogen and Phosphorus when Nitrogen stress exceeds a user input threshold

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.5 apply.f90 File Reference

### Functions/Subroutines

- subroutine [apply](#) (j)

*this subroutine applies pesticide*

### 7.5.1 Detailed Description

file containing the subroutine apply

Author

modified by Javier Burguete

### 7.5.2 Function/Subroutine Documentation

#### 7.5.2.1 apply()

```
subroutine apply (
    integer, intent(in) j )
```

this subroutine applies pesticide

Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.6 ascrv.f90 File Reference

### Functions/Subroutines

- subroutine [ascrv](#) (x1, x2, x3, x4, x5, x6)  
*this subroutine computes shape parameters x5 and x6 for the S curve equation*

#### 7.6.1 Detailed Description

file containing the subroutine ascrv

Author

modified by Javier Burguete

#### 7.6.2 Function/Subroutine Documentation



### 7.6.2.1 ascrv()

```
subroutine ascrv (
    real*8, intent(in) x1,
    real*8, intent(in) x2,
    real*8, intent(in) x3,
    real*8, intent(in) x4,
    real*8, intent(out) x5,
    real*8, intent(out) x6 )
```

this subroutine computes shape parameters x5 and x6 for the S curve equation

$$x = \frac{y}{y + \exp(x5 + x6 y)}$$

given 2 (x,y) points along the curve. x5 is determined by solving the equation with  $x$  and  $y$  values measured around the midpoint of the curve (approx. 50% of the maximum value for  $x$ ) and x6 is determined by solving the equation with  $x$  and  $y$  values measured close to one of the endpoints of the curve (100% of the maximum value for  $x$ ). This subroutine is called from [readbsn.f90](#) and [readplant.f90](#)

#### Parameters

|     |    |  |
|-----|----|--|
| in  | x1 | value for x in the above equation for first datapoint, x1 should be close to 0.5 (the midpoint of the curve) |
| in  | x2 | value for x in the above equation for second datapoint, x2 should be close to 0.0 or 1.0                     |
| in  | x3 | value for y in the above equation corresponding to x1  |
| in  | x4 | value for y in the above equation corresponding to x2  |
| out | x5 | 1st shape parameter for S curve equation characterizing the midpoint of the curve                            |
| out | x6 | 2nd shape parameter for S curve equation characterizing the regions close to the endpoints of the curve      |

## 7.7 atri.f90 File Reference

### Functions/Subroutines

- real \*8 function [atri](#) (at1, at2, at3, at4i)

*this function generates a random number from a triangular distribution given X axis points at start, end, and peak Y value*

### 7.7.1 Detailed Description

file containing the function atri

#### Author

modified by Javier Burguete

### 7.7.2 Function/Subroutine Documentation

### 7.7.2.1 atri()

```
real*8 function atri (
    real*8, intent(in) at1,
    real*8, intent(in) at2,
    real*8, intent(in) at3,
    integer, intent(inout) at4i )
```

this function generates a random number from a triangular distribution given X axis points at start, end, and peak Y value

#### Parameters

|         |      |                                      |
|---------|------|--------------------------------------|
| in      | at1  | lower limit for distribution (none)  |
| in      | at2  | monthly mean for distribution (none) |
| in      | at3  | upper limit for distribution (none)  |
| in, out | at4i | random number seed (none)            |

#### Returns

daily value generated for distribution (none)

## 7.8 aunif.f90 File Reference

### Functions/Subroutines

- real \*8 function [aunif](#) (x1)

*This function generates random numbers ranging from 0.0 to 1.0. In the process of calculating the random number, the seed (x1) is set to a new value. This function implements the prime-modulus generator.*

### 7.8.1 Detailed Description

file containing the function aunif

#### Author

modified by Javier Burguete

### 7.8.2 Function/Subroutine Documentation

#### 7.8.2.1 aunif()

```
real*8 function aunif (
    integer, intent(inout) x1 )
```

This function generates random numbers ranging from 0.0 to 1.0. In the process of calculating the random number, the seed (x1) is set to a new value. This function implements the prime-modulus generator.

$$xi = 16807 xi \bmod (2^{31} - 1)$$

using code which ensures that no intermediate result uses more than 31 bits. The theory behind the code is summarized in [1]

**Parameters**

|                      |                 |  |
|----------------------|-----------------|--|
| <code>in, out</code> | <code>x1</code> | random number generator seed (integer) where $0 < x1 < 2147483647$ |
|----------------------|-----------------|--|

**Returns**

random number ranging from 0.0 to 1.0

## 7.9 autoirr.f90 File Reference

**Functions/Subroutines**

- subroutine [autoirr](#) (j)  
*this subroutine performs the auto-irrigation operation*

### 7.9.1 Detailed Description

file containing the subroutine autoirr

**Author**

modified by Javier Burguete

### 7.9.2 Function/Subroutine Documentation

#### 7.9.2.1 autoirr()

```
subroutine autoirr (
    integer, intent(in) j )
```

this subroutine performs the auto-irrigation operation

**Parameters**

|                 |                |            |
|-----------------|----------------|------------|
| <code>in</code> | <code>j</code> | HRU number |
|-----------------|----------------|------------|

## 7.10 bacteria.f90 File Reference

**Functions/Subroutines**

- subroutine [bacteria](#) (j)  
*this subroutine calculates bacteria growth, transport with runoff and loss due to percolation into soil*

### 7.10.1 Detailed Description

file containing the subroutine bacteria

Author

modified by Javier Burguete

### 7.10.2 Function/Subroutine Documentation

#### 7.10.2.1 bacteria()

```
subroutine bacteria (
    integer, intent(in) j )
```

this subroutine calculates bacteria growth, transport with runoff and loss due to percolation into soil

Parameters

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.11 bmpinit.f90 File Reference

### Functions/Subroutines

- subroutine [bmpinit](#) (ii)  
*this subroutine sets default values for urban bmp parameters*

#### 7.11.1 Detailed Description

file containing the subroutine bmpinit

Author

modified by Javier Burguete

### 7.11.2 Function/Subroutine Documentation

#### 7.11.2.1 bmpinit()

```
subroutine bmpinit (
    integer, intent(in) ii )
```

this subroutine sets default values for urban bmp parameters

## Parameters

|    |    |                 |
|----|----|-----------------|
| in | ii | subbasin number |
|----|----|-----------------|

## 7.12 burnop.f90 File Reference

### Functions/Subroutines

- subroutine [burnop](#) (j)  
*this subroutine performs burning*

#### 7.12.1 Detailed Description

file containing the subroutine burnop

## Author

modified by Javier Burguete

#### 7.12.2 Function/Subroutine Documentation

##### 7.12.2.1 burnop()

```
subroutine burnop (
    integer, intent(in) j )
```

this subroutine performs burning

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.13 canopyint.f90 File Reference

### Functions/Subroutines

- subroutine [canopyint](#)  
*this subroutine computes canopy interception of rainfall used for methods other than curve number*

### 7.13.1 Detailed Description

file containing the subroutine canopyint

Author

modified by Javier Burguete

## 7.14 caps.f90 File Reference

### Functions/Subroutines

- subroutine `caps` (`file_name`)

*this subroutine reads the input and output names given in file.cio and converts all capital letters to lowercase letters.*

### 7.14.1 Detailed Description

file containing the subroutine caps

Author

modified by Javier Burguete

### 7.14.2 Function/Subroutine Documentation

#### 7.14.2.1 caps()

```
subroutine caps (
    character (len=*) file_name )
```

this subroutine reads the input and output names given in file.cio and converts all capital letters to lowercase letters.

Parameters

|                        |  |
|------------------------|--|
| <code>file_name</code> | dummy argument, file name character string |
|------------------------|--|

## 7.15 carbon\_new.f90 File Reference

### Functions/Subroutines

- subroutine `carbon` (`i`, `j`)

*This code simulates organic C, N, and P cycling in the soil. It has been adapted from [2]. and crafted to accomodate to SWAT conventions. Plant residues and manure residues are decomposed separately. For convenience, the denitrification subroutine is called from here. March 2009: testing has been minimal and further adjustments are expected. Manuscript describing this subroutine to be submitted to Ecological Modelling (September, 2010). Use with caution and report anomalous results to [akemanian@psu.edu](mailto:akemanian@psu.edu), [jeff.arnold@ars.usda.edu](mailto:jeff.arnold@ars.usda.edu) and [stefan.julich@tudor.lu](mailto:stefan.julich@tudor.lu).*

- real \*8 function **fwf** (fc, wc, pwp)
- real \*8 function **fof** (void, por)
- real \*8 function **fcgd** (xx)
- real \*8 function **ftilf** (tillage, wc, sat)
- real \*8 function **fcx** (pclay)
- real \*8 function **fsol\_cdec** (pcarbon, cx, cfdec, tilf, csf, sol\_cmass)
- real \*8 function **fcnnew** (yy1, yy2, CNpool, yy5)
- real \*8 function **fhc** (pclay, pcarbon, cx)
- real \*8 function **fnetmin** (poold, R1, R2, hc, dummy, poolm, xinorg, cc1)

### 7.15.1 Detailed Description

file containing the subroutine carbon

#### Author

Armen R. Kemanian,  
Stefan Julich,  
modified by Javier Burguete

### 7.15.2 Function/Subroutine Documentation

#### 7.15.2.1 carbon()

```
subroutine carbon (
    integer, intent(in) i,
    integer, intent(in) j )
```

This code simulates organic C, N, and P cycling in the soil. It has been adapted from [2]. and crafted to accomodate to SWAT conventions. Plant residues and manure residues are decomposed separately. For convenience, the denitrification subroutine is called from here. March 2009: testing has been minimal and further adjustments are expected. Manuscript describing this subroutine to be submitted to Ecological Modelling (September, 2010). Use with caution and report anomalous results to [akemanian@psu.edu](mailto:akemanian@psu.edu), [jeff.arnold@ars.usda.edu](mailto:jeff.arnold@ars.usda.edu) and [stefan.julich@tudor.lu](mailto:stefan.julich@tudor.lu).

#### Parameters

|          |  |
|----------|--|
| <i>i</i> | current day in simulation—loop counter (julian date) |
| <i>j</i> | HRU number   |

## 7.16 carbon\_zhang2.f90 File Reference

### Functions/Subroutines

- subroutine [carbon\\_zhang2](#) (j)

#### 7.16.1 Detailed Description

file containing the subroutine carbon\_zhang2

##### Author

modified by Javier Burguete

#### 7.16.2 Function/Subroutine Documentation

##### 7.16.2.1 carbon\_zhang2()

```
subroutine carbon_zhang2 (  
    integer, intent(in) j )
```

##### Parameters

|          |            |
|----------|------------|
| <i>j</i> | HRU number |
|----------|------------|

## 7.17 cfactor.f90 File Reference

### Functions/Subroutines

- subroutine [cfactor](#)  
*this subroutine predicts daily soil loss caused by water erosion using the modified universal soil loss equation*

#### 7.17.1 Detailed Description

file containing the subroutine cfactor

##### Author

modified by Javier Burguete



## 7.18 clgen.f90 File Reference

### Functions/Subroutines

- subroutine [clgen](#) (j)

*this subroutine calculates the daylength, distribution of radiation throughout the day and maximum radiation for day*

#### 7.18.1 Detailed Description

file containing the subroutine clgen

##### Author

modified by Javier Burguete

#### 7.18.2 Function/Subroutine Documentation

##### 7.18.2.1 clgen()

```
subroutine clgen (
    integer, intent(in) j )
```

this subroutine calculates the daylength, distribution of radiation throughout the day and maximum radiation for day

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.19 clicon.f90 File Reference

### Functions/Subroutines

- subroutine [clicon](#) (i)

*this subroutine controls weather inputs to SWAT. Precipitation and temperature data is read in and the weather generator is called to fill in radiation, wind speed and relative humidity as well as missing precipitation and temperatures. Adjustments for climate changes studies are also made in this subroutine.*

#### 7.19.1 Detailed Description

file containing the subroutine clicon

##### Author

modified by Javier Burguete

## 7.19.2 Function/Subroutine Documentation

### 7.19.2.1 clicon()

```
subroutine clicon (
    integer, intent(in) i )
```

this subroutine controls weather inputs to SWAT. Precipitation and temperature data is read in and the weather generator is called to fill in radiation, wind speed and relative humidity as well as missing precipitation and temperatures. Adjustments for climate changes studies are also made in this subroutine.

#### Parameters

|    |   |   |
|----|---|---|
| in | i | current day of simulation (julian date) |
|----|---|---|

## 7.20 command.f90 File Reference

### Functions/Subroutines

- subroutine [command](#) (i)

*for every day of simulation, this subroutine steps through the command lines in the watershed configuration (.fig) file. Depending on the command code on the .fig file line, a command loop is accessed*

### 7.20.1 Detailed Description

file containing the subroutine command

#### Author

modified by Javier Burguete

## 7.20.2 Function/Subroutine Documentation

### 7.20.2.1 command()

```
subroutine command (
    integer, intent(in) i )
```

for every day of simulation, this subroutine steps through the command lines in the watershed configuration (.fig) file. Depending on the command code on the .fig file line, a command loop is accessed

## Parameters

|    |          |  |
|----|----------|--|
| in | <i>i</i> | current day in simulation—loop counter (julian date) |
|----|----------|--|

## 7.21 conapply.f90 File Reference

### Functions/Subroutines

- subroutine `conapply` (*j*)  
*this subroutine applies continuous pesticide*

#### 7.21.1 Detailed Description

file containing the subroutine `conapply`

#### Author

modified by Javier Burguete

#### 7.21.2 Function/Subroutine Documentation

##### 7.21.2.1 `conapply()`

```
subroutine conapply (
    integer, intent(in) j )
```

this subroutine applies continuous pesticide

## Parameters

|    |          |            |
|----|----------|------------|
| in | <i>h</i> | HRU number |
|----|----------|------------|

## 7.22 confert.f90 File Reference

### Functions/Subroutines

- subroutine `confert` (*j*)  
*this subroutine simulates a continuous fertilizer operation*

### 7.22.1 Detailed Description

file containing the subroutine confert

Author

modified by Javier Burguete

### 7.22.2 Function/Subroutine Documentation

#### 7.22.2.1 confert()

```
subroutine confert (
    integer, intent(in) j )
```

this subroutine simulates a continuous fertilizer operation

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.23 crackflow.f90 File Reference

### Functions/Subroutines

- subroutine [crackflow](#)  
*this subroutine modifies surface runoff to account for crack flow*

#### 7.23.1 Detailed Description

file containing the subroutine crackflow

Author

modified by Javier Burguete

## 7.24 crackvol.f90 File Reference

### Functions/Subroutines

- subroutine [crackvol](#)  
*this subroutine computes total crack volume for the soil profile and modifies surface runoff to account for crack flow*

### 7.24.1 Detailed Description

file containing the subroutine crackvol

Author

modified by Javier Burguete

## 7.25 curno.f90 File Reference

### Functions/Subroutines

- subroutine `curno` (cnn, h)

*this subroutine determines the curve numbers for moisture conditions I and III and calculates coefficients and shape parameters for the water retention curve. The coefficients and shape parameters are calculated by one of two methods:*

*the default method is to make them a function of soil water,*

*the alternative method (labeled new) is to make them a function of accumulated PET, precipitation and surface runoff*

### 7.25.1 Detailed Description

file containing the subroutine curno

Author

modified by Javier Burguete

### 7.25.2 Function/Subroutine Documentation

#### 7.25.2.1 curno()

```
subroutine curno (
    real*8, intent(in) cnn,
    integer, intent(in) h )
```

this subroutine determines the curve numbers for moisture conditions I and III and calculates coefficients and shape parameters for the water retention curve. The coefficients and shape parameters are calculated by one of two methods:

the default method is to make them a function of soil water,

the alternative method (labeled new) is to make them a function of accumulated PET, precipitation and surface runoff

#### Parameters

|    |            |   |
|----|------------|---|
| in | <i>cnn</i> | SCS runoff curve number for moisture condition II |
| in | <i>h</i>   | HRU number  |

## 7.26 `dailycn.f90` File Reference

### Functions/Subroutines

- subroutine `dailycn`  
*calculates curve number for the day in the HRU*

#### 7.26.1 Detailed Description

file containing the subroutine `dailycn`

Author

modified by Javier Burguete

## 7.27 `decay.f90` File Reference

### Functions/Subroutines

- subroutine `decay` (*j*)  
*this subroutine calculates degradation of pesticide in the soil and on the plants*

#### 7.27.1 Detailed Description

file containing the subroutine `decay`

Author

modified by Javier Burguete

#### 7.27.2 Function/Subroutine Documentation

##### 7.27.2.1 `decay()`

```
subroutine decay (
    integer, intent(in) j )
```

this subroutine calculates degradation of pesticide in the soil and on the plants

Parameters

|                 |                |            |
|-----------------|----------------|------------|
| <code>in</code> | <code>j</code> | HRU number |
|-----------------|----------------|------------|

## 7.28 depstor.f90 File Reference

### Functions/Subroutines

- subroutine [depstor](#) (j)

*this subroutine computes maximum surface depressional storage depth based on random and oriented roughness and slope steepness*

#### 7.28.1 Detailed Description

file containing the subroutine depstor

Author

modified by Javier Burguete

#### 7.28.2 Function/Subroutine Documentation

##### 7.28.2.1 depstor()

```
subroutine depstor (
    integer, intent(in) j )
```

this subroutine computes maximum surface depressional storage depth based on random and oriented roughness and slope steepness

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.29 dormant.f90 File Reference

### Functions/Subroutines

- subroutine [dormant](#) (j)

*this subroutine checks the dormant status of the different plant types*

#### 7.29.1 Detailed Description

file containing the subroutine dormant

Author

modified by Javier Burguete

## 7.29.2 Function/Subroutine Documentation

### 7.29.2.1 dormant()

```
subroutine dormant (
    integer, intent(in) j )
```

this subroutine checks the dormant status of the different plant types

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.30 drains.f90 File Reference

### Functions/Subroutines

- subroutine [drains](#) (*j*)

*this subroutine finds the effective lateral hydraulic conductivity and computes drainage or subirrigation flux*

### 7.30.1 Detailed Description

file containing the subroutine drains

#### Author

modified by Javier Burguete

## 7.30.2 Function/Subroutine Documentation

### 7.30.2.1 drains()

```
subroutine drains (
    integer, intent(in) j )
```

this subroutine finds the effective lateral hydraulic conductivity and computes drainage or subirrigation flux

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|



## 7.31 dstn1.f90 File Reference

### Functions/Subroutines

- real \*8 function [dstn1](#) (rn1, rn2)

*this function computes the distance from the mean of a normal distribution with mean = 0 and standard deviation = 1, given two random numbers*

#### 7.31.1 Detailed Description

file containing the function dstn1

Author

modified by Javier Burguete

#### 7.31.2 Function/Subroutine Documentation

##### 7.31.2.1 dstn1()

```
real*8 function dstn1 (  
    real*8, intent(in) rn1,  
    real*8, intent(in) rn2 )
```

this function computes the distance from the mean of a normal distribution with mean = 0 and standard deviation = 1, given two random numbers

Parameters

|    |            |                      |
|----|------------|----------------------|
| in | <i>rn1</i> | first random number  |
| in | <i>rn2</i> | second random number |

Returns

distance from the mean

## 7.32 ee.f90 File Reference

### Functions/Subroutines

- real \*8 function [ee](#) (tk)

*this function calculates saturation vapor pressure at a given air temperature*

### 7.32.1 Detailed Description

file containing the function ee

Author

modified by Javier Burguete

### 7.32.2 Function/Subroutine Documentation

#### 7.32.2.1 ee()

```
real*8 function ee (
      real*8, intent(in) tk )
```

this function calculates saturation vapor pressure at a given air temperature

Parameters

|    |    |                              |
|----|----|------------------------------|
| in | tk | mean air temperature (deg C) |
|----|----|------------------------------|

Returns

saturation vapor pressure (kPa)

## 7.33 eiusle.f90 File Reference

### Functions/Subroutines

- subroutine [eiusle](#) (j)  
*this subroutine computes the USLE erosion index (EI)*

#### 7.33.1 Detailed Description

file containing the subroutine eiusle

Author

modified by Javier Burguete

## 7.34 enrsb.f90 File Reference

### Functions/Subroutines

- subroutine [enrsb](#) (iwave, j)  
*this subroutine calculates the enrichment ratio for nutrient and pesticide transport with runoff*

### 7.34.1 Detailed Description

file containing the subroutine enrsb

#### Author

modified by Javier Burguete

### 7.34.2 Function/Subroutine Documentation

#### 7.34.2.1 enrsb()

```
subroutine enrsb (  
    integer, intent(in) iwave,  
    integer, intent(in) j )
```

this subroutine calculates the enrichment ratio for nutrient and pesticide transport with runoff

#### Parameters

|    |              |   |
|----|--------------|---|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| in | <i>j</i>     | HRU number  |

## 7.35 estimate\_ksat.f90 File Reference

### Functions/Subroutines

- subroutine [estimate\\_ksat](#) (perc\_clay, esti\_ksat)

*This subroutine calculates ksat value for a soil layer given the % of clay in the soil layer.*

### 7.35.1 Detailed Description

file containing the subroutine estimate\_ksat

#### Author

modified by Javier Burguete

### 7.35.2 Function/Subroutine Documentation

### 7.35.2.1 estimate\_ksat()

```
subroutine estimate_ksat (
    real*8, intent(in) perc_clay,
    real*8, intent(out) esti_ksat )
```

This subroutine calculates ksats value for a soil layer given the % of clay in the soil layer.

Background: published work of Walter Rawls. Calculated ksats values based on soil texture (sand, silt and clay).  
Idea: there exists a relationship between % clay and Ksat. Equations used in this subroutine are based on the above idea (Jimmy Willimas)

#### Parameters

|     |                  |                     |
|-----|------------------|---------------------|
| in  | <i>perc_clay</i> | clay percentage (%) |
| out | <i>esti_ksat</i> | estimated ksats     |

## 7.36 etact.f90 File Reference

### Functions/Subroutines

- subroutine [etact](#) (j)  
*this subroutine calculates potential plant transpiration for Priestley- Taylor and Hargreaves ET methods, and potential and actual soil evaporation. NO3 movement into surface soil layer due to evaporation is also calculated.*

### 7.36.1 Detailed Description

file containing the subroutine etact

#### Author

modified by Javier Burguete

## 7.37 etpot.f90 File Reference

### Functions/Subroutines

- subroutine [etpot](#) (j)  
*this subroutine calculates potential evapotranspiration using one of three methods. If Penman-Monteith is being used, potential plant transpiration is also calculated.*

### 7.37.1 Detailed Description

file containing the subroutine etpot

#### Author

modified by Javier Burguete

## 7.37.2 Function/Subroutine Documentation

### 7.37.2.1 etpot()

```
subroutine etpot (
    integer, intent(in) j )
```

this subroutine calculates potential evapotranspiration using one of three methods. If Penman-Monteith is being used, potential plant transpiration is also calculated.

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.38 expo.f90 File Reference

### Functions/Subroutines

- real \*8 function `expo` (xx)

*this function checks the argument against upper and lower boundary values prior to taking the Exponential*

### 7.38.1 Detailed Description

file containing the function expo

#### Author

modified by Javier Burguete

## 7.38.2 Function/Subroutine Documentation

### 7.38.2.1 expo()

```
real*8 function expo (
    real*8, intent(in) xx )
```

this function checks the argument against upper and lower boundary values prior to taking the Exponential

#### Parameters

|    |    |                             |
|----|----|-----------------------------|
| in | xx | exponential argument (none) |
|----|----|-----------------------------|

**Returns**

$\exp(xx)$

**7.39 fert.f90 File Reference****Functions/Subroutines**

- subroutine [fert](#) (j)  
*this subroutine applies N and P specified by date and amount in the management file (.mgt)*

**7.39.1 Detailed Description**

file containing the subroutine fert

**Author**

modified by Javier Burquete

**7.39.2 Function/Subroutine Documentation****7.39.2.1 fert()**

```
subroutine fert (
    integer, intent(in) j )
```

this subroutine applies N and P specified by date and amount in the management file (.mgt)

**Parameters**

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

**7.40 finalbal.f90 File Reference****Functions/Subroutines**

- subroutine [finalbal](#)  
*this subroutine calculates final water balance for watershed*

**7.40.1 Detailed Description**

file containing the subroutine finalbal

Author

modified by Javier Burguete

## 7.41 gcycl.f90 File Reference

### Functions/Subroutines

- subroutine [gcycl](#)

*This subroutine initializes the random number seeds. If the user desires a different set of random numbers for each simulation run, the random number generator is used to reset the values of the seeds.*

#### 7.41.1 Detailed Description

file containing the subroutine gcycl

Author

modified by Javier Burguete

## 7.42 getallo.f90 File Reference

### Functions/Subroutines

- subroutine [getallo](#)

*This subroutine calculates the number of HRUs, subbasins, etc. in the simulation. These values are used to allocate array sizes.*

#### 7.42.1 Detailed Description

file containing the subroutine getallo

Author

modified by Javier Burguete

## 7.43 graze.f90 File Reference

### Functions/Subroutines

- subroutine [graze](#) (j)

*this subroutine simulates biomass lost to grazing*

### 7.43.1 Detailed Description

file containing the subroutine graze

Author

modified by Javier Burguete

### 7.43.2 Function/Subroutine Documentation

#### 7.43.2.1 graze()

```
subroutine graze (
    integer, intent(in) j )
```

this subroutine simulates biomass lost to grazing

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.44 grow.f90 File Reference

### Functions/Subroutines

- subroutine [grow](#) (j)  
*this subroutine adjusts plant biomass, leaf area index, and canopy height taking into account the effect of water, temperature and nutrient stresses on the plant*

#### 7.44.1 Detailed Description

file containing the subroutine grow

Author

modified by Javier Burguete

#### 7.44.2 Function/Subroutine Documentation

##### 7.44.2.1 grow()

```
subroutine grow (
    integer, intent(in) j )
```

this subroutine adjusts plant biomass, leaf area index, and canopy height taking into account the effect of water, temperature and nutrient stresses on the plant



## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.45 gwmod.f90 File Reference

### Functions/Subroutines

- subroutine [gwmod](#) (j)  
*this subroutine estimates groundwater contribution to streamflow*

#### 7.45.1 Detailed Description

file containing the subroutine gwmod

## Author

modified by Javier Burguete

#### 7.45.2 Function/Subroutine Documentation

##### 7.45.2.1 gwmod()

```
subroutine gwmod (
    integer, intent(in) j )
```

this subroutine estimates groundwater contribution to streamflow

## Parameters

|   |            |
|---|------------|
| j | HRU number |
|---|------------|

## 7.46 gwmod\_deep.f90 File Reference

### Functions/Subroutines

- subroutine [gwmod\\_deep](#) (j)  
*this subroutine estimates groundwater contribution to streamflow*

### 7.46.1 Detailed Description

file containing the subroutine gwmod\_deep

Author

modified by Javier Burguete

### 7.46.2 Function/Subroutine Documentation

#### 7.46.2.1 gwmod\_deep()

```
subroutine gwmod_deep (
    integer, intent(in) j )
```

this subroutine estimates groundwater contribution to streamflow

Parameters

|          |            |
|----------|------------|
| <i>j</i> | HRU number |
|----------|------------|

## 7.47 h2omgt\_init.f90 File Reference

### Functions/Subroutines

- subroutine [h2omgt\\_init](#)

*This subroutine initializes variables related to water management (irrigation, consumptive water use, etc.)*

#### 7.47.1 Detailed Description

file containing the subroutine h2omgt\_init

Author

modified by Javier Burguete

## 7.48 harvestop.f90 File Reference

### Functions/Subroutines

- subroutine [harvestop](#) (j)

*this subroutine performs the harvest operation (no kill)*

### 7.48.1 Detailed Description

file containing the subroutine harvestop

Author

modified by Javier Burguete

### 7.48.2 Function/Subroutine Documentation

#### 7.48.2.1 harvestop()

```
subroutine harvestop (
    integer, intent(in) j )
```

this subroutine performs the harvest operation (no kill)

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.49 harvkillop.f90 File Reference

### Functions/Subroutines

- subroutine [harvkillop](#) (j)  
*this subroutine performs the harvest and kill operation*

### 7.49.1 Detailed Description

file containing the subroutine harvkillop

Author

modified by Javier Burguete

### 7.49.2 Function/Subroutine Documentation

#### 7.49.2.1 harvkillop()

```
subroutine harvkillop (
    integer, intent(in) j )
```

this subroutine performs the harvest and kill operation

## Parameters

|                       |          |            |
|-----------------------|----------|------------|
| <i>i</i> <sub>n</sub> | <i>j</i> | HRU number |
|-----------------------|----------|------------|

## 7.50 headout.f90 File Reference

### Functions/Subroutines

- subroutine [headout](#)  
*this subroutine writes the headings to the major output files*

#### 7.50.1 Detailed Description

file containing the subroutine headout

## Author

modified by Javier Burguete

## 7.51 hmeas.f90 File Reference

### Functions/Subroutines

- subroutine [hmeas](#)  
*this subroutine reads in relative humidity data from file and assigns the data to the HRUs*

#### 7.51.1 Detailed Description

file containing the subroutine hmeas

## Author

modified by Javier Burguete

## 7.52 hruaa.f90 File Reference

### Functions/Subroutines

- subroutine [hruaa](#) (years)  
*this subroutine writes average annual HRU output to the output.hru file*

### 7.52.1 Detailed Description

file containing the subroutine hruaa

Author

modified by Javier Burguete

### 7.52.2 Function/Subroutine Documentation

#### 7.52.2.1 hruaa()

```
subroutine hruaa (
    real*8, intent(in) years )
```

this subroutine writes average annual HRU output to the output.hru file

Parameters

|    |       |                              |
|----|-------|------------------------------|
| in | years | length of simulation (years) |
|----|-------|------------------------------|

## 7.53 hruallo.f90 File Reference

### Functions/Subroutines

- subroutine [hruallo](#)  
*This subroutine calculates the number of management operation types, etc. used in the simulation. These values are used to allocate array sizes for processes occurring in the HRU.*

#### 7.53.1 Detailed Description

file containing the subroutine hruallo

Author

modified by Javier Burguete

## 7.54 hrumon.f90 File Reference

### Functions/Subroutines

- subroutine [hrumon](#)  
*this subroutine writes monthly HRU output to the output.hru file*

### 7.54.1 Detailed Description

file containing the subroutine hrumon

Author

modified by Javier Burguete

## 7.55 hruyr.f90 File Reference

### Functions/Subroutines

- subroutine [hruyr](#)  
*this subroutine writes annual HRU output to the output.hru file*

### 7.55.1 Detailed Description

file containing the subroutine hruyr

Author

modified by Javier Burguete

## 7.56 hydroinit.f90 File Reference

### Functions/Subroutines

- subroutine [hydroinit](#)  
*This subroutine computes variables related to the watershed hydrology: the time of concentration for the subbasins, lagged surface runoff, the coefficient for the peak runoff rate equation, and lateral flow travel time.*

### 7.56.1 Detailed Description

file containing the subroutine hydroinit

Author

modified by Javier Burguete

## 7.57 icl.f90 File Reference

### Functions/Subroutines

- integer function [icl](#) (id)  
*this function determines the month and day, given the julian date*

### 7.57.1 Detailed Description

file containing the function icl

Author

modified by Javier Burguete

### 7.57.2 Function/Subroutine Documentation

#### 7.57.2.1 icl()

```
integer function icl (
    integer, intent(in) id )
```

this function determines the month and day, given the julian date

Parameters

|    |    |             |
|----|----|-------------|
| in | id | julian date |
|----|----|-------------|

## 7.58 impnd\_init.f90 File Reference

### Functions/Subroutines

- subroutine [impnd\\_init](#)  
*this subroutine initializes variables related to impoundments (ponds, wetlands, reservoirs and potholes)*

#### 7.58.1 Detailed Description

file containing the subroutine impnd\_init

Author

modified by Javier Burguete

## 7.59 impndmon.f90 File Reference

### Functions/Subroutines

- subroutine [impndmon](#)  
*this subroutine writes monthly HRU impoundment output to the output.wtr file*

### 7.59.1 Detailed Description

file containing the subroutine `impndmon`

Author

modified by Javier Burguete

## 7.60 `impndyr.f90` File Reference

### Functions/Subroutines

- subroutine `impndyr`  
*this subroutine writes annual HRU impoundment output to the `output.wtr` file*

### 7.60.1 Detailed Description

file containing the subroutine `impndyr`

Author

modified by Javier Burguete

## 7.61 `irrigate.f90` File Reference

### Functions/Subroutines

- subroutine `irrigate` (`j`, `volmm`)  
*this subroutine applies irrigation water to HRU*

### 7.61.1 Detailed Description

file containing the subroutine `irrigate`

Author

modified by Javier Burguete

### 7.61.2 Function/Subroutine Documentation

#### 7.61.2.1 `irrigate()`

```
subroutine irrigate (  
    integer, intent(in) j,  
    real*8, intent(in) volmm )
```

this subroutine applies irrigation water to HRU



## Parameters

|    |              |  |
|----|--------------|--|
| in | <i>j</i>     | HRU number (none)                              |
| in | <i>volmm</i> | depth irrigation water applied to HRU (mm H2O) |

## 7.62 irrsb.f90 File Reference

### Functions/Subroutines

- subroutine [irrsb](#) (*j*)

*this subroutine performs the irrigation operation when the source is the shallow or deep aquifer or a source outside the watershed*

#### 7.62.1 Detailed Description

file containing the subroutine irrsb

#### Author

modified by Javier Burguete

#### 7.62.2 Function/Subroutine Documentation

##### 7.62.2.1 irrsb()

```
subroutine irrsb (
    integer, intent(in) j )
```

this subroutine performs the irrigation operation when the source is the shallow or deep aquifer or a source outside the watershed

## Parameters

|    |          |                   |
|----|----------|-------------------|
| in | <i>j</i> | HRU number (none) |
|----|----------|-------------------|

## 7.63 jdt.f90 File Reference

### Functions/Subroutines

- integer function [jdt](#) (numdays, i, m)

*this function computes the julian date given the month and the day of the month*

### 7.63.1 Detailed Description

file containing the function jdt

Author

modified by Javier Burguete

### 7.63.2 Function/Subroutine Documentation

#### 7.63.2.1 jdt()

```
integer function jdt (
    integer, dimension (13), intent(in) numdays,
    integer, intent(in) i,
    integer, intent(in) m )
```

this function computes the julian date given the month and the day of the month

Parameters

|    |                |   |
|----|----------------|---|
| in | <i>numdays</i> | julian date for last day of preceding month (where the array location is the number of the month). The dates are for leap years (numdays=ndays) (julian date) |
| in | <i>i</i>       | day   |
| in | <i>m</i>       | month   |

## 7.64 killop.f90 File Reference

### Functions/Subroutines

- subroutine [killop](#) (j)  
*this subroutine performs the kill operation*

#### 7.64.1 Detailed Description

file containing the subroutine killop

Author

modified by Javier Burguete

#### 7.64.2 Function/Subroutine Documentation

### 7.64.2.1 killop()

```
subroutine killop (
    integer, intent(in) j )
```

this subroutine performs the kill operation

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.65 lid\_cistern.f90 File Reference

### Functions/Subroutines

- subroutine [lid\\_cistern](#) (sb, j, k, lid\_prec)  
*simulate cistern processes*

### 7.65.1 Detailed Description

file containing the subroutine lid\_cistern

#### Author

modified by Javier Burguete

### 7.65.2 Function/Subroutine Documentation

#### 7.65.2.1 lid\_cistern()

```
subroutine lid_cistern (
    integer, intent(in) sb,
    integer, intent(in) j,
    integer, intent(in) k,
    real*8, intent(in) lid_prec )
```

simulate cistern processes

#### Parameters

|    |                 |   |
|----|-----------------|---|
| in | <i>sb</i>       | subbasin number (none)  |
| in | <i>j</i>        | HRU number (none)   |
| in | <i>k</i>        | subdaily time index (none)  |
| in | <i>lid_prec</i> | precipitation depth a LID receives in a simulation time interval (mm) |

## 7.66 lid\_greenroof.f90 File Reference

### Functions/Subroutines

- subroutine [lid\\_greenroof](#) (sb, j, k, lid\_prec)  
*simulate green roof processes*

#### 7.66.1 Detailed Description

file containing the subroutine lid\_greenroof

#### Author

modified by Javier Burguete

#### 7.66.2 Function/Subroutine Documentation

##### 7.66.2.1 lid\_greenroof()

```
subroutine lid_greenroof (
    integer, intent(in) sb,
    integer, intent(in) j,
    integer, intent(in) k,
    real*8, intent(in) lid_prec )
```

simulate green roof processes

#### Parameters

|    |                 |   |
|----|-----------------|---|
| in | <i>sb</i>       | subbasin number (none)  |
| in | <i>j</i>        | HRU number (none)   |
| in | <i>k</i>        | subdaily time index (none)  |
| in | <i>lid_prec</i> | precipitation depth a LID receives in a simulation time interval (mm) |

## 7.67 lid\_porpavement.f90 File Reference

### Functions/Subroutines

- subroutine [lid\\_porpavement](#) (sb, j, k, lid\_prec)  
*simulate porous pavement processes*

### 7.67.1 Detailed Description

file containing the subroutine lid\_porpavement

Author

modified by Javier Burguete

### 7.67.2 Function/Subroutine Documentation

#### 7.67.2.1 lid\_porpavement()

```
subroutine lid_porpavement (
    integer, intent(in) sb,
    integer, intent(in) j,
    integer, intent(in) k,
    real*8, intent(in) lid_prec )
```

simulate porous pavement processes

Parameters

|    |                 |   |
|----|-----------------|---|
| in | <i>sb</i>       | subbasin number (none)  |
| in | <i>j</i>        | HRU number (none)   |
| in | <i>k</i>        | subdaily time index (none)  |
| in | <i>lid_prec</i> | precipitation depth a LID receives in a simulation time interval (mm) |

## 7.68 lid\_raingarden.f90 File Reference

### Functions/Subroutines

- subroutine [lid\\_raingarden](#) (sb, j, k, lid\_prec)  
*simulate rain garden processes*

### 7.68.1 Detailed Description

file containing the subroutine lid\_raingarden

Author

modified by Javier Burguete

## 7.68.2 Function/Subroutine Documentation

### 7.68.2.1 lid\_raingarden()

```
subroutine lid_raingarden (
    integer, intent(in) sb,
    integer, intent(in) j,
    integer, intent(in) k,
    real*8, intent(in) lid_prec )
```

simulate rain garden processes

#### Parameters

|    |                 |   |
|----|-----------------|---|
| in | <i>sb</i>       | subbasin number (none)  |
| in | <i>j</i>        | HRU number (none)   |
| in | <i>k</i>        | subdaily time index (none)  |
| in | <i>lid_prec</i> | precipitation depth a LID receives in a simulation time interval (mm) |

## 7.69 lidinit.f90 File Reference

### Functions/Subroutines

- subroutine [lidinit](#) (i)  
*this subroutine sets default values for LID parameters*

### 7.69.1 Detailed Description

file containing the subroutine lidinit

#### Author

modified by Javier Burguete

## 7.69.2 Function/Subroutine Documentation

### 7.69.2.1 lidinit()

```
subroutine lidinit (
    integer, intent(in) i )
```

this subroutine sets default values for LID parameters

## Parameters

|    |          |                 |
|----|----------|-----------------|
| in | <i>i</i> | subbasin number |
|----|----------|-----------------|

## 7.70 lids.f90 File Reference

### Functions/Subroutines

- subroutine [lids](#) (sb, j, k, lid\_prec)  
*call subroutines to simulate green roof, rain garden, cistern and porous pavement processes*

#### 7.70.1 Detailed Description

file containing the subroutine lids

## Author

modified by Javier Burquete

#### 7.70.2 Function/Subroutine Documentation

##### 7.70.2.1 lids()

```
subroutine lids (
    integer, intent(in) sb,
    integer, intent(in) j,
    integer, intent(in) k,
    real*8, intent(in) lid_prec )
```

call subroutines to simulate green roof, rain garden, cistern and porous pavement processes

## Parameters

|    |                 |   |
|----|-----------------|---|
| in | <i>sb</i>       | subbasin number (none)  |
| in | <i>j</i>        | HRU number (none)   |
| in | <i>k</i>        | subdaily time index (none)  |
| in | <i>lid_prec</i> | precipitation depth a LID receives in a simulation time interval (mm) |

## 7.71 lwqdef.f90 File Reference

### Functions/Subroutines

- subroutine [lwqdef](#) (ii)

*this subroutine assigns default values for the lake water quality (.lwq) when the lake water quality file does not exists*

#### 7.71.1 Detailed Description

file containing the subroutine lwqdef

Author

modified by Javier Burguete

#### 7.71.2 Function/Subroutine Documentation

##### 7.71.2.1 lwqdef()

```
subroutine lwqdef (
    integer, intent(in) ii )
```

this subroutine assigns default values for the lake water quality (.lwq) when the lake water quality file does not exists

Parameters

|    |    |                         |
|----|----|-------------------------|
| in | ii | reservoir number (none) |
|----|----|-------------------------|

## 7.72 main.f90 File Reference

### Functions/Subroutines

- program [main](#)

*this is the main program that reads input, calls the main simulation model, and writes output*

#### 7.72.1 Detailed Description

file containing the main program that reads input, calls the main simulation model, and writes output.

Author

modified by Javier Burguete Tolosa



## 7.73 modparm.f90 File Reference

### Data Types

- interface [parm::fcgd](#)
- interface [parm::regres](#)
- interface [parm::tair](#)
- interface [parm::theta](#)

### Modules

- module [parm](#)  
*main module containing the global variables*

### Variables

- integer, parameter [parm::mvaro](#) = 33  
*max number of variables routed through the reach*
- integer, parameter [parm::mhruo](#) = 79  
*maximum number of variables written to HRU output file (output.hru) (none)*
- integer, parameter [parm::mrcho](#) = 62  
*maximum number of variables written to reach output file (.rch) (none)*
- integer, parameter [parm::msubo](#) = 24  
*maximum number of variables written to subbasin output file (output.sub) (none)*
- integer, parameter [parm::mstdo](#) = 113  
*max number of variables summarized in output.std*
- integer, parameter [parm::motot](#) = 600
- character(len=80), parameter [parm::prog](#) = "SWAT Sep 7 VER 2018/Rev 670"  
*SWAT program header string (name and version)*
- character(len=13), dimension(mhruo), parameter [parm::heds](#) = (/ "PRECIPmm", "SNOFALLmm", "SNOMELTmm", "IRRmm", "PETmm", "ETmm", "SW\_INITmm", "SW\_ENDmm", "PERCmm", "GW\_RCHGmm", "DA\_RCHGmm", "REVAPmm", "SA\_IRRmm", "DA\_IRRmm", "SA\_STmm", "DA\_STmm", "SURQ\_GENmm", "SURQ\_CNTmm", "TLOSSmm", "LATQGENmm", "GW\_Qmm", "WYLDmm", "DAILYCN", "TMP\_AVDgC", "TMP\_MXdgC", "TMP\_MNdgc", "SOL\_TMPdgC", "SOLARMJ/m2", "SYLDt/ha", "USLEt/ha", "N\_APPkg/ha", "P\_APPkg/ha", "NAUTOkg/ha", "PAUTOkg/ha", "NGRZkg/ha", "PGRZkg/ha", "NCFRTkg/ha", "PCFRTkg/ha", "NRAINkg/ha", "NFIKkg/ha", "F-MNkg/ha", "A-MNkg/ha", "A-SNkg/ha", "F-MPkg/ha", "AO-L\_Pkg/ha", "L-APkg/ha", "A-SPkg/ha", "DNITkg/ha", "NUPkg/ha", "PUPkg/ha", "ORGNkg/ha", "ORGPkg/ha", "SEDPkg/ha", "NSURQkg/ha", "NLATQkg/ha", "NO3Lkg/ha", "NO3GWkg/ha", "SOLPkg/ha", "P\_GWkg/ha", "W\_STRS", "TMP\_STRS", "N\_STRS", "P\_STRS", "BIOMt/ha", "LAI", "YLDt/ha", "BACTPct", "BACTL\_Pct", "WTAB CLIm", "WTAB SOLm", "SNOmm", "CMUPkg/ha", "CMTOTkg/ha", "QTILEmm", "TNO3kg/ha", "LNO3kg/ha", "GW\_Q\_Dmm", "LATQCNTmm", "TVAPkg/ha"/)  
*column headers for HRU output file*
- character(len=13), dimension(msubo), parameter [parm::hedb](#) = (/ "PRECIPmm", "SNOMELTmm", "PETmm", "ETmm", "SWmm", "PERCmm", "SURQmm", "GW\_Qmm", "WYLDmm", "SYLDt/ha", "ORGNkg/ha", "ORGPkg/ha", "NSURQkg/ha", "SOLPkg/ha", "SEDPkg/ha", "LAT Q(mm)", "LATNO3kg/h", "GWN\_O3kg/ha", "CHOLAmic/L", "CBODU mg/L", "DOXQ mg/L", "TNO3kg/ha", "QTILEmm", "TVAPkg/ha"/)  
*column headers for subbasin output file*



*Surface runoff lag time. This parameter is needed in subbasins where the time of concentration is greater than 1 day. SURLAG is used to create a "storage" for surface runoff to allow the runoff to take longer than 1 day to reach the subbasin outlet (days)*

- real \*8, dimension(:), allocatable **parm::co\_p**
- real \*8, dimension(:), allocatable **parm::cmn**  
*rate factor for humus mineralization on active organic N*
- real \*8, dimension(:), allocatable **parm::phoskd**  
*phosphorus soil partitioning coefficient. Ratio of soluble phosphorus in surface layer attached to sediment to phosphorus dissolved in soil water*
- real \*8, dimension(:), allocatable **parm::psp**  
*phosphorus availability index. The fraction of fertilizer P remaining in labile pool after initial rapid phase of P sorption (none)*
- real \*8, dimension(:), allocatable **parm::sdnco**  
*denitrification threshold: fraction of field capacity triggering denitrification*
- real \*8 **parm::r2adj\_bsn**  
*basinwide retention parameter adjustment factor (greater than 1)*
- real \*8 **parm::pst\_kg**  
*amount of pesticide applied to HRU (kg/ha)*
- real \*8 **parm::yield**  
*yield (dry weight) (kg)*
- real \*8 **parm::burn\_frlb**  
*fraction of biomass and residue that burn(input in management file) range (0 - 1.0) (none)*
- real \*8 **parm::yieldgrn**
- real \*8 **parm::yieldbms**
- real \*8 **parm::yieldtbr**
- real \*8 **parm::yieldn**
- real \*8 **parm::yieldp**
- real \*8 **parm::hi\_bms**
- real \*8 **parm::hi\_rsd**
- real \*8 **parm::yieldrsd**
- real \*8, dimension(:), allocatable **parm::l\_k1**
- real \*8, dimension(:), allocatable **parm::l\_k2**
- real \*8, dimension(:), allocatable **parm::l\_lambda**
- real \*8, dimension(:), allocatable **parm::l\_beta**
- real \*8, dimension(:), allocatable **parm::l\_gama**
- real \*8, dimension(:), allocatable **parm::l\_harea**
- real \*8, dimension(:), allocatable **parm::l\_vleng**
- real \*8, dimension(:), allocatable **parm::l\_vslope**
- real \*8, dimension(:), allocatable **parm::l\_ktc**
- real \*8, dimension(:), allocatable **parm::biofilm\_mumax**
- real \*8, dimension(:), allocatable **parm::biofilm\_kinv**
- real \*8, dimension(:), allocatable **parm::biofilm\_klw**
- real \*8, dimension(:), allocatable **parm::biofilm\_kla**
- real \*8, dimension(:), allocatable **parm::biofilm\_cdet**
- real \*8, dimension(:), allocatable **parm::biofilm\_bm**
- real \*8, dimension(:,:), allocatable **parm::hru\_rufr**
- real \*8, dimension(:,:), allocatable **parm::daru\_km**
- real \*8, dimension(:,:), allocatable **parm::ru\_k**
- real \*8, dimension(:,:), allocatable **parm::ru\_c**
- real \*8, dimension(:,:), allocatable **parm::ru\_eiq**
- real \*8, dimension(:,:), allocatable **parm::ru\_ovsl**
- real \*8, dimension(:,:), allocatable **parm::ru\_a**
- real \*8, dimension(:,:), allocatable **parm::ru\_ovs**
- real \*8, dimension(:,:), allocatable **parm::ru\_ktc**

- real \*8, dimension(:), allocatable **parm::gwq\_ru**
- real \*8, dimension(:), allocatable **parm::qdayout**
- integer, dimension(:), allocatable **parm::ils2**
- integer, dimension(:), allocatable **parm::ils2flag**
- integer **parm::ipest**  
*pesticide identification number from pest.dat (none)*
- integer **parm::iru**
- integer **parm::mru**
- integer **parm::irch**
- integer **parm::isub**
- integer **parm::mhyd\_bsn**
- integer **parm::ils\_nofig**
- integer **parm::mhru1**
- integer, dimension(:), allocatable **parm::mhyd1**
- integer, dimension(:), allocatable **parm::irtun**
- real \*8 **parm::wshd\_sepno3**
- real \*8 **parm::wshd\_sepnh3**
- real \*8 **parm::wshd\_seporgn**
- real \*8 **parm::wshd\_sepfon**
- real \*8 **parm::wshd\_seporgp**
- real \*8 **parm::wshd\_sepfop**
- real \*8 **parm::wshd\_sepsolp**
- real \*8 **parm::wshd\_sepbod**
- real \*8 **parm::wshd\_sepmm**
- integer, dimension(:), allocatable **parm::isep\_hru**
- real \*8 **parm::fixco**  
*nitrogen fixation coefficient*
- real \*8 **parm::nfixmx**  
*maximum daily n-fixation (kg/ha)*
- real \*8 **parm::res\_stlr\_co**  
*reservoir sediment settling coefficient*
- real \*8 **parm::rsd\_covco**  
*residue cover factor for computing fraction of cover*
- real \*8 **parm::vcrit**  
*critical velocity*
- real \*8 **parm::wshd\_snob**  
*average amount of water stored in snow at the beginning of the simulation for the entire watershed (mm H2O)*
- real \*8 **parm::wshd\_sw**  
*water in soil at beginning of simulation, or average amount of water stored in soil for the entire watershed, or difference between mass balance calculated from watershed averages and actual value for water in soil at end of simulation (goal is to have wshd\_sw = 0.) (mm H2O)*
- real \*8 **parm::wshd\_pndfr**  
*fraction of watershed area which drains into ponds (none)*
- real \*8 **parm::wshd\_pndsed**  
*total amount of suspended sediment in ponds in the watershed (metric tons), or mass balance discrepancy for pond sediment expressed as loading per unit hectare of drainage area (metric tons/ha)*
- real \*8 **parm::wshd\_pndv**  
*total volume of water in ponds in the watershed ( $m^3$ ), or mass balance discrepancy for pond water volume expressed as depth over drainage area (mm H2O)*
- real \*8 **parm::percop**  
*pesticide percolation coefficient (0-1)*  
*0: concentration of pesticide in surface runoff is zero*  
*1: percolate has same concentration of pesticide as surface runoff*

- real \*8 `parm::wshd_resfr`  
*fraction of watershed area that drains into reservoirs (none)*
- real \*8 `parm::wshd_pndha`  
*watershed area in hectares which drains into ponds (ha)*
- real \*8 `parm::wshd_resha`  
*watershed area in hectares which drains into reservoirs (ha)*
- real \*8 `parm::wshd_wetfr`  
*fraction of watershed area which drains into wetlands (none)*
- real \*8 `parm::wshd_fminp`  
*average annual amount of mineral P applied in watershed (kg P/ha)*
- real \*8 `parm::wshd_fnh3`  
*average annual amount of NH<sub>3</sub>-N applied in watershed (kg N/ha)*
- real \*8 `parm::wshd_fno3`  
*average annual amount of NO<sub>3</sub>-N applied in watershed (kg N/ha)*
- real \*8 `parm::wshd_forgn`  
*average annual amount of organic N applied in watershed (kg N/ha)*
- real \*8 `parm::wshd_ftotn`  
*average annual amount of N (mineral & organic) applied in watershed (kg N/ha)*
- real \*8 `parm::wshd_forgp`  
*average annual amount of organic P applied in watershed (kg P/ha)*
- real \*8 `parm::wshd_ftotp`  
*average annual amount of P (mineral & organic) applied in watershed (kg P/ha)*
- real \*8 `parm::wshd_yldn`  
*amount of nitrogen removed from soil in watershed in the yield (kg N/ha)*
- real \*8 `parm::wshd_yldp`  
*amount of phosphorus removed from soil in watershed in the yield (kg P/ha)*
- real \*8 `parm::wshd_fixn`  
*average annual amount of nitrogen added to plant biomass via fixation (kg N/ha)*
- real \*8 `parm::wshd_pup`  
*average annual amount of plant uptake of phosphorus (kg P/ha)*
- real \*8 `parm::wshd_nstrs`  
*average annual number of nitrogen stress units in watershed (stress units)*
- real \*8 `parm::wshd_pstrs`  
*average annual number of phosphorus stress units in watershed (stress units)*
- real \*8 `parm::wshd_tstrs`  
*average annual number of temperature stress units in watershed (stress units)*
- real \*8 `parm::wshd_wstrs`  
*average annual number of water stress units in watershed (stress units)*
- real \*8 **`parm::wshd_astrs`**
- real \*8 `parm::ffcb`  
*initial soil water content expressed as a fraction of field capacity*
- real \*8 `parm::wshd_dnit`  
*average annual amount of nitrogen lost from nitrate pool due to denitrification in watershed (kg N/ha)*
- real \*8 `parm::wshd_hmn`  
*average annual amount of nitrogen moving from active organic to nitrate pool in watershed (kg N/ha)*
- real \*8 `parm::wshd_hmp`  
*average annual amount of phosphorus moving from organic to labile pool in watershed (kg P/ha)*
- real \*8 `parm::wshd_rmn`  
*average annual amount of nitrogen moving from fresh organic (residue) to nitrate and active organic pools in watershed (kg N/ha)*
- real \*8 `parm::wshd_rwn`

- average annual amount of nitrogen moving from active organic to stable organic pool in watershed (kg N/ha)*

  - real \*8 **parm::wdpq**
- die-off factor for persistent bacteria in soil solution (1/day)*

  - real \*8 **parm::wshd\_rmp**
- average annual amount of phosphorus moving from fresh organic (residue) to labile and organic pools in watershed (kg P/ha)*

  - real \*8 **parm::wshd\_nitn**
- average annual amount of nitrogen moving from the NH3 to the NO3 pool by nitrification in the watershed (kg N/ha)*

  - real \*8 **parm::wshd\_voin**
- average annual amount of nitrogen lost by ammonia volatilization in watershed (kg N/ha)*

  - real \*8 **parm::wshd\_pal**
- average annual amount of phosphorus moving from labile mineral to active mineral pool in watershed (kg P/ha)*

  - real \*8 **parm::wshd\_pas**
- average annual amount of phosphorus moving from active mineral to stable mineral pool in watershed (kg P/ha)*

  - real \*8 **parm::wof\_p**
- fraction of persistent bacteria on foliage that is washed off by a rainfall event (none)*

  - real \*8 **parm::wshd\_raino3**
- average annual amount of NO3 added to soil by rainfall in watershed (kg N/ha)*

  - real \*8 **parm::wshd\_plch**
- average annual amount of phosphorus leached into second soil layer (kg P/ha)*

  - real \*8 **parm::ressedc**
- real \*8 **parm::basno3f**
- real \*8 **parm::basorgnf**
- real \*8 **parm::wshd\_pinlet**
- real \*8 **parm::wshd\_ptile**
- real \*8 **parm::sftmp**

*Snowfall temperature (deg C)*
- real \*8 **parm::smfmn**

*Minimum melt rate for snow during year (Dec. 21) where deg C refers to the air temperature. (mm/deg C/day)*
- real \*8 **parm::smfmx**

*Maximum melt rate for snow during year (June 21) where deg C refers to the air temperature. SMFMX and SMFMN allow the rate of snow melt to vary through the year. These parameters are accounting for the impact of soil temperature on snow melt. (mm/deg C/day)*
- real \*8 **parm::smtmp**

*Snow melt base temperature. Mean air temperature at which snow melt will occur. (deg C)*
- real \*8 **parm::wgpq**

*growth factor for persistent bacteria in soil solution (1/day)*
- real \*8 **parm::basminpf**
- real \*8 **parm::basorgpf**
- real \*8 **parm::wdlpq**

*die-off factor for less persistent bacteria in soil solution (1/day)*
- real \*8 **parm::wshd\_ressed**

*total amount of suspended sediment in reservoirs in the watershed (metric tons),  
or mass balance discrepancy for reservoir sediment expressed as loading per unit hectare of drainage area (metric tons/ha)*
- real \*8 **parm::wshd\_resv**

*total volume of water in all reservoirs in the watershed (m<sup>3</sup>),  
or mass balance discrepancy for reservoir water volume expressed as depth over drainage area (mm H2O)*
- real \*8 **parm::basminpi**

*average amount of phosphorus initially in the mineral P pool in watershed soil (kg P/ha)*
- real \*8 **parm::basno3i**

*average amount of nitrogen initially in the nitrate pool in watershed soil (kg N/ha)*

- real \*8 **parm::basorgni**  
*average amount of nitrogen initially in the organic N pool in watershed soil (kg N/ha)*
- real \*8 **parm::wdps**  
*die-off factor for persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **parm::wglpq**  
*growth factor for less persistent bacteria in soil solution (1/day)*
- real \*8 **parm::basorgpi**  
*average amount of phosphorus initially in the organic P pool in watershed soil (kg P/ha)*
- real \*8 **parm::peakr**  
*peak runoff rate for the day in HRU ( $m^3/s$ )*
- real \*8 **parm::albdy**  
*albedo of ground for the day in HRU, the fraction of the solar radiation reflected at the soil surface back into space (none)*
- real \*8 **parm::pndsedin**  
*sediment inflow to the pond from HRU (metric tons)*
- real \*8 **parm::sw\_excess**  
*amount of water stored in soil layer on the current day that exceeds field capacity (gravity drained water) (mm H<sub>2</sub>O)*
- real \*8 **parm::timp**  
*Snow pack temperature lag factor (0-1)  
1 = no lag (snow pack temp=current day air temp) as the lag factor goes to zero, the snow pack's temperature will be less influenced by the current day's air temperature.*
- real \*8 **parm::wt\_shall**  
*shallow water table depth above the impervious layer (mm H<sub>2</sub>O)*
- real \*8 **parm::wtabelo**
- real \*8 **parm::tilep**
- real \*8 **parm::sq\_rto**
- real \*8 **parm::qtile**  
*drainage tile flow in HRU soil layer for the day (mm H<sub>2</sub>O)*
- real \*8 **parm::infiltrpcp**  
*amount of precipitation that infiltrates into soil (enters soil) (mm H<sub>2</sub>O)*
- real \*8 **parm::crk**  
*percolation due to crack flow (mm H<sub>2</sub>O)*
- real \*8 **parm::fixn**  
*amount of nitrogen added to the plant biomass via fixation on the day in HRU (kg N/ha)*
- real \*8 **parm::latlyr**  
*amount of water in lateral flow in layer in HRU for the day (mm H<sub>2</sub>O)*
- real \*8 **parm::snofall**  
*amount of precipitation falling as freezing rain/snow on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **parm::snomlt**  
*amount of water in snow melt for the day in HRU (mm H<sub>2</sub>O)*
- real \*8 **parm::tloss**  
*amount of water removed from surface runoff via transmission losses on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **parm::pndloss**
- real \*8 **parm::wetloss**
- real \*8 **parm::potloss**
- real \*8 **parm::lpndloss**
- real \*8 **parm::lwetloss**
- real \*8 **parm::bioday**  
*biomass generated on current day in HRU (kg)*
- real \*8 **parm::cfertn**  
*total amount of nitrogen applied to soil during continuous fertilizer operation in HRU on day (kg N/ha)*
- real \*8 **parm::cfertp**

- amount of phosphorus applied to soil during continuous fertilizer operation in HRU on day (kg P/ha)*

  - real \*8 **parm::fertn**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::sepday**
- micropore percolation from bottom of the soil layer on day in HRU (mm H<sub>2</sub>O)*

  - real \*8 **parm::sol\_rd**
- current rooting depth (mm)*

  - real \*8 **parm::sedrch**
- water entering cracks in soil (mm H<sub>2</sub>O)*

  - real \*8 **parm::sepcrk**
- water entering cracks in soil (mm H<sub>2</sub>O)*

  - real \*8 **parm::sepcrktot**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::fertno3**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::fertnh3**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::fertorgn**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::fertsolp**
- total amount of nitrogen applied to soil in HRU on day (kg N/ha)*

  - real \*8 **parm::fertorgp**
- growth factor for persistent bacteria adsorbed to soil particles (1/day)*

  - real \*8 **parm::wgps**
- fraction of water yield that is surface runoff (none)*

  - real \*8 **parm::qdfr**
- total amount of phosphorus applied to soil in HRU on day (kg P/ha)*

  - real \*8 **parm::fertp**
- amount of nitrogen added to soil in grazing on the day in HRU (kg N/ha)*

  - real \*8 **parm::grazn**
- amount of phosphorus added to soil in grazing on the day in HRU (kg P/ha)*

  - real \*8 **parm::grazp**
- saturation dissolved oxygen concentration (mg/L)*

  - real \*8 **parm::soxy**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::sditi**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::rtwtr**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::ressa**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::wdlps**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::da\_km**
- area of the watershed in square kilometers (km<sup>2</sup>)*

  - real \*8 **parm::rttime**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::rchdep**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::rtevp**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::rttlc**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::resflwi**
- die-off factor for less persistent bacteria absorbed to soil particles (1/day)*

  - real \*8 **parm::wdprch**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::resflwo**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::respcp**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::resev**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::ressep**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::ressedi**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::ressedo**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::dtot**
- die-off factor for persistent bacteria in streams (1/day)*

  - real \*8 **parm::pperco\_bsn**



- phosphorus percolation coefficient. Ratio of soluble phosphorus in surface to soluble phosphorus in percolate*

  - real \*8 **parm::nperco\_bsn**  
*basin nitrate percolation coefficient (0-1)*  
*0:concentration of nitrate in surface runoff is zero*  
*1:percolate has same concentration of nitrate as surface runoff*
- real \*8 **parm::rsdco**  
*residue decomposition coefficient. The fraction of residue which will decompose in a day assuming optimal moisture, temperature, C:N ratio, and C:P ratio*
- real \*8 **parm::voltot**  
*total volume of cracks expressed as depth per unit area (mm)*
- real \*8 **parm::phoskd\_bsn**
- real \*8 **parm::msk\_x**  
*weighting factor controlling relative importance of inflow rate and outflow rate in determining storage on reach*
- real \*8 **parm::volcrmin**  
*minimum crack volume allowed in any soil layer (mm), or  
minimum soil volume in profile (mm)*
- real \*8 **parm::bactkdq**  
*bacteria soil partitioning coefficient. Ratio of solution bacteria in surface layer to solution bacteria in runoff soluble and sorbed phase in surface runoff.*
- real \*8 **parm::wdpf**  
*die-off factor for persistent bacteria on foliage (1/day)*
- real \*8 **parm::canev**  
*amount of water evaporated from canopy storage (mm H2O)*
- real \*8 **parm::precipday**  
*precipitation, or effective precipitation reaching soil surface, for the current day in HRU (mm H2O)*
- real \*8 **parm::uno3d**  
*plant nitrogen deficiency for day in HRU (kg N/ha)*
- real \*8 **parm::usle**  
*daily soil loss predicted with USLE equation (metric tons/ha)*
- real \*8 **parm::rcn**  
*concentration of nitrogen in the rainfall (mg/L)*
- real \*8 **parm::surlag\_bsn**
- real \*8 **parm::thbact**  
*temperature adjustment factor for bacteria die-off/growth*
- real \*8 **parm::wlpq20**  
*overall rate change for less persistent bacteria in soil solution (1/day)*
- real \*8 **parm::wlps20**  
*overall rate change for less persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **parm::wpq20**  
*overall rate change for persistent bacteria in soil solution (1/day)*
- real \*8 **parm::wps20**  
*overall rate change for persistent bacteria adsorbed to soil particles (1/day)*
- real \*8 **parm::bactrop**  
*persistent bacteria transported to main channel with surface runoff (# colonies/ha)*
- real \*8 **parm::bactsedp**  
*persistent bacteria transported with sediment in surface runoff (# colonies/ha)*
- real \*8 **parm::wgpf**  
*growth factor for persistent bacteria on foliage (1/day)*
- real \*8 **parm::bactlchp**  
*less persistent bacteria removed from soil surface layer by percolation (# colonies/ha)*
- real \*8 **parm::bactlchp**  
*persistent bacteria removed from soil surface layer by percolation (# colonies/ha)*

- real \*8 **parm::enratio**  
*enrichment ratio calculated for current day in HRU (none)*
- real \*8 **parm::pndpcp**  
*precipitation on pond during day ( $m^3 H_2O$ )*
- real \*8 **parm::wetpcp**
- real \*8 **parm::wetsep**  
*seepage from wetland bottom for day ( $m^3 H_2O$ )*
- real \*8 **parm::pndev**  
*evaporation from pond on day ( $m^3 H_2O$ )*
- real \*8 **parm::pndflwi**  
*volume of water flowing into pond on day ( $m^3 H_2O$ )*
- real \*8 **parm::pndsedo**  
*sediment leaving pond during day (metric tons)*
- real \*8 **parm::pndsep**  
*seepage from pond on day ( $m^3 H_2O$ )*
- real \*8 **parm::wetev**  
*evaporation from wetland for day ( $m^3 H_2O$ )*
- real \*8 **parm::wetflwi**  
*volume of water flowing in wetland on day ( $m^3 H_2O$ )*
- real \*8 **parm::wetsedo**  
*sediment loading from wetland for day (metric tons)*
- real \*8 **parm::da\_ha**  
*drainage area of watershed in hectares (ha)*
- real \*8 **parm::pndflwo**  
*volume of water flowing out of pond on day ( $m^3 H_2O$ )*
- real \*8 **parm::vpd**  
*vapor pressure deficit (kPa)*
- real \*8 **parm::wetflwo**  
*volume of water flowing out wetland on day ( $m^3 H_2O$ )*
- real \*8 **parm::wetsedi**  
*sediment loading to wetland for day (metric tons)*
- real \*8 **parm::evlai**  
*leaf area index at which no evaporation occurs. This variable is used in ponded HRUs where evaporation from the water surface is restricted by the plant canopy cover. Evaporation from the water surface equals potential ET when LAI = 0 and decreased linearly to 0 when LAI = EVLAI*
- real \*8 **parm::evrch**  
*Reach evaporation adjustment factor. Evaporation from the reach is multiplied by EVRCH. This variable was created to limit the evaporation predicted in arid regions.*
- real \*8 **parm::wdlpf**  
*die-off factor for less persistent bacteria on foliage (1/day)*
- real \*8 **parm::ep\_day**  
*actual amount of transpiration that occurs on day in HRU (mm H<sub>2</sub>O)*
- real \*8 **parm::pet\_day**  
*potential evapotranspiration on current day in HRU (mm H<sub>2</sub>O)*
- real \*8 **parm::bactrolp**  
*less persistent bacteria transported to main channel with surface runoff (# colonies/ha)*
- real \*8 **parm::bactsedlp**  
*less persistent bacteria transported with sediment in surface runoff (# colonies/ha)*
- real \*8 **parm::adj\_pkr**  
*peak rate adjustment factor in the subbasin. Used in the MUSLE equation to account for impact of peak flow on erosion (none)*

- real \*8 **parm::n\_updis**  
nitrogen uptake distribution parameter. This parameter controls the amount of nitrogen removed from the different soil layer layers by the plant. In particular, this parameter allows the amount of nitrogen removed from the surface layer via plant uptake to be controlled. While the relationship between UBN and N removed from the surface layer is affected by the depth of the soil profile, in general, as UBN increases the amount of N removed from the surface layer relative to the amount removed from the entire profile increases
- real \*8 **parm::nactfr**  
nitrogen active pool fraction. The fraction of organic nitrogen in the active pool (none)
- real \*8 **parm::p\_updis**  
phosphorus uptake distribution parameter This parameter controls the amount of phosphorus removed from the different soil layers by the plant. In particular, this parameter allows the amount of phosphorus removed from the surface layer via plant uptake to be controlled. While the relationship between UBP and P uptake from the surface layer is affected by the depth of the soil profile, in general, as UBP increases the amount of P removed from the surface layer relative to the amount removed from the entire profile increases
- real \*8 **parm::snoev**  
amount of water in snow lost through sublimation on current day in HRU (mm H2O)
- real \*8 **parm::sno3up**  
amount of nitrate moving upward in the soil profile in watershed (kg N/ha)
- real \*8 **parm::reactw**
- real \*8 **parm::es\_day**  
actual amount of evaporation (soil et) that occurs on day in HRU (mm H2O)
- real \*8 **parm::sdiegro1pq**  
average annual change in the number of less persistent bacteria colonies in soil solution in watershed (# cfu/m<sup>2</sup>)
- real \*8 **parm::sdiegro1ps**  
average annual change in the number of less persistent bacteria colonies on soil particles in watershed (# cfu/m<sup>2</sup>)
- real \*8 **parm::sdiegro2pq**  
average annual change in the number of persistent bacteria colonies in soil solution in watershed (# cfu/m<sup>2</sup>)
- real \*8 **parm::sdiegro2ps**  
average annual change in the number of persistent bacteria colonies on soil particles in watershed (# cfu/m<sup>2</sup>)
- real \*8 **parm::wof\_lp**  
fraction for less persistent bacteria on foliage that is washed off by a rainfall event (none)
- real \*8 **parm::ep\_max**  
maximum amount of transpiration (plant et) that can occur on day in HRU (mm H2O)
- real \*8 **parm::sbactrop**
- real \*8 **parm::sbactrolp**
- real \*8 **parm::sbactsedp**
- real \*8 **parm::sbactsedlp**
- real \*8 **parm::sbactlchlp**  
average annual number of less persistent bacteria lost from soil surface layer by percolation (# cfu/m<sup>2</sup>)
- real \*8 **parm::sbactlchp**  
average annual number of persistent bacteria lost from soil surface layer by percolation (# cfu/m<sup>2</sup>)
- real \*8 **parm::psp\_bsn**
- real \*8 **parm::rchwtr**
- real \*8 **parm::resuspst**
- real \*8 **parm::setlpst**
- real \*8 **parm::bsprev**  
surface runoff lagged from prior day of simulation (mm H2O)
- real \*8 **parm::bssprev**  
lateral flow lagged from prior day of simulation (mm H2O)
- real \*8 **parm::spadyo**
- real \*8 **parm::spadyev**
- real \*8 **parm::spadyisp**
- real \*8 **parm::spadyrfv**

- real \*8 **parm::spadyosp**
- real \*8 **parm::qday**  
amount of surface runoff loading to main channel from HRU on current day (mm H2O)
- real \*8 **parm::al5**  
fraction of total rainfall that occurs during 0.5h of highest intensity rain (none)
- real \*8 **parm::no3pcp**  
nitrate added to the soil in rainfall (kg N/ha)
- real \*8 **parm::pndsedc**  
net change in sediment in pond during day (metric tons)
- real \*8 **parm::usle\_ei**  
USLE rainfall erosion index on day for HRU (100(ft·tn in)/(acre·hr))
- real \*8 **parm::rcharea**
- real \*8 **parm::volatpst**
- real \*8 **parm::ubw**  
water uptake distribution parameter. This parameter controls the amount of water removed from the different soil layers by the plant. In particular, this parameter allows the amount of water removed from the surface layer via plant uptake to be controlled. While the relationship between UBW and H2O removed from the surface layer is affected by the depth of the soil profile, in general, as UBW increases the amount of water removed from the surface layer relative to the amount removed from the entire profile increases
- real \*8 **parm::uobn**  
nitrogen uptake normalization parameter. This variable normalizes the nitrogen uptake so that the model can easily verify that uptake from the different soil layers sums to 1.0
- real \*8 **parm::uobp**  
phosphorus uptake normalization parameter. This variable normalizes the phosphorus uptake so that the model can easily verify that uptake from the different soil layers sums to 1.0
- real \*8 **parm::uobw**  
water uptake normalization parameter. This variable normalizes the water uptake so that the model can easily verify that uptake from the different soil layers sums to 1.0
- real \*8 **parm::wglpf**  
growth factor for less persistent bacteria on foliage (1/day)
- real \*8 **parm::wetsedc**  
net change in sediment in wetland during day (metric tons)
- real \*8 **parm::respesti**
- real \*8 **parm::rcor**  
correction coefficient for generated rainfall to ensure that the annual means for generated and observed values are comparable (needed only if IDIST=1)
- real \*8 **parm::rexp**  
value of exponent for mixed exponential rainfall distribution (needed only if IDIST=1)
- real \*8 **parm::snocov1**  
1st shape parameter for snow cover equation. This parameter is determined by solving the equation for 50% snow cover
- real \*8 **parm::snocov2**  
2nd shape parameter for snow cover equation. This parameter is determined by solving the equation for 95% snow cover
- real \*8 **parm::snocovmx**  
Minimum snow water content that corresponds to 100% snow cover. If the snow water content is less than SNOC← OVMX, then a certain percentage of the ground will be bare (mm H2O)
- real \*8 **parm::lyrtile**  
drainage tile flow in soil layer for day in HRU (mm H2O)
- real \*8 **parm::lyrtilex**
- real \*8 **parm::sno50cov**  
Fraction of SNOCOVMX that corresponds to 50% snow cover. SWAT assumes a nonlinear relationship between snow water and snow cover.

- real \*8 `parm::ai0`  
*ratio of chlorophyll-a to algal biomass (ug chla/mg alg)*
- real \*8 `parm::ai1`  
*fraction of algal biomass that is nitrogen (mg N/mg alg)*
- real \*8 `parm::ai2`  
*fraction of algal biomass that is phosphorus (mg P/mg alg)*
- real \*8 `parm::ai3`  
*the rate of oxygen production per unit of algal photosynthesis (mg O2/mg alg)*
- real \*8 `parm::ai4`  
*the rate of oxygen uptake per unit of algae respiration (mg O2/mg alg)*
- real \*8 `parm::ai5`  
*the rate of oxygen uptake per unit of NH3 nitrogen oxidation (mg O2/mg N)*
- real \*8 `parm::ai6`  
*the rate of oxygen uptake per unit of NO2 nitrogen oxidation (mg O2/mg N)*
- real \*8 `parm::rhoq`  
*algal respiration rate (1/day or 1/hr)*
- real \*8 `parm::tfact`  
*fraction of solar radiation computed in the temperature heat balance that is photosynthetically active*
- real \*8 `parm::k_l`  
*half-saturation coefficient for light (MJ/(m2\*hr))*
- real \*8 `parm::k_n`  
*michaelis-menton half-saturation constant for nitrogen (mg N/L)*
- real \*8 `parm::k_p`  
*michaelis-menton half saturation constant for phosphorus (mg P/L)*
- real \*8 `parm::lambda0`  
*non-algal portion of the light extinction coefficient (1/m)*
- real \*8 `parm::lambda1`  
*linear algal self-shading coefficient (1/(m\*ug chla/L))*
- real \*8 `parm::lambda2`  
*nonlinear algal self-shading coefficient ((1/m)(ug chla/L)\*\*(-2/3))*
- real \*8 `parm::mumax`  
*maximum specific algal growth rate (1/day or 1/hr)*
- real \*8 `parm::p_n`  
*algal preference factor for ammonia*
- real \*8 `parm::rnum1`  
*variable to hold value for rnum1s(:) (none)*
- real \*8 `parm::etday`  
*actual evapotranspiration occurring on day in HRU (mm H2O)*
- real \*8 `parm::auton`  
*amount of nitrogen applied in auto-fert application (kg N/ha)*
- real \*8 `parm::autop`  
*amount of phosphorus applied in auto-fert application (kg P/ha)*
- real \*8 `parm::hmntl`  
*amount of nitrogen moving from active organic to nitrate pool in soil profile on current day in HRU (kg N/ha)*
- real \*8 `parm::hmptl`  
*amount of phosphorus moving from active organic to nitrate pool in soil profile on current day in HRU (kg P/ha)*
- real \*8 `parm::rmn2tl`  
*amount of nitrogen moving from the fresh organic (residue) to the nitrate(80%) and active organic(20%) pools in soil profile on current day in HRU (kg N/ha)*
- real \*8 `parm::rwntl`  
*amount of nitrogen moving from active organic to stable organic pool in soil profile on current day in HRU (kg N/ha)*

- real \*8 **parm::gwseep**  
amount of water recharging deep aquifer on current day in HRU (mm H2O)
- real \*8 **parm::revapday**  
amount of water moving from the shallow aquifer into the soil profile or being taken up by plant roots in the shallow aquifer (mm H2O)
- real \*8 **parm::rmp1tl**  
amount of phosphorus moving from the labile mineral pool to the active mineral pool in the soil profile on the current day in the HRU (kg P/ha)
- real \*8 **parm::rmp1tl**  
amount of phosphorus moving from the fresh organic (residue) to the labile(80%) and organic(20%) pools in soil profile on current day in HRU (kg P/ha)
- real \*8 **parm::roct1**  
amount of phosphorus moving from the active mineral pool to the stable mineral pool in the soil profile on the current day in the HRU (kg P/ha)
- real \*8 **parm::wdntl**  
amount of nitrogen lost from nitrate pool by denitrification in soil profile on current day in HRU (kg N/ha)
- real \*8 **parm::cmn\_bsn**
- real \*8 **parm::reswtr**
- real \*8 **parm::wdlprch**  
die-off factor for less persistent bacteria in streams (1/day)
- real \*8 **parm::wdpres**  
die-off factor for persistent bacteria in reservoirs (1/day)
- real \*8 **parm::petmeas**  
potential ET value read in for day (mm H2O)
- real \*8 **parm::bury**
- real \*8 **parm::difus**
- real \*8 **parm::reactb**
- real \*8 **parm::solpesto**
- real \*8 **parm::wdlpres**  
die-off factor for less persistent bacteria in reservoirs (1/day)
- real \*8 **parm::sorpesto**
- real \*8 **parm::spcon\_bsn**
- real \*8 **parm::spexp\_bsn**
- real \*8 **parm::solpesti**
- real \*8 **parm::sorpesti**
- real \*8 **parm::msk\_co1**  
calibration coefficient to control impact of the storage time constant for the reach at bankfull depth ( $\phi(10, \cdot)$ ) upon the storage time constant for the reach used in the Muskingum flow method
- real \*8 **parm::msk\_co2**  
calibration coefficient to control impact of the storage time constant for the reach at 0.1 bankfull depth ( $\phi(13, \cdot)$ ) upon the storage time constant for the reach used in the Muskingum flow method
- real \*8 **parm::deepstp**  
depth of water in deep aquifer in HRU (mm H2O)
- real \*8 **parm::shallstp**  
depth of water in shallow aquifer in HRU on previous day (mm H2O)
- real \*8 **parm::snoprev**  
amount of water stored as snow on previous day (mm H2O)
- real \*8 **parm::swprev**  
amount of water stored in soil profile in the HRU on the previous day (mm H2O)
- real \*8 **parm::ressolpo**
- real \*8 **parm::resorgno**
- real \*8 **parm::resorgpo**
- real \*8 **parm::resno3o**

- real \*8 **parm::reschlao**
- real \*8 **parm::resno2o**
- real \*8 **parm::potevmm**  
*volume of water evaporated from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **parm::potflwo**  
*volume of water released to main channel from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **parm::potpcpmm**  
*precipitation falling on pothole water body expressed as depth over HRU (mm H2O)*
- real \*8 **parm::potsepmmm**  
*seepage from pothole expressed as depth over HRU (mm H2O)*
- real \*8 **parm::resnh3o**
- real \*8 **parm::qdbank**
- real \*8 **parm::bactminlp**  
*Threshold detection level for less persistent bacteria. When bacteria levels drop to this amount the model considers bacteria in the soil to be insignificant and sets the levels to zero (cfu/m<sup>2</sup>)*
- real \*8 **parm::bactminp**  
*Threshold detection level for persistent bacteria. When bacteria levels drop to this amount the model considers bacteria in the soil to be insignificant and sets the levels to zero (cfu/m<sup>2</sup>)*
- real \*8 **parm::trnsrch**  
*fraction of transmission losses from main channel that enter deep aquifer*
- real \*8 **parm::wp20p\_plt**  
*overall rate change for persistent bacteria on foliage (1/day)*
- real \*8 **parm::potsedo**  
*sediment released to main channel from HRU (metric tons/ha)*
- real \*8 **parm::pest\_sol**
- real \*8 **parm::bact\_swf**  
*fraction of manure containing active colony forming units (cfu)*
- real \*8 **parm::bactmx**  
*bacteria percolation coefficient. Ratio of solution bacteria in surface layer to solution bacteria in percolate*
- real \*8 **parm::cncoef**  
*plant ET curve number coefficient*
- real \*8 **parm::wp20lp\_plt**  
*overall rate change for less persistent bacteria on foliage (1/day)*
- real \*8 **parm::cdn\_bsn**
- real \*8 **parm::sdnco\_bsn**
- real \*8 **parm::bactmin**
- real \*8 **parm::cn\_froz**  
*drainage coefficient (mm day<sup>-1</sup>)*
- real \*8 **parm::dorm\_hr**  
*time threshold used to define dormant (hours)*
- real \*8 **parm::smxco**  
*adjustment factor for max curve number s factor (0-1)*
- real \*8 **parm::tb\_adj**  
*adjustment factor for subdaily unit hydrograph basetime*
- real \*8 **parm::chla\_subco**  
*regional adjustment on sub chla\_a loading (fraction)*
- real \*8 **parm::depimp\_bsn**  
*depth to impervious layer. Used to model perched water tables in all HRUs in watershed (mm)*
- real \*8 **parm::ddrain\_bsn**  
*depth to the sub-surface drain (mm)*
- real \*8 **parm::tdrain\_bsn**  
*time to drain soil to field capacity (hours)*

- real \*8 **parm::gdrain\_bsn**
- real \*8 **parm::rch\_san**
- real \*8 **parm::rch\_sil**
- real \*8 **parm::rch\_cla**
- real \*8 **parm::rch\_sag**
- real \*8 **parm::rch\_lag**
- real \*8 **parm::rch\_gra**
- real \*8 **parm::hlife\_ngw\_bsn**  
*Half-life of nitrogen in groundwater? (days)*
- real \*8 **parm::ch\_opco\_bsn**
- real \*8 **parm::ch\_onco\_bsn**
- real \*8 **parm::decr\_min**  
*Minimum daily residue decay.*
- real \*8 **parm::rcn\_sub\_bsn**  
*Concentration of nitrogen in the rainfall (mg/kg)*
- real \*8 **parm::bc1\_bsn**
- real \*8 **parm::bc2\_bsn**
- real \*8 **parm::bc3\_bsn**
- real \*8 **parm::bc4\_bsn**
- real \*8 **parm::anion\_excl\_bsn**
- real \*8, dimension(:), allocatable **parm::wat\_tbl**  
*water table based on depth from soil surface (mm)*
- real \*8, dimension(:), allocatable **parm::sol\_swpwt**
- real \*8, dimension(:,:), allocatable **parm::vwt**
- real \*8 **parm::re\_bsn**  
*Effective radius of drains (range 3.0 - 40.0) (mm)*
- real \*8 **parm::sdrain\_bsn**  
*Distance bewtween two drain or tile tubes (range 7600.0 - 30000.0) (mm)*
- real \*8 **parm::sstmaxd\_bsn**
- real \*8 **parm::drain\_co\_bsn**  
*Drainage coefficient (range 10.0 - 51.0) (mm-day-1)*
- real \*8 **parm::latksatf\_bsn**  
*Multiplication factor to determine lateral ksat from SWAT ksat input value for HRU (range 0.01 - 4.0)*
- real \*8 **parm::pc\_bsn**  
*Pump capacity (def val = 1.042 mm h-1 or 25 mm day-1) (mm h-1)*
- integer **parm::idlast**  
*number of days simulated in month (none)*
- integer **parm::i\_subhw**
- integer **parm::imgt**
- integer **parm::iwtr**
- integer **parm::ifrttyp**
- integer **parm::mo\_atmo**
- integer **parm::mo\_atmo1**
- integer **parm::ifirstatmo**
- integer **parm::iyr\_atmo**
- integer **parm::iyr\_atmo1**
- integer **parm::matmo**
- integer **parm::mch**  
*maximum number of channels*
- integer **parm::mcr**  
*maximum number of crops grown per year*
- integer **parm::mcldb**



- maximum number of crops/landcover in database file (crop.dat)*
- integer **parm::mfctst**
  - maximum number of forecast stations*
- integer **parm::mfdb**
  - maximum number of fertilizers in fert.dat*
- integer **parm::mhru**
  - maximum number of HRUs in watershed*
- integer **parm::mhyd**
  - maximum number of hydrograph nodes*
- integer **parm::mpdb**
  - maximum number of pesticides in pest.dat*
- integer **parm::mrg**
  - maximum number of rainfall/temp gages (none)*
- integer **parm::mcut**
  - maximum number of cuttings per year*
- integer **parm::mgr**
  - maximum number of grazings per year*
- integer **parm::mnr**
  - maximum number of years of rotation*
- integer **parm::myr**
  - maximum number of years of simulation*
- integer **parm::isubwq**
  - subbasin water quality code*
  - 0 do not calculate algae/CBOD 1 calculate algae/CBOD drainmod tile equations*
- integer **parm::ffcst**
- integer **parm::isproj**
  - special project code (none):*
  - 1 test rewind (run simulation twice)*
- integer **parm::nbyr**
  - number of calendar years simulated (none)*
- integer **parm::irte**
  - water routing method (none):*
  - 0 variable storage method*
  - 1 Muskingum method*
- integer **parm::nrch**
  - number of reaches in watershed (none)*
- integer **parm::nres**
  - total number of reservoirs in watershed (none)*
- integer **parm::nhru**
  - number of last HRU in previous subbasin or*
  - number of HRUs in watershed (none)*
- integer **parm::i\_mo**
  - current month being simulated or month of next day of simulation (none)*
- integer **parm::immo**
  - current cumulative month of simulation (none)*
- integer **parm::mo**
- integer **parm::wndsim**
  - wind speed input code (noen)*
  - 1 measured data read for each subbasin*
  - 2 data simulated for each subbasin*
- integer **parm::ihru**

- HRU number (none)
- integer **parm::icode**
  - variable to hold value for icode(:) (none)
- integer **parm::ihout**
  - variable to hold value for ihouts(:) (none)
- integer **parm::inum1**
  - variable to hold value for inum1s(:) (subbasin number) (none)
- integer **parm::inum2**
  - variable to hold value for inum2s(:) (none)
- integer **parm::inum3**
  - variable to hold value for inum3s(:) (none)
- integer **parm::inum4**
  - variable to hold value for inum4s(:) (none)
- integer **parm::icfac**
  - icfac = 0 for C-factor calculation using Cmin (as described in manual)
  - = 1 for new C-factor calculation from RUSLE (no minimum needed)
- integer **parm::inum5**
- integer **parm::inum6**
- integer **parm::inum7**
- integer **parm::inum8**
- integer **parm::mrech**
  - maximum number of rehour files
- integer **parm::nrgage**
  - number of raingage files (none)
- integer **parm::nrgfil**
  - number of rain gages per file (none)
- integer **parm::nrtot**
  - total number of rain gages (none)
- integer **parm::ntgage**
  - number of temperature gage files (none)
- integer **parm::ntgfil**
  - number of temperature gages per file (none)
- integer **parm::nttot**
  - total number of temperature gages (none)
- integer **parm::tmpsim**
  - temperature input code (none)
  - 1 measured data read for each subbasin
  - 2 data simulated for each subbasin
- integer **parm::icrk**
  - crack flow code
  - 1: simulate crack flow in watershed
- integer **parm::irtpest**
  - number of pesticide to be routed through the watershed. Redefined to the sequence number of pesticide in NPNO(:) which is to be routed through the watershed (none)
- integer **parm::igropt**
  - Qual2E option for calculating the local specific growth rate of algae
  - 1: multiplicative.
- integer **parm::lao**
  - Qual2E light averaging option. Qual2E defines four light averaging options. The only option currently available in SWAT is #2.
- integer **parm::npmx**
  - number of different pesticides used in the simulation (none)

- integer [parm::curyr](#)  
current year in simulation (sequence) (none)
- integer [parm::itdrn](#)  
tile drainage equations flag/code  
1 simulate tile flow using subroutine drains(wt\_shall)  
0 simulate tile flow using subroutine origtile(wt\_shall,d)
- integer [parm::iwtdn](#)  
water table depth algorithms flag/code  
1 simulate wt\_shall using subroutine new water table depth routine  
0 simulate wt\_shall using subroutine original water table depth routine
- integer [parm::ismax](#)  
maximum depressional storage selection flag/code (none)  
0 = static depressional storage (stmaxd) read from .bsn for the global value or .sdr for specific HRUs  
1 = dynamic storage (stmaxd) based on random roughness, tillage and cumulative rainfall intensity by [depstor.f90](#)
- integer [parm::iroutunit](#)  
not being implemented in this version drainmod tile equations
- integer [parm::ires\\_nut](#)
- integer [parm::iclb](#)  
auto-calibration flag
- integer [parm::mrecc](#)  
maximum number of reccnst files
- integer [parm::mrecd](#)  
maximum number of recday files
- integer [parm::mrecm](#)  
maximum number of recmon files
- integer [parm::mtill](#)  
max number of tillage types in till.dat
- integer [parm::mudb](#)  
maximum number of urban land types in urban.dat
- integer [parm::idist](#)  
rainfall distribution code  
0 for skewed normal dist  
1 for mixed exponential distribution
- integer [parm::mrecy](#)  
maximum number of recyear files
- integer [parm::nyskip](#)  
number of years to skip output summarization and printing (none)
- integer [parm::slrsim](#)  
solar radiation input code (none)  
1 measured data read for each subbasin  
2 data simulated for each subbasin
- integer [parm::ideg](#)  
channel degradation code  
0: do not compute channel degradation  
1: compute channel degradation (downcutting and widening)
- integer [parm::ievent](#)  
rainfall/runoff code (none)  
0 daily rainfall/curve number technique 1 sub-daily rainfall/Green&Ampt/hourly routing 3 sub-daily rainfall/↔  
Green&Ampt/hourly routing
- integer [parm::ipet](#)  
code for potential ET method (none)  
0 Priestley-Taylor method  
1 Penman/Monteith method  
2 Hargreaves method  
3 read in daily potential ET data

- integer **parm::iopera**
- integer **parm::idaf**  
*beginning day of simulation (julian date)*
- integer **parm::idal**  
*ending day of simulation (julian date)*
- integer **parm::rhsim**  
*relative humidity input code (none)*  
*1 measured data read for each subbasin*  
*2 data simulated for each subbasin*
- integer **parm::leapyr**  
*leap year flag (none)*  
*0 leap year*  
*1 regular year*
- integer **parm::id1**  
*first day of simulation in current year (julian date)*
- integer **parm::mo\_chk**  
*current month of simulation (none)*
- integer **parm::nhtot**  
*total number of relative humidity records in file*
- integer **parm::nstot**  
*total number of solar radiation records in file (none)*
- integer **parm::nwtot**  
*total number of wind speed records in file*
- integer **parm::ifirsts**  
*solar radiation data search code (none)*  
*0 first day of solar radiation data located in file*  
*1 first day of solar radiation data not located in file*
- integer **parm::ifirsth**  
*relative humidity data search code (none)*  
*0 first day of relative humidity data located in file*  
*1 first day of relative humidity data not located in file*
- integer **parm::ifirstw**  
*wind speed data search code (none)*  
*0 first day of wind speed data located in file*  
*1 first day of wind speed data not located in file*
- integer **parm::icst**
- integer **parm::ilog**  
*streamflow print code (none)*  
*0 print streamflow in reach*  
*1 print Log10 streamflow in reach*
- integer **parm::itotr**  
*number of output variables printed (output.rch)*
- integer **parm::iy**  
*current year of simulation (year)*
- integer **parm::iwq**  
*stream water quality code*  
*0 do not model stream water quality*  
*1 model stream water quality (QUAL2E & pesticide transformations)*
- integer **parm::iskip**  
*flag for calculations performed only for the first year of simulation (none)*
- integer **parm::ifirstpet**  
*potential ET data search code (none)*  
*0 first day of potential ET data located in file*  
*1 first day of potential ET data not located in file*

- integer **parm::iprp**  
*print code for output.pst file*  
*0 do not print pesticide output*  
*1 print pesticide output*
- integer **parm::itotb**  
*number of output variables printed (output.sub)*
- integer **parm::itots**  
*number of output variables printed (output.hru)*
- integer **parm::itoth**  
*number of HRUs printed (output.hru/output.wtr)*
- integer **parm::pcpsim**  
*rainfall input code (none)*  
*1 measured data read for each subbasin*  
*2 data simulated for each subbasin*
- integer **parm::nd\_30**
- integer **parm::iops**
- integer **parm::iphr**
- integer **parm::isto**
- integer **parm::isol**
- integer **parm::fcstcycles**  
*number of times forecast period is simulated (using different weather generator seeds each time)*
- integer **parm::fcstday**  
*beginning date of forecast period (julian date)*
- integer **parm::fcstyr**  
*beginning year of forecast period*
- integer **parm::iscen**  
*scenarios counter*
- integer **parm::subtot**  
*number of subbasins in watershed (none)*
- integer **parm::ogen**
- integer **parm::mapp**  
*maximum number of applications*
- integer **parm::mlyr**  
*maximum number of soil layers*
- integer **parm::mpst**  
*max number of pesticides used in wshed*
- integer **parm::mres**  
*maximum number of reservoirs*
- integer **parm::msub**  
*maximum number of subbasins*
- integer **parm::igen**  
*random number generator seed code (none):*  
*0: use default numbers*  
*1: generate new numbers in every simulation*
- integer **parm::iprint**  
*print code (none): 0=monthly, 1=daily, 2=annually*
- integer **parm::iida**  
*day being simulated (current julian date) (julian date)*
- integer **parm::icn**  
*CN method flag (for testing alternative method):*  
*0 use traditional SWAT method which bases CN on soil moisture*  
*1 use alternative method which bases CN on plant ET*  
*2 use traditional SWAT method which bases CN on soil moisture but retention is adjusted for mildly-sloped tiled-drained watersheds.*

- integer **parm::ised\_det**  
*max half-hour rainfall fraction calc option:*  
*0 generate max half-hour rainfall fraction from triangular distribution*  
*1 use monthly mean max half-hour rainfall fraction*
- integer **parm::fcstcnt**
- integer **parm::mtran**
- integer **parm::idtill**
- integer, dimension(100) **parm::ida\_lup**
- integer, dimension(100) **parm::iyр\_lup**
- integer **parm::no\_lup**
- integer **parm::no\_up**
- integer **parm::nostep**
- character(len=8) **parm::date**  
*date simulation is performed where leftmost eight characters are set to a value of yyyymmdd, where yyyy is the year, mm is the month and dd is the day*
- character(len=10) **parm::time**  
*time simulation is performed where leftmost ten characters are set to a value of hhmmss.sss, where hh is the hour, mm is the minutes and ss.sss is the seconds and milliseconds*
- character(len=5) **parm::zone**  
*time difference with respect to Coordinated Universal Time (ie Greenwich Mean Time)*
- character(len=13) **parm::calfile**  
*name of file containing calibration parameters*
- character(len=13) **parm::rhfile**  
*relative humidity file name (.hmd)*
- character(len=13) **parm::slrfile**  
*solar radiation file name (.slr)*
- character(len=13) **parm::wndfile**  
*wind speed file name (.wnd)*
- character(len=13) **parm::petfile**  
*potential ET file name (.pet)*
- character(len=13) **parm::atmofile**
- character(len=13) **parm::lucfile**
- character(len=13) **parm::septdb**  
*name of septic tank database file (septwq1.dat)*
- character(len=13) **parm::dpd\_file**
- character(len=13) **parm::wpd\_file**
- character(len=13) **parm::rib\_file**
- character(len=13) **parm::sfb\_file**
- character(len=13) **parm::lid\_file**
- integer, dimension(9) **parm::idg**  
*array location of random number seed used for a given process*
- integer, dimension(:), allocatable **parm::ifirstr**
- integer, dimension(:), allocatable **parm::ifirsthr**
- integer, dimension(8) **parm::values**  
*values(1): year simulation is performed*  
*values(2): month simulation is performed*  
*values(3): day in month simulation is performed*  
*values(4): time difference with respect to Coordinated Universal Time (ie Greenwich Mean Time)*  
*values(5): hour simulation is performed*  
*values(6): minute simulation is performed*  
*values(7): second simulation is performed*  
*values(8): millisecond simulation is performed*
- integer, dimension(13) **parm::ndays**

*julian date for last day of preceding month (where the array location is the number of the month). The dates are for leap years (julian date)*

- integer **parm::mapex**
- real \*8, dimension(:), allocatable **parm::flodaya**
- real \*8, dimension(:), allocatable **parm::seddaya**
- real \*8, dimension(:), allocatable **parm::orgndaya**
- real \*8, dimension(:), allocatable **parm::orgpdaya**
- real \*8, dimension(:), allocatable **parm::no3daya**
- real \*8, dimension(:), allocatable **parm::minpdaya**
- real \*8, dimension(:), allocatable **parm::hi\_targ**  
*harvest index target of cover defined at planting ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable **parm::bio\_targ**  
*biomass target (kg/ha)*
- real \*8, dimension(:), allocatable **parm::tnyld**  
*modifier for autofertilization target nitrogen content for plant (kg N/kg yield)*
- integer, dimension(:), allocatable **parm::idapa**
- integer, dimension(:), allocatable **parm::iypa**
- integer, dimension(:), allocatable **parm::ifirsta**
- integer, dimension(100) **parm::mo\_transb**
- integer, dimension(100) **parm::mo\_transe**
- integer, dimension(100) **parm::ih\_tran**
- integer **parm::msdb**  
*maximum number of sept wq data database (none)*
- integer **parm::iseptic**
- real \*8, dimension(:), allocatable **parm::sptqs**  
*flow rate of the septic tank effluent per capita (m3/d)*
- real \*8, dimension(:), allocatable **parm::percp**
- real \*8, dimension(:), allocatable **parm::sptbodconcs**  
*Biological Oxygen Demand of the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::spttssconcs**  
*concentration of total suspended solid in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptnconcs**  
*concentration of total nitrogen in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptnh4concs**  
*concentration of total phosphorus of the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptno3concs**  
*concentration of nitrate in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptno2concs**  
*concentration of nitrite in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptorgnconcs**  
*concentration of organic nitrogen in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::spttpconcs**  
*concentration of total phosphorus in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptminps**  
*concentration of mineral phosphorus in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptorgps**  
*concentration of organic phosphorus in the septic tank effluent (mg/l)*
- real \*8, dimension(:), allocatable **parm::sptfcolis**  
*concentration of the facel caliform in the septic tank effluent (cfu/100ml)*
- real \*8, dimension(:), allocatable **parm::failyr**
- real \*8, dimension(:), allocatable **parm::qstemm**
- real \*8, dimension(:), allocatable **parm::bio\_bod**

- *BOD concentration in biozone (kg/ha)*
- real \*8, dimension(:), allocatable [parm::biom](#)
- *biomass of live bacteria in biozone (kg/ha)*
- real \*8, dimension(:), allocatable [parm::rbiom](#)
- *daily change in biomass of live bacteria (kg/ha)*
- real \*8, dimension(:), allocatable [parm::bio\\_amn](#)
- real \*8, dimension(:), allocatable [parm::fcoli](#)
- *concentration of the fecal coliform in the biozone septic tank effluent (cfu/100ml)*
- real \*8, dimension(:), allocatable [parm::bio\\_ntr](#)
- real \*8, dimension(:), allocatable [parm::bz\\_perc](#)
- real \*8, dimension(:), allocatable [parm::sep\\_cap](#)
- *number of permanent residents in the house (none)*
- real \*8, dimension(:), allocatable [parm::plqm](#)
- *plaque in biozone (kg/ha)*
- real \*8, dimension(:), allocatable [parm::bz\\_area](#)
- real \*8, dimension(:), allocatable [parm::bz\\_z](#)
- *depth of biozone layer (mm)*
- real \*8, dimension(:), allocatable [parm::bz\\_thk](#)
- *thickness of biozone (mm)*
- real \*8, dimension(:), allocatable [parm::bio\\_bd](#)
- *density of biomass (kg/m<sup>3</sup>)*
- real \*8, dimension(:), allocatable [parm::cmup\\_kgh](#)
- *current soil carbon for first soil layer (kg/ha)*
- real \*8, dimension(:), allocatable [parm::cmtot\\_kgh](#)
- *current soil carbon integrated - aggregating (kg/ha)*
- real \*8, dimension(:), allocatable [parm::coeff\\_denitr](#)
- *denitrification rate coefficient (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_bod\\_dc](#)
- *BOD decay rate coefficient (m<sup>3</sup>/day)*
- real \*8, dimension(:), allocatable [parm::coeff\\_bod\\_conv](#)
- *BOD to live bacteria biomass conversion factor (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_fc1](#)
- *field capacity calibration parameter 1 (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_fc2](#)
- *field capacity calibration parameter 2 (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_fecal](#)
- *fecal coliform bacteria decay rate coefficient (m<sup>3</sup>/day)*
- real \*8, dimension(:), allocatable [parm::coeff\\_mrt](#)
- *mortality rate coefficient (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_nitr](#)
- *nitrification rate coefficient (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_plq](#)
- *conversion factor for plaque from TDS (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_rsp](#)
- *respiration rate coefficient (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_slg1](#)
- *slough-off calibration parameter (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_slg2](#)
- *slough-off calibration parameter (none)*
- real \*8, dimension(:), allocatable [parm::coeff\\_pdistrb](#)
- real \*8, dimension(:), allocatable [parm::coeff\\_solpslp](#)



- real \*8, dimension(:), allocatable **parm::coeff\_solpintc**
- real \*8, dimension(:), allocatable **parm::coeff\_psorpmx**
- integer, dimension(:), allocatable **parm::isep\_typ**  
*septic system type (none)*
- integer, dimension(:), allocatable **parm::i\_sep**  
*soil layer where biozone exists (none)*
- integer, dimension(:), allocatable **parm::isep\_opt**  
*septic system operation flag (1=active, 2=failing, 3 or 0=not operated) (none)*
- integer, dimension(:), allocatable **parm::sep\_tsincefail**
- integer, dimension(:), allocatable **parm::isep\_tfail**
- integer, dimension(:), allocatable **parm::isep\_iyr**
- integer, dimension(:), allocatable **parm::sep\_strm\_dist**
- integer, dimension(:), allocatable **parm::sep\_den**
- real \*8, dimension(:), allocatable **parm::sol\_sumno3**
- real \*8, dimension(:), allocatable **parm::sol\_sumsolp**
- real \*8, dimension(:), allocatable **parm::strsw\_sum**
- real \*8, dimension(:), allocatable **parm::strstmp\_sum**
- real \*8, dimension(:), allocatable **parm::strsn\_sum**
- real \*8, dimension(:), allocatable **parm::strsp\_sum**
- real \*8, dimension(:), allocatable **parm::strsa\_sum**
- real \*8, dimension(:), allocatable **parm::spill\_hru**
- real \*8, dimension(:), allocatable **parm::tile\_out**
- real \*8, dimension(:), allocatable **parm::hru\_in**
- real \*8, dimension(:), allocatable **parm::spill\_precip**
- real \*8, dimension(:), allocatable **parm::pot\_seep**
- real \*8, dimension(:), allocatable **parm::pot\_evap**
- real \*8, dimension(:), allocatable **parm::pot\_sedin**
- real \*8, dimension(:), allocatable **parm::pot\_solp**  
*soluble P loss rate in the pothole (.01 - 0.5) (1/d)*
- real \*8, dimension(:), allocatable **parm::pot\_solpi**
- real \*8, dimension(:), allocatable **parm::pot\_orgp**
- real \*8, dimension(:), allocatable **parm::pot\_orgpi**
- real \*8, dimension(:), allocatable **parm::pot\_orgn**
- real \*8, dimension(:), allocatable **parm::pot\_orgni**
- real \*8, dimension(:), allocatable **parm::pot\_mps**
- real \*8, dimension(:), allocatable **parm::pot\_mpsi**
- real \*8, dimension(:), allocatable **parm::pot\_mpa**
- real \*8, dimension(:), allocatable **parm::pot\_mpai**
- real \*8, dimension(:), allocatable **parm::pot\_no3i**
- real \*8, dimension(:), allocatable **parm::precip\_in**
- real \*8, dimension(:), allocatable **parm::tile\_sedo**
- real \*8, dimension(:), allocatable **parm::tile\_no3o**
- real \*8, dimension(:), allocatable **parm::tile\_solpo**
- real \*8, dimension(:), allocatable **parm::tile\_orgno**
- real \*8, dimension(:), allocatable **parm::tile\_orgpo**
- real \*8, dimension(:), allocatable **parm::tile\_minpso**
- real \*8, dimension(:), allocatable **parm::tile\_minpao**
- integer **parm::ia\_b**
- integer **parm::ihumus**
- integer **parm::itemp**
- integer **parm::isnow**
- integer, dimension(46) **parm::ipdvar**  
*output variable codes for output.rch file (none)*
- integer, dimension(mhruo) **parm::ipdvas**

- output variable codes for output.hru file (none)*
- integer, dimension(msubo) [parm::ipdvab](#)
  - output variable codes for output.sub file (none)*
- integer, dimension(:), allocatable [parm::ipdhru](#)
  - HRUs whose output information will be printed to the output.hru and output.wtr files.*
- real \*8, dimension(mstdo) [parm::wshddayo](#)
  - wshddayo(1) average amount of precipitation in watershed for the day (mm H2O)*
  - wshddayo(3) surface runoff in watershed for day (mm H2O)*
  - wshddayo(4) lateral flow contribution to streamflow in watershed for day (mm H2O)*
  - wshddayo(5) water percolation past bottom of soil profile in watershed for day (mm H2O)*
  - wshddayo(6) water yield to streamflow from HRUs in watershed for day (mm H2O)*
  - wshddayo(7) actual evapotranspiration in watershed for day (mm H2O)*
  - wshddayo(12) sediment yield from HRUs in watershed for day (metric tons or metric tons/ha)*
  - wshddayo(35) amount of water stored in soil profile in watershed at end of day (mm H2O)*
  - wshddayo(41) organic P loading to stream in watershed for day (kg P/ha)*
  - wshddayo(42) nitrate loading to stream in surface runoff in watershed for day (kg N/ha)*
  - wshddayo(43) soluble P loading to stream in watershed for day (kg P/ha)*
  - wshddayo(44) plant uptake of N in watershed for day (kg N/ha)*
  - wshddayo(45) nitrate loading to stream in lateral flow in watershed for day (kg N/ha)*
  - wshddayo(46) nitrate percolation past bottom of soil profile in watershed for day (kg N/ha)*
  - wshddayo(104) groundwater contribution to stream in watershed on day (mm H2O)*
  - wshddayo(108) potential evapotranspiration in watershed on day (mm H2O)*
  - wshddayo(109) drainage tile flow contribution to stream in watershed on day (mm H2O)*
- real \*8, dimension(mstdo) [parm::wshdmono](#)
  - watershed monthly output array (see definitions for wshddayo array elements) (varies)*
  - wshdmono(1) average amount of precipitation in watershed for the month (mm H2O)*
  - wshdmono(3) surface runoff in watershed for month (mm H2O)*
  - wshdmono(4) lateral flow contribution to streamflow in watershed for month (mm H2O)*
  - wshdmono(5) water percolation past bottom of soil profile in watershed for month (mm H2O)*
  - wshdmono(6) water yield to streamflow from HRUs in watershed for month (mm H2O)*
  - wshdmono(7) actual evapotranspiration in watershed for month (mm H2O)*
  - wshdmono(8) average maximum temperature in watershed for the month (deg C)*
  - wshdmono(9) average minimum temperature in watershed for the month (deg C)*
  - wshdmono(12) sediment yield from HRUs in watershed for the month (metric tons)*
  - wshdmono(39) freezing rain/snow fall in watershed for the month (mm H2O)*
  - wshdmono(40) organic N loading to stream in watershed for the month (kg N/ha)*
  - wshdmono(41) organic P loading to stream in watershed for the month (kg P/ha)*
  - wshdmono(42) nitrate loading to stream in surface runoff in watershed for the month (kg N/ha)*
  - wshdmono(43) soluble P loading to stream in watershed for the month (kg P/ha)*
  - wshdmono(44) plant uptake of N in watershed for the month (kg N/ha)*
  - wshdmono(45) nitrate loading to stream in lateral flow in watershed for the month (kg N/ha)*
  - wshdmono(46) nitrate percolation past bottom of soil profile in watershed for the month (kg N/ha)*
  - wshdmono(104) groundwater contribution to stream in watershed for the month (mm H2O)*
  - wshdmono(108) potential evapotranspiration in watershed for the month (mm H2O)*
  - wshdmono(109) drainage tile flow contribution to stream in watershed for the month (mm H2O)*
- real \*8, dimension(mstdo) [parm::wshdyro](#)
  - watershed annual output array (varies)*
  - wshdyro(1) average amount of precipitation in watershed for the year (mm H2O)*
  - wshdyro(3) surface runoff in watershed for year (mm H2O)*
  - wshdyro(4) lateral flow contribution to streamflow in watershed for year (mm H2O)*
  - wshdyro(5) water percolation past bottom of soil profile in watershed for year (mm H2O)*
  - wshdyro(6) water yield to streamflow from HRUs in watershed for year (mm H2O)*
  - wshdyro(7) actual evapotranspiration in watershed for year (mm H2O)*
  - wshdyro(8) average maximum temperature in watershed for the year (deg C)*
  - wshdyro(9) average minimum temperature in watershed for the year (deg C)*
  - wshdyro(12) sediment yield from HRUs in watershed for the year (metric tons)*
  - wshdyro(40) organic N loading to stream in watershed for the year (kg N/ha)*
  - wshdyro(41) organic P loading to stream in watershed for the year (kg P/ha)*
  - wshdyro(42) nitrate loading to stream in surface runoff in watershed for the year (kg N/ha)*
  - wshdyro(43) soluble P loading to stream in watershed for the year (kg P/ha)*
  - wshdyro(44) plant uptake of N in watershed for the year*

- wshdyro(45) nitrate loading to stream in lateral flow in watershed for the year (kg N/ha)*
- wshdyro(46) nitrate percolation past bottom of soil profile in watershed for the year (kg N/ha)*
- wshdyro(104) groundwater contribution to stream in watershed for the year (mm H2O)*
- wshdyro(108) potential evapotranspiration in watershed for the year (mm H2O)*
- wshdyro(109) drainage tile flow contribution to stream in watershed for the year (mm H2O)*
- real \*8, dimension(16) **parm::fcstaa0**
- real \*8, dimension(mstdo) **parm::wshdaao**
  - watershed average annual output array (varies)*
  - wshdaao(1) precipitation in watershed (mm H2O)*
  - wshdaao(3) surface runoff loading to main channel in watershed (mm H2O)*
  - wshdaao(4) lateral flow loading to main channel in watershed (mm H2O)*
  - wshdaao(5) percolation of water out of root zone in watershed (mm H2O)*
  - wshdaao(7) actual evapotranspiration in watershed (mm H2O)*
  - wshdaao(13) sediment loading to ponds in watershed (metric tons)*
  - wshdaao(14) sediment loading from ponds in watershed (metric tons)*
  - wshdaao(15) net change in sediment level in ponds in watershed (metric tons)*
  - wshdaao(19) evaporation from ponds in watershed ( $m^3$  H2O)*
  - wshdaao(20) seepage from ponds in watershed ( $m^3$  H2O)*
  - wshdaao(21) precipitation on ponds in watershed ( $m^3$  H2O)*
  - wshdaao(22) volume of water entering ponds in watershed ( $m^3$  H2O)*
  - wshdaao(23) volume of water leaving ponds in watershed ( $m^3$  H2O)*
  - wshdaao(38) transmission losses in watershed (mm H2O)*
- real \*8, dimension(:, :), allocatable **parm::wpstdayo**
- real \*8, dimension(:, :), allocatable **parm::wpstmono**
- real \*8, dimension(:, :), allocatable **parm::wpstyro**
- real \*8, dimension(:, :), allocatable **parm::bio\_hv**
  - harvested biomass (dry weight) (kg/ha)*
- real \*8, dimension(:, :), allocatable **parm::yldkg**
  - yield (dry weight) by crop type in the HRU (kg/ha)*
- real \*8, dimension(:, :), allocatable **parm::rchmono**
  - reach monthly output array (varies)*
  - rchmono(1,:) flow into reach during month ( $m^3$ /s)*
  - rchmono(2,:) flow out of reach during month ( $m^3$ /s)*
  - rchmono(3,:) sediment transported into reach during month (metric tons)*
  - rchmono(4,:) sediment transported out of reach during month (metric tons)*
  - rchmono(5,:) sediment concentration in outflow during month (mg/L)*
  - rchmono(6,:) organic N transported into reach during month (kg N)*
  - rchmono(7,:) organic N transported out of reach during month (kg N)*
  - rchmono(8,:) organic P transported into reach during month (kg P)*
  - rchmono(9,:) organic P transported out of reach during month (kg P)*
  - rchmono(10,:) evaporation from reach during month ( $m^3$ /s)*
  - rchmono(11,:) transmission losses from reach during month ( $m^3$ /s)*
  - rchmono(12,:) conservative metal #1 transported out of reach during month (kg)*
  - rchmono(13,:) conservative metal #2 transported out of reach during month (kg)*
  - rchmono(14,:) conservative metal #3 transported out of reach during month (kg)*
  - rchmono(15,:) nitrate transported into reach during month (kg N)*
  - rchmono(16,:) nitrate transported out of reach during month (kg N)*
  - rchmono(17,:) soluble P transported into reach during month (kg P)*
  - rchmono(18,:) soluble P transported out of reach during month (kg P)*
  - rchmono(19,:) soluble pesticide transported into reach during month (mg pst)*
  - rchmono(20,:) soluble pesticide transported out of reach during month (mg pst)*
  - rchmono(21,:) sorbed pesticide transported into reach during month (mg pst)*
  - rchmono(22,:) sorbed pesticide transported out of reach during month (mg pst)*
  - rchmono(23,:) amount of pesticide lost through reactions in reach during month (mg pst)*
  - rchmono(24,:) amount of pesticide lost through volatilization from reach during month (mg pst)*
  - rchmono(25,:) amount of pesticide settling out of reach to bed sediment during month (mg pst)*
  - rchmono(26,:) amount of pesticide resuspended from bed sediment to reach during month (mg pst)*
  - rchmono(27,:) amount of pesticide diffusing from reach to bed sediment during month (mg pst)*
  - rchmono(28,:) amount of pesticide in sediment layer lost through reactions during month (mg pst)*
  - rchmono(29,:) amount of pesticide in sediment layer lost through burial during month (mg pst)*
  - rchmono(30,:) chlorophyll-a transported into reach during month (kg chl a)*

- rchmono(31,:)* chlorophyll-a transported out of reach during month (kg chla)
- rchmono(32,:)* ammonia transported into reach during month (kg N)
- rchmono(33,:)* ammonia transported out of reach during month (kg N)
- rchmono(34,:)* nitrite transported into reach during month (kg N)
- rchmono(35,:)* nitrite transported out of reach during month (kg N)
- rchmono(36,:)* CBOD transported into reach during month (kg O2)
- rchmono(37,:)* CBOD transported out of reach during month (kg O2)
- rchmono(38,:)* dissolved oxygen transported into reach during month (kg O2)
- rchmono(39,:)* dissolved oxygen transported out of reach during month (kg O2)
- rchmono(40,:)* persistent bacteria transported out of reach during month (kg bact)
- rchmono(41,:)* less persistent bacteria transported out of reach during month (kg bact)
- rchmono(43,:)* total N (org N + no3 + no2 + nh4 outs) (kg)
- rchmono(44,:)* total P (org P + sol p outs) (kg)
- real \*8, dimension(:,:), allocatable **parm::rchyro**
  - reach* annual output array (varies)
  - rchyro(1,:)* flow into reach during year ( $m^3/s$ )
  - rchyro(2,:)* flow out of reach during year ( $m^3/s$ )
  - rchyro(3,:)* sediment transported into reach during year (metric tons)
  - rchyro(4,:)* sediment transported out of reach during year (metric tons)
  - rchyro(5,:)* sediment concentration in outflow during year (mg/L)
  - rchyro(6,:)* organic N transported into reach during year (kg N)
  - rchyro(7,:)* organic N transported out of reach during year (kg N)
  - rchyro(8,:)* organic P transported into reach during year (kg P)
  - rchyro(9,:)* organic P transported out of reach during year (kg P)
  - rchyro(10,:)* evaporation from reach during year ( $m^3/s$ )
  - rchyro(11,:)* transmission losses from reach during year ( $m^3/s$ )
  - rchyro(12,:)* conservative metal #1 transported out of reach during year (kg)
  - rchyro(13,:)* conservative metal #2 transported out of reach during year (kg)
  - rchyro(14,:)* conservative metal #3 transported out of reach during year (kg)
  - rchyro(15,:)* nitrate transported into reach during year (kg N)
  - rchyro(16,:)* nitrate transported out of reach during year (kg N)
  - rchyro(17,:)* soluble P transported into reach during year (kg P)
  - rchyro(18,:)* soluble P transported out of reach during year (kg P)
  - rchyro(19,:)* soluble pesticide transported into reach during year (mg pst)
  - rchyro(20,:)* soluble pesticide transported out of reach during year (mg pst)
  - rchyro(21,:)* sorbed pesticide transported into reach during year (mg pst)
  - rchyro(22,:)* sorbed pesticide transported out of reach during year (mg pst)
  - rchyro(23,:)* amount of pesticide lost through reactions in reach during year!> (mg pst)
  - rchyro(24,:)* amount of pesticide lost through volatilization from reach during year (mg pst)
  - rchyro(25,:)* amount of pesticide settling out of reach to bed sediment during year (mg pst)
  - rchyro(26,:)* amount of pesticide resuspended from bed sediment to reach during year (mg pst)
  - rchyro(27,:)* amount of pesticide diffusing from reach to bed sediment during year (mg pst)
  - rchyro(28,:)* amount of pesticide in sediment layer lost through reactions during year (mg pst)
  - rchyro(29,:)* amount of pesticide in sediment layer lost through burial during year (mg pst)
  - rchyro(30,:)* chlorophyll-a transported into reach during year (kg chla)
  - rchyro(31,:)* chlorophyll-a transported out of reach during year (kg chla)
  - rchyro(32,:)* ammonia transported into reach during year (kg N)
  - rchyro(33,:)* ammonia transported out of reach during year (kg N)
  - rchyro(34,:)* nitrite transported into reach during year (kg N)
  - rchyro(35,:)* nitrite transported out of reach during year (kg N)
  - rchyro(36,:)* CBOD transported into reach during year (kg O2)
  - rchyro(37,:)* CBOD transported out of reach during year (kg O2)
  - rchyro(38,:)* dissolved oxygen transported into reach during year (kg O2)
  - rchyro(39,:)* dissolved oxygen transported out of reach during year (kg O2)
  - rchyro(40,:)* persistent bacteria transported out of reach during year (kg bact)
  - rchyro(41,:)* less persistent bacteria transported out of reach during year (kg bact)
- real \*8, dimension(:,:), allocatable **parm::wpstaa0**
- real \*8, dimension(:,:), allocatable **parm::hrumono**
  - HRU* monthly output data array (varies)
  - hrumono(1,:)* precipitation in HRU during month (mm H2O)
  - hrumono(2,:)* amount of precipitation falling as freezing rain/snow in HRU during month (mm H2O)
  - hrumono(3,:)* amount of snow melt in HRU during month (mm H2O)
  - hrumono(4,:)* amount of surface runoff to main channel from HRU during month (ignores impact of transmission)

losses) (mm H2O)  
 humono(5,:) amount of lateral flow contribution to main channel from HRU during month (mm H2O)  
 humono(6,:) amount of groundwater flow contribution to main channel from HRU during month (mm H2O)  
 humono(7,:) amount of water moving from shallow aquifer to plants or soil profile in HRU during month (mm H2O)  
 humono(8,:) amount of water recharging deep aquifer in HRU during month (mm H2O)  
 humono(9,:) total amount of water entering both aquifers from HRU during month (mm H2O)  
 humono(10,:) water yield (total amount of water entering main channel) from HRU during month (mm H2O)  
 humono(11,:) amount of water percolating out of the soil profile and into the vadose zone in HRU during month (mm H2O)  
 humono(12,:) actual evapotranspiration in HRU during month (mm H2O)  
 humono(13,:) amount of transmission losses from tributary channels in HRU for month (mm H2O)  
 humono(14,:) sediment yield from HRU for month (metric tons/ha)  
 humono(15,:) actual amount of transpiration that occurs during month in HRU (mm H2O)  
 humono(16,:) actual amount of evaporation (from soil) that occurs during month in HRU (mm H2O)  
 humono(17,:) amount of nitrogen applied in continuous fertilizer operation during month in HRU (kg N/ha)  
 humono(18,:) amount of phosphorus applied in continuous fertilizer operation during month in HRU (kg P/ha)  
 humono(22,:) amount of irrigation water applied to HRU during month (mm H2O)  
 humono(23,:) amount of water removed from shallow aquifer in HRU for irrigation during month (mm H2O)  
 humono(24,:) amount of water removed from deep aquifer in HRU for irrigation during month (mm H2O)  
 humono(25,:) potential evapotranspiration in HRU during month (mm H2O)  
 humono(26,:) monthly amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)  
 humono(27,:) monthly amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)  
 humono(28,:) monthly amount of N (organic & mineral) auto-applied in HRU (kg N/ha)  
 humono(29,:) monthly amount of P (organic & mineral) auto-applied in HRU (kg P/ha)  
 humono(31,:) water stress days in HRU during month (stress days)  
 humono(32,:) temperature stress days in HRU during month (stress days)  
 humono(33,:) nitrogen stress days in HRU during month (stress days)  
 humono(34,:) phosphorus stress days in HRU during month (stress days)  
 humono(35,:) organic nitrogen in surface runoff in HRU during month (kg N/ha)  
 humono(36,:) organic phosphorus in surface runoff in HRU during month (kg P/ha)  
 humono(37,:) nitrate in surface runoff in HRU during month (kg N/ha)  
 humono(38,:) nitrate in lateral flow in HRU during month (kg N/ha)  
 humono(39,:) soluble phosphorus in surface runoff in HRU during month (kg P/ha)  
 humono(40,:) amount of nitrogen removed from soil by plant uptake in HRU during month (kg N/ha)  
 humono(41,:) nitrate percolating past bottom of soil profile in HRU during month (kg N/ha)  
 humono(42,:) amount of phosphorus removed from soil by plant uptake in HRU during month (kg P/ha)  
 humono(43,:) amount of phosphorus moving from labile mineral to active mineral pool in HRU during month (kg P/ha)  
 humono(44,:) amount of phosphorus moving from active mineral to stable mineral pool in HRU during month (kg P/ha)  
 humono(45,:) amount of nitrogen applied to HRU in fertilizer and grazing operations during month (kg N/ha)  
 humono(46,:) amount of phosphorus applied to HRU in fertilizer and grazing operations during month (kg P/ha)  
 humono(47,:) amount of nitrogen added to soil by fixation in HRU during month (kg N/ha)  
 humono(48,:) amount of nitrogen lost by denitrification in HRU during month (kg N/ha)  
 humono(49,:) amount of nitrogen moving from active organic to nitrate pool in HRU during month (kg N/ha)  
 humono(50,:) amount of nitrogen moving from active organic to stable organic pool in HRU during month (kg N/ha)  
 humono(51,:) amount of phosphorus moving from organic to labile mineral pool in HRU during month (kg P/ha)  
 humono(52,:) amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during month (kg N/ha)  
 humono(53,:) amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during month (kg P/ha)  
 humono(54,:) amount of nitrogen added to soil in rain (kg N/ha)  
 humono(61,:) daily soil loss predicted with USLE equation (metric tons/ha)  
 humono(63,:) less persistent bacteria transported to main channel from HRU during month (bacteria/ha)  
 humono(64,:) persistent bacteria transported to main channel from HRU during month (bacteria/ha)  
 humono(65,:) nitrate loading from groundwater in HRU to main channel during month (kg N/ha)  
 humono(66,:) soluble P loading from groundwater in HRU to main channel during month (kg P/ha)  
 humono(67,:) loading of mineral P attached to sediment in HRU to main channel during month (kg P/ha)

- real \*8, dimension(:,:), allocatable [parm::rchdy](#)

rchdy(1,:) flow into reach on day ( $m^3/s$ )  
 rchdy(2,:) flow out of reach on day ( $m^3/s$ )  
 rchdy(3,:) evaporation from reach on day ( $m^3/s$ )  
 rchdy(4,:) transmission losses from reach on day ( $m^3/s$ )



*rchdy(5,:) sediment transported into reach on day (metric tons)*  
*rchdy(6,:) sediment transported out of reach on day (metric tons)*  
*rchdy(7,:) sediment concentration in outflow (mg/L)*  
*rchdy(8,:) organic N transported into reach on day (kg N)*  
*rchdy(9,:) organic N transported out of reach on day (kg N)*  
*rchdy(10,:) organic P transported into reach on day (kg P)*  
*rchdy(11,:) organic P transported out of reach on day (kg P)*  
*rchdy(12,:) nitrate transported into reach on day (kg N)*  
*rchdy(13,:) nitrate transported out of reach on day (kg N)*  
*rchdy(14,:) ammonia transported into reach on day (kg N)*  
*rchdy(15,:) ammonia transported out of reach on day (kg N)*  
*rchdy(16,:) nitrite transported into reach on day (kg N)*  
*rchdy(17,:) nitrite transported out of reach on day (kg N)*  
*rchdy(18,:) soluble P transported into reach on day (kg P)*  
*rchdy(19,:) soluble P transported out of reach on day (kg P)*  
*rchdy(20,:) chlorophyll-a transported into reach on day (kg chla)*  
*rchdy(21,:) chlorophyll-a transported out of reach on day (kg chla)*  
*rchdy(22,:) CBOD transported into reach on day (kg O2)*  
*rchdy(23,:) CBOD transported out of reach on day (kg O2)*  
*rchdy(24,:) dissolved oxygen transported into reach on day (kg O2)*  
*rchdy(25,:) dissolved oxygen transported out of reach on day (kg O2)*  
*rchdy(26,:) soluble pesticide transported into reach on day (mg pst)*  
*rchdy(27,:) soluble pesticide transported out of reach on day (mg pst)*  
*rchdy(28,:) sorbed pesticide transported into reach on day (mg pst)*  
*rchdy(29,:) sorbed pesticide transported out of reach on day (mg pst)*  
*rchdy(30,:) amount of pesticide lost through reactions in reach on day (mg pst)*  
*rchdy(31,:) amount of pesticide lost through volatilization from reach on day (mg pst)*  
*rchdy(32,:) amount of pesticide settling out of reach to bed sediment on day (mg pst)*  
*rchdy(33,:) amount of pesticide resuspended from bed sediment to reach on day (mg pst)*  
*rchdy(34,:) amount of pesticide diffusing from reach to bed sediment on day (mg pst)*  
*rchdy(35,:) amount of pesticide in sediment layer lost through reactions on day (mg pst)*  
*rchdy(36,:) amount of pesticide in sediment layer lost through burial on day (mg pst)*  
*rchdy(37,:) amount of pesticide stored in river bed sediments (mg pst)*  
*rchdy(38,:) persistent bacteria transported out of reach on day (kg bact)*  
*rchdy(39,:) less persistent bacteria transported out of reach on day (kg bact)*  
*rchdy(40,:) amount of conservative metal #1 transported out of reach on day (kg)*  
*rchdy(41,:) amount of conservative metal #2 transported out of reach on day (kg)*  
*rchdy(42,:) amount of conservative metal #3 transported out of reach on day (kg)*  
*rchdy(43,:) total N (org N + no3 + no2 + nh4 outs) (kg)*  
*rchdy(44,:) total P (org P + sol p outs) (kg)*

- real \*8, dimension(:,:), allocatable [parm::hruyro](#)

*HRU annual output array (varies) hruyro(1,:) precipitation in HRU during year (mm H2O)*  
*hruyro(2,:) amount of precipitation falling as freezing rain/snow in HRU during year (mm H2O)*  
*hruyro(3,:) amount of snow melt in HRU during year (mm H2O)*  
*hruyro(4,:) amount of surface runoff to main channel from HRU during year (ignores impact of transmission losses) (mm H2O)*  
*hruyro(5,:) amount of lateral flow contribution to main channel from HRU during year (mm H2O)*  
*hruyro(6,:) amount of groundwater flow contribution to main channel from HRU during year (mm H2O)*  
*hruyro(7,:) amount of water moving from shallow aquifer to plants or soil profile in HRU during year (mm H2O)*  
*hruyro(8,:) amount of water recharging deep aquifer in HRU during year (mm H2O)*  
*hruyro(9,:) total amount of water entering both aquifers from HRU during year (mm H2O)*  
*hruyro(10,:) water yield (total amount of water entering main channel) from HRU during year (mm H2O)*  
*hruyro(11,:) amount of water percolating out of the soil profile and into the vadose zone in HRU during year (mm H2O)*  
*hruyro(12,:) actual evapotranspiration in HRU during year (mm H2O)*  
*hruyro(13,:) amount of transmission losses from tributary channels in HRU for year (mm H2O)*  
*hruyro(14,:) sediment yield from HRU for year (metric tons/ha)*  
*hruyro(15,:) actual amount of transpiration that occurs during year in HRU (mm H2O)*  
*hruyro(16,:) actual amount of evaporation (from soil) that occurs during year in HRU (mm H2O)*  
*hruyro(17,:) amount of nitrogen applied in continuous fertilizer operation during year in HRU (kg N/ha)*  
*hruyro(18,:) amount of phosphorus applied in continuous fertilizer operation during year in HRU (kg P/ha)*  
*hruyro(23,:) amount of water removed from shallow aquifer in HRU for irrigation during year (mm H2O)*  
*hruyro(24,:) amount of water removed from deep aquifer in HRU for irrigation during year (mm H2O)*

- hruyro(25,:) potential evapotranspiration in HRU during year (mm H2O)*
  - hruyro(26,:) annual amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)*
  - hruyro(27,:) annual amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)*
  - hruyro(28,:) annual amount of N (organic & mineral) auto-applied in HRU (kg N/ha)*
  - hruyro(29,:) annual amount of P (organic & mineral) auto-applied in HRU (kg P/ha)*
  - hruyro(31,:) water stress days in HRU during year (stress days)*
  - hruyro(32,:) temperature stress days in HRU during year (stress days)*
  - hruyro(33,:) nitrogen stress days in HRU during year (stress days)*
  - hruyro(34,:) phosphorus stress days in HRU during year (stress days)*
  - hruyro(35,:) organic nitrogen in surface runoff in HRU during year (kg N/ha)*
  - hruyro(36,:) organic phosphorus in surface runoff in HRU during year (kg P/ha)*
  - hruyro(37,:) nitrate in surface runoff in HRU during year (kg N/ha)*
  - hruyro(38,:) nitrate in lateral flow in HRU during year (kg N/ha)*
  - hruyro(39,:) soluble phosphorus in surface runoff in HRU during year (kg P/ha)*
  - hruyro(40,:) amount of nitrogen removed from soil by plant uptake in HRU during year (kg N/ha)*
  - hruyro(41,:) nitrate percolating past bottom of soil profile in HRU during year (kg N/ha)*
  - hruyro(42,:) amount of phosphorus removed from soil by plant uptake in HRU during year (kg P/ha)*
  - hruyro(43,:) amount of phosphorus moving from labile mineral to active mineral pool in HRU during year (kg P/ha)*
  - hruyro(44,:) amount of phosphorus moving from active mineral to stable mineral pool in HRU during year (kg P/ha)*
  - hruyro(45,:) amount of nitrogen applied to HRU in fertilizer and grazing operations during year (kg N/ha)*
  - hruyro(46,:) amount of phosphorus applied to HRU in fertilizer and grazing operations during year (kg P/ha)*
  - hruyro(47,:) amount of nitrogen added to soil by fixation in HRU during year (kg N/ha)*
  - hruyro(48,:) amount of nitrogen lost by denitrification in HRU during year (kg N/ha)*
  - hruyro(49,:) amount of nitrogen moving from active organic to nitrate pool in HRU during year (kg N/ha)*
  - hruyro(50,:) amount of nitrogen moving from active organic to stable organic pool in HRU during year (kg N/ha)*
  - hruyro(51,:) amount of phosphorus moving from organic to labile mineral pool in HRU during year (kg P/ha)*
  - hruyro(52,:) amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during year (kg N/ha)*
  - hruyro(53,:) amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during year (kg P/ha)*
  - hruyro(54,:) amount of nitrogen added to soil in rain during year (kg N/ha)*
  - hruyro(61,:) daily soil loss predicted with USLE equation (metric tons/ha)*
  - hruyro(63,:) less persistent bacteria transported to main channel from HRU during year (# bacteria/ha)*
  - hruyro(64,:) persistent bacteria transported to main channel from HRU during year (# bacteria/ha)*
  - hruyro(65,:) nitrate loading from groundwater in HRU to main channel during year (kg N/ha)*
  - hruyro(66,:) soluble P loading from groundwater in HRU to main channel during year (kg P/ha)*
  - hruyro(67,:) loading of mineral P attached to sediment in HRU to main channel during year (kg P/ha)*
- **real \*8, dimension(:,:), allocatable [parm::rchaa0](#)**
    - reach average annual output array (varies)*
    - rchaa0(1,:) flow into reach during simulation (m<sup>3</sup>/s)*
    - rchaa0(2,:) flow out of reach during simulation (m<sup>3</sup>/s)*
    - rchaa0(3,:) sediment transported into reach during simulation (metric tons)*
    - rchaa0(4,:) sediment transported out of reach during simulation (metric tons)*
    - rchaa0(5,:) sediment concentration in outflow during simulation (mg/L)*
    - rchaa0(6,:) organic N transported into reach during simulation (kg N)*
    - rchaa0(7,:) organic N transported out of reach during simulation (kg N)*
    - rchaa0(8,:) organic P transported into reach during simulation (kg P)*
    - rchaa0(9,:) organic P transported out of reach during simulation (kg P)*
    - rchaa0(10,:) evaporation from reach during simulation (m<sup>3</sup>/s)*
    - rchaa0(11,:) transmission losses from reach during simulation (m<sup>3</sup>/s)*
    - rchaa0(12,:) conservative metal #1 transported out of reach during simulation (kg)*
    - rchaa0(13,:) conservative metal #2 transported out of reach during simulation (kg)*
    - rchaa0(14,:) conservative metal #3 transported out of reach during simulation (kg)*
    - rchaa0(15,:) nitrate transported into reach during simulation (kg N)*
    - rchaa0(16,:) nitrate transported out of reach during simulation (kg N)*
    - rchaa0(17,:) soluble P transported into reach during simulation (kg P)*
    - rchaa0(18,:) soluble P transported out of reach during simulation (kg P)*
    - rchaa0(19,:) soluble pesticide transported into reach during simulation*
    - rchaa0(20,:) soluble pesticide transported out of reach during simulation*
    - rchaa0(21,:) sorbed pesticide transported into reach during simulation*
    - rchaa0(22,:) sorbed pesticide transported out of reach during simulation*
    - rchaa0(23,:) amount of pesticide lost through reactions in reach during simulation*
    - rchaa0(24,:) amount of pesticide lost through volatilization from reach during simulation*

- rchaao(25,:)* amount of pesticide settling out of reach to bed sediment during simulation
- rchaao(26,:)* amount of pesticide resuspended from bed sediment to reach during simulation
- rchaao(27,:)* amount of pesticide diffusing from reach to bed sediment during simulation
- rchaao(28,:)* amount of pesticide in sediment layer lost through reactions during simulation
- rchaao(29,:)* amount of pesticide in sediment layer lost through burial during simulation
- rchaao(30,:)* chlorophyll-a transported into reach during simulation (kg chl<sub>a</sub>)
- rchaao(31,:)* chlorophyll-a transported out of reach during simulation (kg chl<sub>a</sub>)
- rchaao(32,:)* ammonia transported into reach during simulation (kg N)
- rchaao(33,:)* ammonia transported out of reach during simulation (kg N)
- rchaao(34,:)* nitrite transported into reach during simulation (kg N)
- rchaao(35,:)* nitrite transported out of reach during simulation (kg N)
- rchaao(36,:)* CBOD transported into reach during simulation (kg O<sub>2</sub>)
- rchaao(37,:)* CBOD transported out of reach during simulation (kg O<sub>2</sub>)
- rchaao(38,:)* dissolved oxygen transported into reach during simulation (kg O<sub>2</sub>)
- rchaao(39,:)* dissolved oxygen transported out of reach during simulation (kg O<sub>2</sub>)
- rchaao(40,:)* persistent bacteria transported out of reach during simulation (kg bact)
- rchaao(41,:)* less persistent bacteria transported out of reach during simulation (kg bact)
- rchaao(43,:)* Total N (org N + no<sub>3</sub> + no<sub>2</sub> + nh<sub>4</sub> outs) (kg)
- rchaao(44,:)* Total P (org P + sol p outs) (kg)
- real \*8, dimension(:,:), allocatable [parm::submono](#)
  - subbasin* monthly output array (varies)
  - submono(1,:)* precipitation in subbasin for month (mm H<sub>2</sub>O)
  - submono(2,:)* snow melt in subbasin for month (mm H<sub>2</sub>O)
  - submono(3,:)* surface runoff loading in subbasin for month (mm H<sub>2</sub>O)
  - submono(4,:)* water yield from subbasin for month (mm H<sub>2</sub>O)
  - submono(5,:)* potential evapotranspiration in subbasin for month (mm H<sub>2</sub>O)
  - submono(6,:)* actual evapotranspiration in subbasin for month (mm H<sub>2</sub>O)
  - submono(7,:)* sediment yield from subbasin for month (metric tons/ha)
  - submono(8,:)* organic N loading from subbasin for month (kg N/ha)
  - submono(9,:)* organic P loading from subbasin for month (kg P/ha)
  - submono(10,:)* NO<sub>3</sub> loading from surface runoff in subbasin for month (kg N/ha)
  - submono(11,:)* soluble P loading from subbasin for month (kg P/ha)
  - submono(12,:)* groundwater loading from subbasin for month (mm H<sub>2</sub>O)
  - submono(13,:)* percolation out of soil profile in subbasin for month (mm H<sub>2</sub>O)
  - submono(14,:)* loading to reach of mineral P attached to sediment from subbasin for month (kg P/ha)
- real \*8, dimension(:,:), allocatable [parm::subyro](#)
  - subbasin* annual output array (varies)
  - subyro(1,:)* precipitation in subbasin for year (mm H<sub>2</sub>O)
  - subyro(2,:)* snow melt in subbasin for year (mm H<sub>2</sub>O)
  - subyro(3,:)* surface runoff loading in subbasin for year (mm H<sub>2</sub>O)
  - subyro(4,:)* water yield from subbasin for year (mm H<sub>2</sub>O)
  - subyro(5,:)* potential evapotranspiration in subbasin for year (mm H<sub>2</sub>O)
  - subyro(6,:)* actual evapotranspiration in subbasin for year (mm H<sub>2</sub>O)
  - subyro(7,:)* sediment yield from subbasin for year (metric tons/ha)
  - subyro(8,:)* organic N loading from subbasin for year (kg N/ha)
  - subyro(9,:)* organic P loading from subbasin for year (kg P/ha)
  - subyro(10,:)* NO<sub>3</sub> loading from surface runoff in subbasin for year (kg N/ha)
  - subyro(11,:)* soluble P loading from subbasin for year (kg P/ha)
  - subyro(12,:)* groundwater loading from subbasin for year (mm H<sub>2</sub>O)
  - subyro(13,:)* percolation out of soil profile in subbasin for year (mm H<sub>2</sub>O)
  - subyro(14,:)* loading to reach of mineral P attached to sediment from subbasin for year (kg P/ha)
- real \*8, dimension(:,:), allocatable [parm::hruaao](#)
  - HRU* average annual output array (varies)
  - hruaao(1,:)* precipitation in HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(2,:)* amount of precipitation falling as freezing rain/snow in HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(3,:)* amount of snow melt in HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(4,:)* amount of surface runoff to main channel from HRU during simulation (ignores impact of transmission losses) (mm H<sub>2</sub>O)
  - hruaao(5,:)* amount of lateral flow contribution to main channel from HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(6,:)* amount of groundwater flow contribution to main channel from HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(7,:)* amount of water moving from shallow aquifer to plants or soil profile in HRU during simulation (mm H<sub>2</sub>O)
  - hruaao(8,:)* amount of water recharging deep aquifer in HRU during simulation (mm H<sub>2</sub>O)



- hruaao(9,:)* total amount of water entering both aquifers from HRU during simulation (mm H<sub>2</sub>O)  
*hruaao(10,:)* water yield (total amount of water entering main channel) from HRU during simulation (mm H<sub>2</sub>O)  
*hruaao(11,:)* amount of water percolating out of the soil profile and into the vadose zone in HRU during simulation (mm H<sub>2</sub>O)  
*hruaao(12,:)* actual evapotranspiration in HRU during simulation  
*hruaao(13,:)* amount of transmission losses from tributary channels in HRU for simulation (mm H<sub>2</sub>O)  
*hruaao(14,:)* sediment yield from HRU for simulation (metric tons/ha)  
*hruaao(17,:)* amount of nitrogen applied in continuous fertilizer operation in HRU for simulation (kg N/ha)  
*hruaao(18,:)* amount of phosphorus applied in continuous fertilizer operation in HRU for simulation (kg P/ha)  
*hruaao(23,:)* amount of water removed from shallow aquifer in HRU for irrigation during simulation (mm H<sub>2</sub>O)  
*hruaao(24,:)* amount of water removed from deep aquifer in HRU for irrigation during simulation (mm H<sub>2</sub>O)  
*hruaao(25,:)* potential evapotranspiration in HRU during simulation (mm H<sub>2</sub>O)  
*hruaao(26,:)* annual amount of N (organic & mineral) applied in HRU during grazing (kg N/ha)  
*hruaao(27,:)* annual amount of P (organic & mineral) applied in HRU during grazing (kg P/ha)  
*hruaao(28,:)* average annual amount of N (organic & mineral) auto-applied in HRU (kg N/ha)  
*hruaao(29,:)* average annual amount of P (organic & mineral) auto-applied in HRU (kg P/ha)  
*hruaao(31,:)* water stress days in HRU during simulation (stress days)  
*hruaao(32,:)* temperature stress days in HRU during simulation (stress days)  
*hruaao(33,:)* nitrogen stress days in HRU during simulation (stress days)  
*hruaao(34,:)* phosphorus stress days in HRU during simulation (stress days)  
*hruaao(35,:)* organic nitrogen in surface runoff in HRU during simulation (kg N/ha)  
*hruaao(36,:)* organic phosphorus in surface runoff in HRU during simulation (kg P/ha)  
*hruaao(37,:)* nitrate in surface runoff in HRU during simulation (kg N/ha)  
*hruaao(38,:)* nitrate in lateral flow in HRU during simulation (kg N/ha)  
*hruaao(39,:)* soluble phosphorus in surface runoff in HRU during simulation (kg P/ha)  
*hruaao(40,:)* amount of nitrogen removed from soil by plant uptake in HRU during simulation (kg N/ha)  
*hruaao(41,:)* nitrate percolating past bottom of soil profile in HRU during simulation (kg N/ha)  
*hruaao(42,:)* amount of phosphorus removed from soil by plant uptake in HRU during simulation (kg P/ha)  
*hruaao(43,:)* amount of phosphorus moving from labile mineral to active mineral pool in HRU during simulation (kg P/ha)  
*hruaao(44,:)* amount of phosphorus moving from active mineral to stable mineral pool in HRU during simulation (kg P/ha)  
*hruaao(45,:)* amount of nitrogen applied to HRU in fertilizer and grazing operations during simulation (kg N/ha)  
*hruaao(46,:)* amount of phosphorus applied to HRU in fertilizer and grazing operations during simulation (kg P/ha)  
*hruaao(47,:)* amount of nitrogen added to soil by fixation in HRU during simulation (kg N/ha)  
*hruaao(48,:)* amount of nitrogen lost by denitrification in HRU during simulation (kg N/ha)  
*hruaao(49,:)* amount of nitrogen moving from active organic to nitrate pool in HRU during simulation (kg N/ha)  
*hruaao(50,:)* amount of nitrogen moving from active organic to stable organic pool in HRU during simulation (kg N/ha)  
*hruaao(51,:)* amount of phosphorus moving from organic to labile mineral pool in HRU during simulation (kg P/ha)  
*hruaao(52,:)* amount of nitrogen moving from fresh organic to nitrate and active organic pools in HRU during simulation (kg N/ha)  
*hruaao(53,:)* amount of phosphorus moving from fresh organic to the labile mineral and organic pools in HRU during simulation (kg P/ha)  
*hruaao(54,:)* amount of nitrogen added to soil in rain during simulation (kg N/ha)  
*hruaao(61,:)* daily soil loss predicted with USLE equation (metric tons/ha)  
*hruaao(63,:)* less persistent bacteria transported to main channel from HRU during simulation (# bacteria/ha)  
*hruaao(64,:)* persistent bacteria transported to main channel from HRU during simulation (# bacteria/ha)  
*hruaao(65,:)* nitrate loading from groundwater in HRU to main channel during simulation (kg N/ha)  
*hruaao(66,:)* soluble P loading from groundwater in HRU to main channel during simulation (kg P/ha)  
*hruaao(67,:)* loading of mineral P attached to sediment in HRU to main channel during simulation (kg P/ha)
- real \*8, dimension(:, :), allocatable [parm::subaao](#)  
 subbasin average annual output array (varies)
  - real \*8, dimension(:, :), allocatable [parm::resoutm](#)  
 reservoir monthly output array (varies)  
*resoutm(1,:)* flow into reservoir during month (m<sup>3</sup>/s)  
*resoutm(2,:)* flow out of reservoir during month (m<sup>3</sup>/s)  
*resoutm(3,:)* sediment entering reservoir during month (metric tons)  
*resoutm(4,:)* sediment leaving reservoir during month (metric tons)  
*resoutm(5,:)* sediment concentration in reservoir during month (mg/L)  
*resoutm(6,:)* pesticide entering reservoir during month (mg pst)  
*resoutm(7,:)* pesticide lost from reservoir through reactions during month (mg pst)  
*resoutm(8,:)* pesticide lost from reservoir through volatilization during month (mg pst)  
*resoutm(9,:)* pesticide moving from water to sediment through settling during month (mg pst)

*resoutm(10,:) pesticide moving from sediment to water through resuspension during month (mg pst)*  
*resoutm(11,:) pesticide moving from water to sediment through diffusion during month (mg pst)*  
*resoutm(12,:) pesticide lost from reservoir sediment layer through reactions during month (mg pst)*  
*resoutm(13,:) pesticide lost from reservoir sediment layer through burial during month (mg pst)*  
*resoutm(14,:) pesticide transported out of reservoir during month (mg pst)*  
*resoutm(15,:) pesticide concentration in reservoir water during month (mg pst/m<sup>3</sup>)*  
*resoutm(16,:) pesticide concentration in reservoir sediment layer during month (mg pst/m<sup>3</sup>)*  
*resoutm(17,:) evaporation from reservoir during month (m<sup>3</sup> H<sub>2</sub>O)*  
*resoutm(18,:) seepage from reservoir during month (m<sup>3</sup> H<sub>2</sub>O)*  
*resoutm(19,:) precipitation on reservoir during month (m<sup>3</sup> H<sub>2</sub>O)*  
*resoutm(22,:) organic N entering reservoir during month (kg N)*  
*resoutm(23,:) organic N leaving reservoir during month (kg N)*  
*resoutm(24,:) organic P entering reservoir during month (kg P)*  
*resoutm(25,:) organic P leaving reservoir during month (kg P)*  
*resoutm(26,:) nitrate entering reservoir during month (kg N)*  
*resoutm(27,:) nitrate leaving reservoir during month (kg N)*  
*resoutm(28,:) nitrite entering reservoir during month (kg N)*  
*resoutm(29,:) nitrite leaving reservoir during month (kg N)*  
*resoutm(30,:) ammonia entering reservoir during month (kg N)*  
*resoutm(31,:) ammonia leaving reservoir during month (kg N)*  
*resoutm(32,:) mineral P entering reservoir during month (kg P)*  
*resoutm(33,:) mineral P leaving reservoir during month (kg P)*  
*resoutm(34,:) chlorophyll-a entering reservoir during month (kg chla)*  
*resoutm(35,:) chlorophyll-a leaving reservoir during month (kg chla)*  
*resoutm(36,:) organic P concentration in reservoir water during month (mg P/L)*  
*resoutm(37,:) mineral P concentration in reservoir water during month (mg P/L)*  
*resoutm(38,:) organic N concentration in reservoir water during month (mg N/L)*  
*resoutm(39,:) nitrate concentration in reservoir water during month (mg N/L)*  
*resoutm(40,:) nitrite concentration in reservoir water during month (mg N/L)*  
*resoutm(41,:) ammonia concentration in reservoir water during month (mg N/L)*

- real \*8, dimension(:,:), allocatable [parm::resouty](#)

*reservoir annual output array (varies)*  
*resouty(1,:) flow into reservoir during year (m<sup>3</sup>/s)*  
*resouty(2,:) flow out of reservoir during year (m<sup>3</sup>/s)*  
*resouty(3,:) sediment entering reservoir during year (metric tons)*  
*resouty(4,:) sediment leaving reservoir during year (metric tons)*  
*resouty(5,:) sediment concentration in reservoir during year (mg/L)*  
*resouty(6,:) pesticide entering reservoir during year (mg pst)*  
*resouty(7,:) pesticide lost from reservoir through reactions during year (mg pst)*  
*resouty(8,:) pesticide lost from reservoir through volatilization during year (mg pst)*  
*resouty(9,:) pesticide moving from water to sediment through settling during year (mg pst)*  
*resouty(10,:) pesticide moving from sediment to water through resuspension during year (mg pst)*  
*resouty(11,:) pesticide moving from water to sediment through diffusion during year (mg pst)*  
*resouty(12,:) pesticide lost from reservoir sediment layer through reactions during year (mg pst)*  
*resouty(13,:) pesticide lost from reservoir sediment layer through burial during year (mg pst)*  
*resouty(14,:) pesticide transported out of reservoir during year (mg pst)*  
*resouty(15,:) pesticide concentration in reservoir water during year (mg pst/m<sup>3</sup>)*  
*resouty(16,:) pesticide concentration in reservoir sediment layer during year (mg pst/m<sup>3</sup>)*  
*resouty(17,:) evaporation from reservoir during year (m<sup>3</sup> H<sub>2</sub>O)*  
*resouty(18,:) seepage from reservoir during year (m<sup>3</sup> H<sub>2</sub>O)*  
*resouty(19,:) precipitation on reservoir during year (m<sup>3</sup> H<sub>2</sub>O)*  
*resouty(22,:) organic N entering reservoir during year (kg N)*  
*resouty(23,:) organic N leaving reservoir during year (kg N)*  
*resouty(24,:) organic P entering reservoir during year (kg P)*  
*resouty(25,:) organic P leaving reservoir during year (kg P)*  
*resouty(26,:) nitrate entering reservoir during year (kg N)*  
*resouty(27,:) nitrate leaving reservoir during year (kg N)*  
*resouty(28,:) nitrite entering reservoir during year (kg N)*  
*resouty(29,:) nitrite leaving reservoir during year (kg N)*  
*resouty(30,:) ammonia entering reservoir during year (kg N)*  
*resouty(31,:) ammonia leaving reservoir during year (kg N)*  
*resouty(32,:) mineral P entering reservoir during year (kg P)*  
*resouty(33,:) mineral P leaving reservoir during year (kg P)*

- resouty(34,:)* chlorophyll-a entering reservoir during year (kg chla)
- resouty(35,:)* chlorophyll-a leaving reservoir during year (kg chla)
- resouty(36,:)* organic P concentration in reservoir water during year (mg P/L)
- resouty(37,:)* mineral P concentration in reservoir water during year (mg P/L)
- resouty(38,:)* organic N concentration in reservoir water during year (mg N/L)
- resouty(39,:)* nitrate concentration in reservoir water during year (mg N/L)
- resouty(40,:)* nitrite concentration in reservoir water during year (mg N/L)
- resouty(41,:)* ammonia concentration in reservoir water during year (mg N/L)
- real \*8, dimension(:,:), allocatable [parm::resouta](#)
  - reservoir average annual output array (varies)*
  - resouta(3,:)* sediment entering reservoir during simulation (metric tons)
  - resouta(4,:)* sediment leaving reservoir during simulation (metric tons)
  - resouta(17,:)* evaporation from reservoir during simulation ( $m^3$  H2O)
  - resouta(18,:)* seepage from reservoir during simulation ( $m^3$  H2O)
  - resouta(19,:)* precipitation on reservoir during simulation ( $m^3$  H2O)
  - resouta(20,:)* water entering reservoir during simulation ( $m^3$  H2O)
  - resouta(21,:)* water leaving reservoir during simulation ( $m^3$  H2O)
- real \*8, dimension(12, 8) [parm::wshd\\_aamon](#)
  - wshd\_aamon(:,1)* average annual precipitation in watershed falling during month (mm H2O)
  - wshd\_aamon(:,2)* average annual freezing rain in watershed falling during month (mm H2O)
  - wshd\_aamon(:,3)* average annual surface runoff in watershed during month (mm H2O)
  - wshd\_aamon(:,4)* average annual lateral flow in watershed during month (mm H2O)
  - wshd\_aamon(:,5)* average annual water yield in watershed during month (mm H2O)
  - wshd\_aamon(:,6)* average annual actual evapotranspiration in watershed during month (mm H2O)
  - wshd\_aamon(:,7)* average annual sediment yield in watershed during month (metric tons)
  - wshd\_aamon(:,8)* average annual potential evapotranspiration in watershed during month (mm H2O)
- real \*8, dimension(:,:), allocatable [parm::wtrmon](#)
  - HRU monthly output data array for impoundments (varies)*
  - wtrmon(1,:)* evaporation from ponds in HRU for month (mm H2O)
  - wtrmon(2,:)* seepage from ponds in HRU for month (mm H2O)
  - wtrmon(3,:)* precipitation on ponds in HRU for month (mm H2O)
  - wtrmon(4,:)* amount of water entering ponds in HRU for month (mm H2O)
  - wtrmon(5,:)* sediment entering ponds in HRU for month (metric tons/ha)
  - wtrmon(6,:)* amount of water leaving ponds in HRU for month (mm H2O)
  - wtrmon(7,:)* sediment leaving ponds in HRU for month (metric tons/ha)
  - wtrmon(8,:)* precipitation on wetlands in HRU for month (mm H2O)
  - wtrmon(9,:)* volume of water entering wetlands from HRU for month (mm H2O)
  - wtrmon(10,:)* sediment loading to wetlands for month from HRU (metric tons/ha)
  - wtrmon(11,:)* evaporation from wetlands in HRU for month (mm H2O)
  - wtrmon(12,:)* seepage from wetlands in HRU for month (mm H2O)
  - wtrmon(13,:)* volume of water leaving wetlands in HRU for month (mm H2O)
  - wtrmon(14,:)* sediment loading from wetlands in HRU to main channel during month (metric tons/ha)
  - wtrmon(15,:)* precipitation on potholes in HRU for month (mm H2O)
  - wtrmon(16,:)* evaporation from potholes in HRU for month (mm H2O)
  - wtrmon(17,:)* seepage from potholes in HRU for month (mm H2O)
  - wtrmon(18,:)* water leaving potholes in HRU for month (mm H2O)
  - wtrmon(19,:)* water entering potholes in HRU for month (mm H2O)
  - wtrmon(20,:)* sediment entering potholes in HRU for month (metric tons/ha)
  - wtrmon(21,:)* sediment leaving potholes in HRU for month (metric tons/ha)
- real \*8, dimension(:,:), allocatable [parm::wtryr](#)
  - HRU impoundment annual output array (varies)*
  - wtryr(1,:)* evaporation from ponds in HRU for year (mm H2O)
  - wtryr(2,:)* seepage from ponds in HRU for year (mm H2O)
  - wtryr(3,:)* precipitation on ponds in HRU for year (mm H2O)
  - wtryr(4,:)* amount of water entering ponds in HRU for year (mm H2O)
  - wtryr(5,:)* sediment entering ponds in HRU for year (metric tons/ha)
  - wtryr(6,:)* amount of water leaving ponds in HRU for year (mm H2O)
  - wtryr(7,:)* sediment leaving ponds in HRU for year (metric tons/ha)
  - wtryr(8,:)* precipitation on wetlands in HRU for year (mm H2O)
  - wtryr(9,:)* volume of water entering wetlands from HRU for year (mm H2O)
  - wtryr(10,:)* sediment loading to wetlands for year from HRU (metric tons/ha)
  - wtryr(11,:)* evaporation from wetlands in HRU for year (mm H2O)

- wtryr(12,:) seepage from wetlands in HRU for year (mm H2O)*
- wtryr(13,:) volume of water leaving wetlands in HRU for year (mm H2O)*
- wtryr(14,:) sediment loading from wetlands in HRU to main channel during year (metric tons/ha)*
- wtryr(15,:) precipitation on potholes in HRU during year (mm H2O)*
- wtryr(16,:) evaporation from potholes in HRU during year (mm H2O)*
- wtryr(17,:) seepage from potholes in HRU during year (mm H2O)*
- wtryr(18,:) water leaving potholes in HRU during year (mm H2O)*
- wtryr(19,:) water entering potholes in HRU during year (mm H2O)*
- wtryr(20,:) sediment entering potholes in HRU during year (metric tons/ha)*
- wtryr(21,:) sediment leaving potholes in HRU during year (metric tons/ha)*
- real \*8, dimension(:,:), allocatable [parm::wtraa](#)  
*HRU impoundment average annual output array (varies)*
- real \*8, dimension(:,:), allocatable [parm::sub\\_smfmx](#)  
*max melt rate for snow during year (June 21) for subbasin(:) where deg C refers to the air temperature. SUB\_SMFMX and SMFMN allow the rate of snow melt to vary through the year. These parameters are accounting for the impact of soil temperature on snow melt (range: -5.0/5.0) (mm/deg C/day)*
- real \*8, dimension(:,:), allocatable [parm::sub\\_smfmn](#)  
*min melt rate for snow during year (Dec 21) for subbasin(:) (range: -5.0/5.0) where deg C refers to the air temperature (mm/deg C/day)*
- real \*8, dimension(:,:), allocatable [parm::hrupstd](#)  
*hrupstd(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU on day (in solution) (mg pst)*  
*hrupstd(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU on day (sorbed to sediment) (mg pst)*
- real \*8, dimension(:,:), allocatable [parm::hrupstm](#)  
*hrupstm(:, :, :) HRU monthly pesticide output array (varies)*  
*hrupstm(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU during month (in solution) (mg pst)*  
*hrupstm(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU during month (sorbed to sediment) (mg pst)*  
*hrupstm(:,3,:) total pesticide loading to stream in surface runoff from HRU during month (mg pst)*
- real \*8, dimension(:,:), allocatable [parm::hrupsta](#)  
*HRU average annual pesticide output array (varies)*
- real \*8, dimension(:,:), allocatable [parm::hrupsty](#)  
*hrupsty(:, :, :) HRU annual pesticide output array (varies)*  
*hrupsty(:,1,:) amount of pesticide type in surface runoff contribution to stream from HRU during year (in solution) (mg pst)*  
*hrupsty(:,2,:) amount of pesticide type in surface runoff contribution to stream from HRU during year (sorbed to sediment) (mg pst)*
- integer, dimension(:), allocatable [parm::ifirstt](#)  
*temperature data search code (none)*  
*0 first day of temperature data located in file*  
*1 first day of temperature data not located in file*
- integer, dimension(:), allocatable [parm::ifirstpcp](#)
- integer, dimension(:), allocatable [parm::elevp](#)  
*elevation of precipitation gage station (m)*
- integer, dimension(:), allocatable [parm::elevt](#)  
*elevation of temperature gage station (m)*
- real \*8, dimension(:,:), allocatable [parm::ftmpmn](#)  
*avg monthly minimum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::ftmpmx](#)  
*avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::ftmpstdmn](#)  
*standard deviation for avg monthly minimum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::ftmpstdmx](#)  
*standard deviation for avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::fpcp\\_stat](#)

- fpcp\_stat(:,1,:): average amount of precipitation falling in one day for the month (mm/day)*
- fpcp\_stat(:,2,:): standard deviation for the average daily precipitation (mm/day)*
- fpcp\_stat(:,3,:): skew coefficient for the average daily precipitation (none)*
- real \*8, dimension(:,:), allocatable **parm::fpr\_w1**  
*probability of wet day after dry day in month (none)*
- real \*8, dimension(:,:), allocatable **parm::fpr\_w2**  
*probability of wet day after wet day in month (none)*
- real \*8, dimension(:,:), allocatable **parm::fpr\_w3**  
*proportion of wet days in the month (none)*
- real \*8, dimension(:), allocatable **parm::ch\_d**  
*average depth of main channel (m)*
- real \*8, dimension(:), allocatable **parm::flwin**
- real \*8, dimension(:), allocatable **parm::flwout**
- real \*8, dimension(:), allocatable **parm::bankst**
- real \*8, dimension(:), allocatable **parm::ch\_wi**
- real \*8, dimension(:), allocatable **parm::ch\_onco**  
*channel organic n concentration (ppm)*
- real \*8, dimension(:), allocatable **parm::ch\_opco**  
*channel organic p concentration (ppm)*
- real \*8, dimension(:), allocatable **parm::ch\_orgn**
- real \*8, dimension(:), allocatable **parm::ch\_orgp**
- real \*8, dimension(:), allocatable **parm::drift**  
*amount of pesticide drifting onto main channel in subbasin (kg)*
- real \*8, dimension(:), allocatable **parm::rch\_dox**
- real \*8, dimension(:), allocatable **parm::rch\_bactp**
- real \*8, dimension(:), allocatable **parm::alpha\_bnk**  
*alpha factor for bank storage recession curve (days)*
- real \*8, dimension(:), allocatable **parm::alpha\_bnke**  
 *$\exp(-\alpha_{bnk})$  (none)*
- real \*8, dimension(:), allocatable **parm::disolvp**
- real \*8, dimension(:), allocatable **parm::algae**
- real \*8, dimension(:), allocatable **parm::sedst**
- real \*8, dimension(:), allocatable **parm::rchstor**
- real \*8, dimension(:), allocatable **parm::organicn**
- real \*8, dimension(:), allocatable **parm::organicp**
- real \*8, dimension(:), allocatable **parm::chlora**
- real \*8, dimension(:), allocatable **parm::ch\_li**  
*initial length of main channel (km)*
- real \*8, dimension(:), allocatable **parm::ch\_si**  
*initial slope of main channel (m/m)*
- real \*8, dimension(:), allocatable **parm::nitraten**
- real \*8, dimension(:), allocatable **parm::nitriten**
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_san**
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_sil**
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_cla**
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_gra**
- real \*8, dimension(:), allocatable **parm::ch\_bed\_san**
- real \*8, dimension(:), allocatable **parm::ch\_bed\_sil**
- real \*8, dimension(:), allocatable **parm::ch\_bed\_cla**
- real \*8, dimension(:), allocatable **parm::ch\_bed\_gra**
- real \*8, dimension(:), allocatable **parm::depfp**
- real \*8, dimension(:), allocatable **parm::depsanfp**
- real \*8, dimension(:), allocatable **parm::depsilfp**



- real \*8, dimension(:), allocatable **parm::depclafp**
- real \*8, dimension(:), allocatable **parm::depsagfp**
- real \*8, dimension(:), allocatable **parm::deplagfp**
- real \*8, dimension(:), allocatable **parm::depch**
- real \*8, dimension(:), allocatable **parm::depsanch**
- real \*8, dimension(:), allocatable **parm::depsilch**
- real \*8, dimension(:), allocatable **parm::depclach**
- real \*8, dimension(:), allocatable **parm::depsagch**
- real \*8, dimension(:), allocatable **parm::deplagch**
- real \*8, dimension(:), allocatable **parm::depgrach**
- real \*8, dimension(:), allocatable **parm::depgrafp**
- real \*8, dimension(:), allocatable **parm::grast**
- real \*8, dimension(:), allocatable **parm::r2adj**  
*curve number retention parameter adjustment factor to adjust surface runoff for flat slopes (0.5 - 3.0) (dimensionless)*
- real \*8, dimension(:), allocatable **parm::prf**  
*Reach peak rate adjustment factor for sediment routing in the channel. Allows impact of peak flow rate on sediment routing and channel reshaping to be taken into account (none)*
- real \*8, dimension(:), allocatable **parm::depprch**
- real \*8, dimension(:), allocatable **parm::depprfp**
- real \*8, dimension(:), allocatable **parm::spcon**  
*linear parameter for calculating sediment reentrained in channel sediment routing*
- real \*8, dimension(:), allocatable **parm::spexp**  
*exponent parameter for calculating sediment reentrained in channel sediment routing*
- real \*8, dimension(:), allocatable **parm::sanst**
- real \*8, dimension(:), allocatable **parm::silst**
- real \*8, dimension(:), allocatable **parm::clast**
- real \*8, dimension(:), allocatable **parm::sagst**
- real \*8, dimension(:), allocatable **parm::lagst**
- real \*8, dimension(:), allocatable **parm::pot\_san**
- real \*8, dimension(:), allocatable **parm::pot\_sil**
- real \*8, dimension(:), allocatable **parm::pot\_cla**
- real \*8, dimension(:), allocatable **parm::pot\_sag**
- real \*8, dimension(:), allocatable **parm::pot\_lag**
- real \*8, dimension(:), allocatable **parm::potsani**
- real \*8, dimension(:), allocatable **parm::potsili**
- real \*8, dimension(:), allocatable **parm::potclai**
- real \*8, dimension(:), allocatable **parm::potsagi**
- real \*8, dimension(:), allocatable **parm::potlagi**
- real \*8, dimension(:), allocatable **parm::sanyld**
- real \*8, dimension(:), allocatable **parm::silyld**
- real \*8, dimension(:), allocatable **parm::clayld**
- real \*8, dimension(:), allocatable **parm::sagyld**
- real \*8, dimension(:), allocatable **parm::lagyld**
- real \*8, dimension(:), allocatable **parm::grayld**
- real \*8, dimension(:), allocatable **parm::res\_san**
- real \*8, dimension(:), allocatable **parm::res\_sil**
- real \*8, dimension(:), allocatable **parm::res\_cla**
- real \*8, dimension(:), allocatable **parm::res\_sag**
- real \*8, dimension(:), allocatable **parm::res\_lag**
- real \*8, dimension(:), allocatable **parm::res\_gra**
- real \*8, dimension(:), allocatable **parm::pnd\_san**
- real \*8, dimension(:), allocatable **parm::pnd\_sil**
- real \*8, dimension(:), allocatable **parm::pnd\_cla**
- real \*8, dimension(:), allocatable **parm::pnd\_sag**

- real \*8, dimension(:), allocatable **parm::pnd\_lag**
- real \*8, dimension(:), allocatable **parm::wet\_san**
- real \*8, dimension(:), allocatable **parm::wet\_sil**
- real \*8, dimension(:), allocatable **parm::wet\_cla**
- real \*8, dimension(:), allocatable **parm::wet\_lag**
- real \*8, dimension(:), allocatable **parm::wet\_sag**
- real \*8 **parm::ressano**
- real \*8 **parm::ressilo**
- real \*8 **parm::resclao**
- real \*8 **parm::ressago**
- real \*8 **parm::reslago**
- real \*8 **parm::resgrao**
- real \*8 **parm::ressani**
- real \*8 **parm::ressili**
- real \*8 **parm::resclai**
- real \*8 **parm::ressagi**
- real \*8 **parm::reslgi**
- real \*8 **parm::resgrai**
- real \*8 **parm::potsano**
- real \*8 **parm::potsilo**
- real \*8 **parm::potclao**
- real \*8 **parm::potsago**
- real \*8 **parm::potlago**
- real \*8 **parm::pndsarin**
- real \*8 **parm::pndsilin**
- real \*8 **parm::pndclain**
- real \*8 **parm::pndsagin**
- real \*8 **parm::pndlagin**
- real \*8 **parm::pndsano**
- real \*8 **parm::pndsilo**
- real \*8 **parm::pndclao**
- real \*8 **parm::pndsago**
- real \*8 **parm::pndlago**
- real \*8, dimension(:), allocatable **parm::ch\_di**  
*initial depth of main channel (m)*
- real \*8, dimension(:), allocatable **parm::ch\_erod**  
*channel erodibility factor (0.0-1.0) (none)*  
*0 non-erosive channel*  
*1 no resistance to erosion*
- real \*8, dimension(:), allocatable **parm::ch\_l2**  
*length of main channel (km)*
- real \*8, dimension(:), allocatable **parm::ch\_cov**
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_bd**  
*bulk density of channel bank sediment (1.1-1.9) (g/cc)*
- real \*8, dimension(:), allocatable **parm::ch\_bed\_bd**  
*bulk density of channel bed sediment (1.1-1.9) (g/cc)*
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_kd**  
*erodibility of channel bank sediment by jet test (Peter Allen needs to give more info on this)*
- real \*8, dimension(:), allocatable **parm::ch\_bed\_kd**  
*erodibility of channel bed sediment by jet test (Peter Allen needs to give more info on this)*
- real \*8, dimension(:), allocatable **parm::ch\_bnk\_d50**  
*D50(median) particle size diameter of channel bank sediment (0.001 - 20)*
- real \*8, dimension(:), allocatable **parm::ch\_bed\_d50**

- D50(median) particle size diameter of channel bed sediment (micrometers) (0.001 - 20)*

  - real \*8, dimension(:), allocatable [parm::ch\\_cov1](#)  
*channel erodibility factor (0.0-1.0) (none)*  
*0 non-erosive channel*  
*1 no resistance to erosion*
- real \*8, dimension(:), allocatable [parm::ch\\_cov2](#)  
*channel cover factor (0.0-1.0) (none)*  
*0 channel is completely protected from erosion by cover*  
*1 no vegetative cover on channel*
- real \*8, dimension(:), allocatable [parm::tc\\_bed](#)  
*critical shear stress of channel bed (N/m2)*
- real \*8, dimension(:), allocatable [parm::tc\\_bnk](#)  
*critical shear stress of channel bank (N/m2)*
- integer, dimension(:), allocatable [parm::ch\\_eqn](#)  
*sediment routine methods (DAILY):*  
*0 = original SWAT method*  
*1 = Bagnold's*  
*2 = Kodatie*  
*3 = Molinas WU*  
*4 = Yang*
- real \*8, dimension(:), allocatable [parm::chpst\\_rea](#)  
*pesticide reaction coefficient in reach (1/day)*
- real \*8, dimension(:), allocatable [parm::chpst\\_vol](#)  
*pesticide volatilization coefficient in reach (m/day)*
- real \*8, dimension(:), allocatable [parm::chpst\\_conc](#)
- real \*8, dimension(:), allocatable [parm::chpst\\_koc](#)  
*pesticide partition coefficient between water and sediment in reach ( $m^3/g$ )*
- real \*8, dimension(:), allocatable [parm::chpst\\_rsp](#)  
*resuspension velocity in reach for pesticide sorbed to sediment (m/day)*
- real \*8, dimension(:), allocatable [parm::chpst\\_stl](#)  
*settling velocity in reach for pesticide sorbed to sediment (m/day)*
- real \*8, dimension(:), allocatable [parm::ch\\_wdr](#)  
*channel width to depth ratio (m/m)*
- real \*8, dimension(:), allocatable [parm::chpst\\_mix](#)  
*mixing velocity (diffusion/dispersion) for pesticide in reach (m/day)*
- real \*8, dimension(:), allocatable [parm::sedpst\\_conc](#)  
*initil pesticide concentration in river bed sediment ( $mg/m^3$ )*
- real \*8, dimension(:), allocatable [parm::sedpst\\_bry](#)  
*pesticide burial velocity in river bed sediment (m/day)*
- real \*8, dimension(:), allocatable [parm::sedpst\\_rea](#)  
*pesticide reaction coefficient in river bed sediment (1/day)*
- real \*8, dimension(:), allocatable [parm::sedpst\\_act](#)  
*depth of active sediment layer in reach for pesticide (m)*
- real \*8, dimension(:), allocatable [parm::rch\\_cbod](#)
- real \*8, dimension(:), allocatable [parm::rch\\_bactlp](#)
- real \*8, dimension(:), allocatable [parm::chside](#)  
*change in horizontal distance per unit vertical distance (0.0 - 5)*  
*0 = for vertical channel bank*  
*5 = for channel bank with gentl side slope*
- real \*8, dimension(:), allocatable [parm::rs1](#)  
*local algal settling rate in reach at 20 deg C (m/day or m/hour)*
- real \*8, dimension(:), allocatable [parm::rs2](#)



- benthos source rate for dissolved phosphorus in reach at 20 deg C ((mg disP-P)/(m<sup>2</sup>\*day) or (mg disP-P)/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable **parm::rs3**
- benthos source rate for ammonia nitrogen in reach at 20 deg C ((mg NH4-N)/(m<sup>2</sup>\*day) or (mg NH4-N)/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable **parm::rs4**
- rate coefficient for organic nitrogen settling in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rs5**
- organic phosphorus settling rate in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rk1**
- CBOD deoxygenation rate coefficient in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rk2**
- reaeration rate in accordance with Fickian diffusion in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rk3**
- rate of loss of CBOD due to settling in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rk4**
- sediment oxygen demand rate in reach at 20 deg C (mg O2/(m<sup>2</sup>\*day) or mg O2/(m<sup>2</sup>\*hour))*
- real \*8, dimension(:), allocatable **parm::rk5**
- coliform die-off rate in reach (1/day)*
- real \*8, dimension(:), allocatable **parm::rs6**
- rate coefficient for settling of arbitrary non-conservative constituent in reach (1/day)*
- real \*8, dimension(:), allocatable **parm::rs7**
- benthos source rate for arbitrary non-conservative constituent in reach ((mg ANC)/(m<sup>2</sup>\*day))*
- real \*8, dimension(:), allocatable **parm::bc1**
- rate constant for biological oxidation of NH3 to NO2 in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::bc2**
- rate constant for biological oxidation of NO2 to NO3 in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::bc3**
- rate constant for hydrolysis of organic N to ammonia in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::bc4**
- rate constant for the decay of organic P to dissolved P in reach at 20 deg C (1/day or 1/hour)*
- real \*8, dimension(:), allocatable **parm::rk6**
- decay rate for arbitrary non-conservative constituent in reach (1/day)*
- real \*8, dimension(:), allocatable **parm::ammonian**
- real \*8, dimension(:), allocatable **parm::orig\_sedpstconc**
- real \*8, dimension(:,:), allocatable **parm::wurch**
- average daily water removal from the reach for the month (10<sup>4</sup> m<sup>3</sup>/day)*
- integer, dimension(:), allocatable **parm::icanal**
- integer, dimension(:), allocatable **parm::itb**
- real \*8, dimension(:), allocatable **parm::ch\_revap**
- revap coeff: this variable controls the amount of water moving from bank storage to the root zone as a result of soil moisture depletion(none)*
- real \*8, dimension(:), allocatable **parm::dep\_chan**
- real \*8, dimension(:), allocatable **parm::harg\_petco**
- coefficient related to radiation used in hargreaves eq (range: 0.0019 - 0.0032)*
- real \*8, dimension(:), allocatable **parm::subfr\_nowtr**
- real \*8, dimension(:), allocatable **parm::cncoef\_sub**
- soil water depletion coefficient used in the new (modified curve number method) same as soil index coeff used in APEX range: 0.5 - 2.0*
- real \*8, dimension(:), allocatable **parm::dr\_sub**
- real \*8, dimension(:), allocatable **parm::sub\_fr**
- fraction of total watershed area contained in subbasin (km2/km2)*
- real \*8, dimension(:), allocatable **parm::sub\_sw**

- water in soil profile in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable **parm::wcklsp**
- real \*8, dimension(:), allocatable **parm::sub\_minp**
- real \*8, dimension(:), allocatable **parm::sub\_sumfc**
- real \*8, dimension(:), allocatable **parm::sub\_gwno3**
- real \*8, dimension(:), allocatable **parm::sub\_gwsolp**
- real \*8, dimension(:), allocatable **parm::co2**
- CO2 concentration (ppmv)*
- real \*8, dimension(:), allocatable **parm::sub\_km**
- area of subbasin in square kilometers (km<sup>2</sup>)*
- real \*8, dimension(:), allocatable **parm::wlat**
- latitude of weather station used to compile data (degrees)*
- real \*8, dimension(:), allocatable **parm::sub\_tc**
- time of concentration for subbasin (hour)*
- real \*8, dimension(:), allocatable **parm::sub\_pet**
- real \*8, dimension(:), allocatable **parm::welev**
- elevation of weather station used to compile weather generator data (m)*
- real \*8, dimension(:), allocatable **parm::sub\_bd**
- bulk density in subbasin first soil layer (Mg/m<sup>3</sup>)*
- real \*8, dimension(:), allocatable **parm::sub\_orgn**
- amount of nitrogen stored in all organic pools (kg N/ha)*
- real \*8, dimension(:), allocatable **parm::sub\_orgp**
- amount of phosphorus stored in all organic pools (kg P/ha)*
- real \*8, dimension(:), allocatable **parm::sub\_wtmp**
- real \*8, dimension(:), allocatable **parm::sub\_sedpa**
- real \*8, dimension(:), allocatable **parm::sub\_sedps**
- real \*8, dimension(:), allocatable **parm::daylmn**
- shortest daylength occurring during the year (hour)*
- real \*8, dimension(:), allocatable **parm::sub\_minpa**
- amount of phosphorus stored in active mineral pools sorbed to sediment (kg P/ha)*
- real \*8, dimension(:), allocatable **parm::sub\_minps**
- amount of phosphorus stored in stable mineral pools sorbed to sediment (kg P/ha)*
- real \*8, dimension(:), allocatable **parm::latcos**
- cos(latitude) (none)*
- real \*8, dimension(:), allocatable **parm::latsin**
- sin(latitude) (none)*
- real \*8, dimension(:), allocatable **parm::phutot**
- total potential heat units for year (used when no crop is growing) (heat unit)*
- real \*8, dimension(:), allocatable **parm::plaps**
- precipitation lapse rate: precipitation change due to change in elevation (mm H2O/km)*
- real \*8, dimension(:), allocatable **parm::tlaps**
- temperature lapse rate: temperature change due to change in elevation (deg C/km)*
- real \*8, dimension(:), allocatable **parm::tmp\_an**
- average annual air temperature (deg C)*
- real \*8, dimension(:), allocatable **parm::sub\_precip**
- amount of water reaching soil surface in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable **parm::rammo\_sub**
- atmospheric deposition of ammonium values for entire watershed (mg/l)*
- real \*8, dimension(:), allocatable **parm::rcn\_sub**
- atmospheric deposition of nitrate for entire watershed (mg/l)*
- real \*8, dimension(:), allocatable **parm::pcpdays**

- real \*8, dimension(:), allocatable **parm::atmo\_day**
- real \*8, dimension(:), allocatable **parm::sub\_snom**  
*amount of snow melt in subbasin on day (mm H2O)*
- real \*8, dimension(:), allocatable **parm::sub\_qd**  
*surface runoff that reaches main channel during day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable **parm::sub\_sedy**
- real \*8, dimension(:), allocatable **parm::sub\_tran**  
*transmission losses on day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable **parm::sub\_no3**
- real \*8, dimension(:), allocatable **parm::sub\_latno3**
- real \*8, dimension(:, :), allocatable **parm::sub\_sftmp**  
*snowfall temperature for subbasin(:). Mean air temperature at which precip is equally likely to be rain as snow/freezing rain (range: -5.0/5.0) (deg C)*
- real \*8, dimension(:, :), allocatable **parm::sub\_smtmp**  
*snow melt base temperature for subbasin(:) mean air temperature at which snow melt will occur (range: -5.0/5.0) (deg C)*
- real \*8, dimension(:, :), allocatable **parm::sub\_timp**  
*snow pack temperature lag factor (0-1) (none)*  
*1 = no lag (snow pack temp=current day air temp) as the lag factor goes to zero, the snow pack's temperature will be less influenced by the current day's air temperature*
- real \*8, dimension(:), allocatable **parm::sub\_tileno3**
- real \*8, dimension(:), allocatable **parm::sub\_solp**
- real \*8, dimension(:), allocatable **parm::sub\_subp**
- real \*8, dimension(:), allocatable **parm::sub\_etday**
- real \*8, dimension(:), allocatable **parm::sub\_elev**  
*average elevation of HRU (m)*
- real \*8, dimension(:), allocatable **parm::sub\_surfq**  
*surface runoff generated on day in subbasin (mm H2O)*
- real \*8, dimension(:), allocatable **parm::sub\_wyld**
- real \*8, dimension(:), allocatable **parm::qird**
- real \*8, dimension(:), allocatable **parm::sub\_gwq**
- real \*8, dimension(:), allocatable **parm::sub\_sep**
- real \*8, dimension(:), allocatable **parm::sub\_chl**
- real \*8, dimension(:), allocatable **parm::sub\_cbod**
- real \*8, dimension(:), allocatable **parm::sub\_dox**
- real \*8, dimension(:), allocatable **parm::sub\_solpst**
- real \*8, dimension(:), allocatable **parm::sub\_sorpst**
- real \*8, dimension(:), allocatable **parm::sub\_yorgn**
- real \*8, dimension(:), allocatable **parm::sub\_yorgp**
- real \*8, dimension(:), allocatable **parm::sub\_lat**  
*latitude of HRU/subbasin (degrees)*
- real \*8, dimension(:), allocatable **parm::sub\_bactp**
- real \*8, dimension(:), allocatable **parm::sub\_bactlp**
- real \*8, dimension(:), allocatable **parm::sub\_latq**
- real \*8, dimension(:), allocatable **parm::sub\_gwq\_d**
- real \*8, dimension(:), allocatable **parm::sub\_tileq**
- real \*8, dimension(:), allocatable **parm::sub\_vaptile**
- real \*8, dimension(:), allocatable **parm::sub\_dsan**
- real \*8, dimension(:), allocatable **parm::sub\_dsil**
- real \*8, dimension(:), allocatable **parm::sub\_dcla**
- real \*8, dimension(:), allocatable **parm::sub\_dsag**
- real \*8, dimension(:), allocatable **parm::sub\_dlag**
- real \*8 **parm::vap\_tile**
- real \*8, dimension(:), allocatable **parm::wnan**

- real \*8, dimension(:,:), allocatable **parm::sol\_stpwt**
- real \*8, dimension(:,:), allocatable **parm::sub\_pst**  
*amount of pesticide in layer in subbasin (kg/ha)*
- real \*8, dimension(:,:), allocatable **parm::sub\_hhqd**
- real \*8, dimension(:,:), allocatable **parm::sub\_hhwtmp**
- real \*8, dimension(:,:), allocatable **parm::huminc**  
*monthly humidity adjustment. Daily values for relative humidity within the month are raised or lowered by the specified amount (used in climate change studies) (none)*
- real \*8, dimension(:,:), allocatable **parm::radinc**  
*monthly solar radiation adjustment. Daily radiation within the month is raised or lowered by the specified amount (used in climate change studies) (MJ/m<sup>2</sup>)*
- real \*8, dimension(:,:), allocatable **parm::rfinc**  
*monthly rainfall adjustment. Daily rainfall within the month is adjusted to the specified percentage of the original value (used in climate change studies)(%)*
- real \*8, dimension(:,:), allocatable **parm::tmpinc**  
*monthly temperature adjustment. Daily maximum and minimum temperatures within the month are raised or lowered by the specified amount (used in climate change studies) (deg C)*
- real \*8, dimension(:), allocatable **parm::ch\_k1**  
*effective hydraulic conductivity of tributary channel alluvium (mm/hr)*
- real \*8, dimension(:), allocatable **parm::ch\_k2**  
*effective hydraulic conductivity of main channel alluvium (mm/hr)*
- real \*8, dimension(:,:), allocatable **parm::elevb**  
*elevation at the center of the band in subbasin (m)*
- real \*8, dimension(:,:), allocatable **parm::elevb\_fr**  
*fraction of subbasin area within elevation band (the same fractions should be listed for all HRUs within the subbasin) (none)*
- real \*8, dimension(:,:), allocatable **parm::wndav**  
*average wind speed for the month (m/s)*
- real \*8, dimension(:), allocatable **parm::ch\_n1**  
*Manning's "n" value for the tributary channels (none)*
- real \*8, dimension(:), allocatable **parm::ch\_n2**  
*Manning's "n" value for the main channel (none)*
- real \*8, dimension(:), allocatable **parm::ch\_s1**  
*average slope of tributary channels (m/m)*
- real \*8, dimension(:), allocatable **parm::ch\_s2**  
*average slope of main channel (m/m)*
- real \*8, dimension(:), allocatable **parm::ch\_w1**  
*average width of tributary channels (m)*
- real \*8, dimension(:), allocatable **parm::ch\_w2**  
*average width of main channel (m)*
- real \*8, dimension(:,:), allocatable **parm::dewpt**  
*average dew point temperature for the month (deg C)*
- real \*8, dimension(:,:), allocatable **parm::amp\_r**  
*average fraction of total daily rainfall occurring in maximum half-hour period for month (none)*
- real \*8, dimension(:,:), allocatable **parm::solarav**  
*average daily solar radiation for the month (MJ/m<sup>2</sup>/day)*
- real \*8, dimension(:,:), allocatable **parm::tmpstdmx**  
*standard deviation for avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable **parm::pcf**  
*normalization coefficient for precipitation generated from skewed distribution (none)*
- real \*8, dimension(:,:), allocatable **parm::tmpmn**  
*avg monthly minimum air temperature (deg C)*

- real \*8, dimension(:,:), allocatable [parm::tmpmx](#)  
*avg monthly maximum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::tmpstdmn](#)  
*standard deviation for avg monthly minimum air temperature (deg C)*
- real \*8, dimension(:,:), allocatable [parm::otmpstdmn](#)
- real \*8, dimension(:,:), allocatable [parm::otmpmn](#)
- real \*8, dimension(:,:), allocatable [parm::otmpmx](#)
- real \*8, dimension(:,:), allocatable [parm::otmpstdmx](#)
- real \*8, dimension(:,:), allocatable [parm::ch\\_erodmo](#)
- real \*8, dimension(:,:), allocatable [parm::uh](#)
- real \*8, dimension(:,:), allocatable [parm::hqdsave](#)
- real \*8, dimension(:,:), allocatable [parm::hsdsave](#)
- real \*8, dimension(:,:), allocatable [parm::pr\\_w1](#)  
*probability of wet day after dry day in month (none)*
- real \*8, dimension(:,:), allocatable [parm::pr\\_w2](#)  
*probability of wet day after wet day in month (none)*
- real \*8, dimension(:,:), allocatable [parm::pr\\_w3](#)  
*proportion of wet days in the month (none)*
- real \*8, dimension(:,:,:), allocatable [parm::pcp\\_stat](#)
- real \*8, dimension(:,:), allocatable [parm::opr\\_w1](#)
- real \*8, dimension(:,:), allocatable [parm::opr\\_w2](#)
- real \*8, dimension(:,:), allocatable [parm::opr\\_w3](#)
- real \*8, dimension(:,:,:), allocatable [parm::opcp\\_stat](#)
- integer, dimension(:), allocatable [parm::ireg](#)  
*precipitation category (none):*  
*1 precipitation <= 508 mm/yr*  
*2 precipitation > 508 and <= 1016 mm/yr*  
*3 precipitation > 1016 mm/yr*
- integer, dimension(:), allocatable [parm::hrutot](#)  
*number of HRUs in subbasin (none)*
- integer, dimension(:), allocatable [parm::hru1](#)
- integer, dimension(:), allocatable [parm::ihgage](#)  
*HRU relative humidity data code (gage # for relative humidity data used in as HRU) (none)*
- integer, dimension(:), allocatable [parm::isgage](#)  
*HRU solar radiation data code (record # for solar radiation used in HRU) (none)*
- integer, dimension(:), allocatable [parm::iwgage](#)  
*HRU wind speed gage data code (gage # for wind speed data used in HRU) (none)*
- integer, dimension(:), allocatable [parm::subgis](#)  
*GIS code printed to output files (output.sub, .rch) (none)*
- integer, dimension(:), allocatable [parm::irgage](#)  
*subbasin rain gage data code (gage # for rainfall data used in HRU) (none)*
- integer, dimension(:), allocatable [parm::itgage](#)  
*subbasin temp gage data code (gage # for temperature data used in HRU) (none)*
- integer, dimension(:), allocatable [parm::irelh](#)  
*(none) irelh = 0 (dewpoint)*  
*irelh = 1 (relative humidity)*  
*note: inputs > 1.0 (dewpoint)*  
*inputs < 1.0 (relative hum)*
- integer, dimension(:), allocatable [parm::fcst\\_reg](#)
- real \*8, dimension(:,:), allocatable [parm::sol\\_aorgn](#)  
*amount of nitrogen stored in the active organic (humic) nitrogen pool in soil layer (kg N/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_fon](#)  
*amount of nitrogen stored in the fresh organic (residue) pool in soil layer (kg N/ha)*

- real \*8, dimension(:,:), allocatable [parm::sol\\_tmp](#)  
*average temperature of soil layer on previous day or  
daily average temperature of soil layer (deg C)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_aws](#)  
*available water capacity of soil layer (mm H2O/mm soil)*
- real \*8, dimension(:,:), allocatable [parm::volcr](#)  
*crack volume for soil layer (mm)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_prk](#)  
*percolation storage from soil layer on current day (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::pperco\\_sub](#)  
*subbasin phosphorus percolation coefficient. Ratio of soluble phosphorus in surface to soluble phosphorus in percolate*
- real \*8, dimension(:,:), allocatable [parm::sol\\_stap](#)  
*amount of phosphorus in the soil layer stored in the stable mineral phosphorus pool (kg P/ha)*
- real \*8, dimension(:,:), allocatable [parm::conv\\_wt](#)  
*factor which converts kg/kg soil to kg/ha (none)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_actp](#)  
*amount of phosphorus stored in the active mineral phosphorus pool (kg P/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_solp](#)  
*soluble P concentration in top soil layer (mg P/kg soil) or  
amount of inorganic phosphorus stored in solution in soil layer. NOTE UNIT CHANGE! (kg P/ha)*
- real \*8, dimension(:,:), allocatable [parm::crdep](#)  
*maximum or potential crack volume (mm)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_fc](#)  
*amount of water available to plants in soil layer at field capacity (fc - wp water) (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_ul](#)  
*amount of water held in the soil layer at saturation (sat - wp water) (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_bd](#)  
*bulk density of the soil layer in HRU ( $\text{Mg/m}^3$ )*
- real \*8, dimension(:,:), allocatable [parm::sol\\_z](#)  
*depth to bottom of each soil profile layer in a given HRU (mm)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_st](#)  
*amount of water stored in the soil layer on any given day (less wilting point water) (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_up](#)  
*water content of soil at -0.033 MPa (field capacity) (mm H2O/mm soil)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_clay](#)  
*percent clay content in soil layer in HRU (UNIT CHANGE!) (% or none)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_hk](#)  
*beta coefficient to calculate hydraulic conductivity (none)*
- real \*8, dimension(:,:), allocatable [parm::flat](#)  
*lateral flow storage in soil layer on current day (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_nh3](#)  
*amount of nitrogen stored in the ammonium pool in soil layer (kg N/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_ec](#)  
*electrical conductivity of soil layer (dS/m)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_orgn](#)  
*amount of nitrogen stored in the stable organic N pool. NOTE UNIT CHANGE! (mg N/kg soil or kg N/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_por](#)  
*total porosity of soil layer expressed as a fraction of the total volume (none)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_wp](#)  
*water content of soil at -1.5 MPa (wilting point) (mm H2O/mm soil)*

- real \*8, dimension(:,:), allocatable [parm::sol\\_orgp](#)  
*amount of phosphorus stored in the organic P pool in soil layer. NOTE UNIT CHANGE! (mg P/kg soil or kg P/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_hum](#)  
*amount of organic matter in the soil layer classified as humic substances (kg humus/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_wpmm](#)  
*water content of soil at -1.5 MPa (wilting point) (mm H2O)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_no3](#)  
*amount of nitrogen stored in the nitrate pool in the soil layer. This variable is read in as a concentration and converted to kg/ha (this value is read from the .sol file in units of mg/kg) (kg N/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_cbn](#)  
*percent organic carbon in soil layer (%)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_k](#)  
*saturated hydraulic conductivity of soil layer (mm/hour)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_rsd](#)  
*amount of organic matter in the soil layer classified as residue (kg/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_fop](#)  
*amount of phosphorus stored in the fresh organic (residue) pool in soil layer (kg P/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_rock](#)  
*percent of rock fragments in soil layer (%)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_silt](#)  
*percent silt content in soil material (UNIT CHANGE!) (% or none)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_sand](#)  
*percent sand content of soil material (%)*
- real \*8, dimension(:,:), allocatable [parm::orig\\_solno3](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solorgn](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solsofp](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solorgp](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_soltmp](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solrsd](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solfop](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solfon](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solaorgn](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solst](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solactp](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_solstap](#)
- real \*8, dimension(:,:), allocatable [parm::orig\\_volcr](#)
- real \*8, dimension(:,:), allocatable [parm::conk](#)  
*lateral saturated hydraulic conductivity for each profile layer in a give HRU. For example (conk(2,1) is conductivity of layer from sol\_z(1,1) to sol\_z(2,1) in HRU1 (mm/hr)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_pst](#)  
*sol\_pst(:,1) initial amount of pesticide in first layer read in from .chm file (mg/kg)*  
*sol\_pst(:,,:) amount of pesticide in soil layer. NOTE UNIT CHANGE! (kg/ha)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_kp](#)  
*pesticide sorption coefficient, Kp; the ratio of the concentration in the solid phase to the concentration in solution ((mg/kg)/(mg/L) or m<sup>3</sup>/ton)*
- real \*8, dimension(:,:), allocatable [parm::orig\\_solpst](#)
- real \*8, dimension(:), allocatable [parm::velsetlr](#)
- real \*8, dimension(:), allocatable [parm::velsetlp](#)
- real \*8, dimension(:), allocatable [parm::br1](#)  
*1st shape parameter for reservoir surface area equation (none)*
- real \*8, dimension(:), allocatable [parm::evrsv](#)  
*lake evaporation coefficient (none)*
- real \*8, dimension(:), allocatable [parm::res\\_k](#)



- hydraulic conductivity of the reservoir bottom (mm/hr)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_conc](#)  
*pesticide concentration in lake water (mg/m<sup>3</sup>)*
- volume of water needed to fill the reservoir to the emergency spillway (read in as 10<sup>4</sup> m<sup>3</sup> and converted to m<sup>3</sup>) (m<sup>3</sup>)*

  - real \*8, dimension(:), allocatable [parm::res\\_evol](#)
- volume of water needed to fill the reservoir to the principal spillway (read in as 10<sup>4</sup> m<sup>3</sup> and converted to m<sup>3</sup>) (m<sup>3</sup>)*

  - real \*8, dimension(:), allocatable [parm::res\\_pvol](#)
- reservoir volume (read in as 10<sup>4</sup> m<sup>3</sup> and converted to m<sup>3</sup>) (m<sup>3</sup>)*

  - real \*8, dimension(:), allocatable [parm::res\\_vol](#)
- reservoir surface area when reservoir is filled to principal spillway (ha)*

  - real \*8, dimension(:), allocatable [parm::res\\_psa](#)
- pesticide reaction coefficient in lake water (1/day)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_rea](#)
- pesticide volatilization coefficient in lake water (m/day)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_vol](#)
- 2nd shape parameter for reservoir surface area equation (none)*

  - real \*8, dimension(:), allocatable [parm::br2](#)
- average daily principal spillway release volume (read in as a release rate in m<sup>3</sup>/s and converted to m<sup>3</sup>/day) (m<sup>3</sup>/day)*

  - real \*8, dimension(:), allocatable [parm::res\\_rr](#)
- amount of sediment in reservoir (read in as mg/L and converted to kg/L) (kg/L)*

  - real \*8, dimension(:), allocatable [parm::res\\_sed](#)
- pesticide partition coefficient between water and sediment in lake water (m<sup>3</sup>/g)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_koc](#)
- mixing velocity (diffusion/dispersion) in lake water for pesticide (m/day)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_mix](#)
- resuspension velocity in lake water for pesticide sorbed to sediment (m/day)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_rsp](#)
- settling velocity in lake water for pesticide sorbed to sediment (m/day)*

  - real \*8, dimension(:), allocatable [parm::lkpst\\_stl](#)
- pesticide concentration in lake bed sediment (mg/m<sup>3</sup>)*

  - real \*8, dimension(:), allocatable [parm::lkspst\\_conc](#)
- pesticide reaction coefficient in lake bed sediment (1/day)*

  - real \*8, dimension(:), allocatable [parm::lkspst\\_rea](#)
- real \*8, dimension(:), allocatable [parm::theta\\_n](#)
- real \*8, dimension(:), allocatable [parm::theta\\_p](#)
- real \*8, dimension(:), allocatable [parm::con\\_nirr](#)
- real \*8, dimension(:), allocatable [parm::con\\_pirr](#)
- real \*8, dimension(:), allocatable [parm::lkspst\\_act](#)  
*depth of active sediment layer in lake for for pesticide (m)*
- pesticide burial velocity in lake bed sediment (m/day)*

  - real \*8, dimension(:), allocatable [parm::lkspst\\_bry](#)
- real \*8, dimension(:), allocatable [parm::sed\\_stlr](#)
- real \*8, dimension(7) [parm::resdata](#)
- normal amount of sediment in reservoir (read in as mg/L and convert to kg/L) (kg/L)*

  - real \*8, dimension(:), allocatable [parm::res\\_nsed](#)
- fraction of water removed from the reservoir via WURES which is returned and becomes flow from the reservoir outlet (none)*

  - real \*8, dimension(:), allocatable [parm::wurtmf](#)
- real \*8, dimension(:), allocatable [parm::chlar](#)



- chlorophyll-a production coefficient for reservoir (none)*
- real \*8, dimension(:), allocatable [parm::res\\_no3](#)  
*amount of nitrate in reservoir (kg N)*
- real \*8, dimension(:), allocatable [parm::res\\_orn](#)  
*amount of organic N in reservoir (kg N)*
- real \*8, dimension(:), allocatable [parm::res\\_orgp](#)  
*amount of organic P in reservoir (kg P)*
- real \*8, dimension(:), allocatable [parm::res\\_solp](#)  
*amount of soluble P in reservoir (kg P)*
- real \*8, dimension(:), allocatable [parm::res\\_seci](#)  
*secchi-disk depth (m)*
- real \*8, dimension(:), allocatable [parm::res\\_chla](#)
- real \*8, dimension(:), allocatable [parm::res\\_esa](#)  
*reservoir surface area when reservoir is filled to emergency spillway (ha)*
- real \*8, dimension(:), allocatable [parm::res\\_nh3](#)  
*amount of ammonia in reservoir (kg N)*
- real \*8, dimension(:), allocatable [parm::res\\_no2](#)  
*amount of nitrite in reservoir (kg N)*
- real \*8, dimension(:), allocatable [parm::seccir](#)  
*water clarity coefficient for reservoir (none)*
- real \*8, dimension(:), allocatable [parm::res\\_bactp](#)
- real \*8, dimension(:), allocatable [parm::res\\_bactlp](#)
- real \*8, dimension(:), allocatable [parm::oflowmn\\_fps](#)  
*minimum reservoir outflow as a fraction of the principal spillway volume (0-1) (fraction)*
- real \*8, dimension(:), allocatable [parm::starg\\_fps](#)  
*target volume as a fraction of the principal spillway volume (.1-5) (fraction)*
- real \*8, dimension(:), allocatable [parm::weirc](#)
- real \*8, dimension(:), allocatable [parm::weirk](#)
- real \*8, dimension(:), allocatable [parm::weirw](#)
- real \*8, dimension(:), allocatable [parm::acoef](#)
- real \*8, dimension(:), allocatable [parm::bcoef](#)
- real \*8, dimension(:), allocatable [parm::ccoef](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resvol](#)
- real \*8, dimension(:), allocatable [parm::orig\\_ressed](#)
- real \*8, dimension(:), allocatable [parm::orig\\_lkpstconc](#)
- real \*8, dimension(:), allocatable [parm::orig\\_lkspstconc](#)
- real \*8, dimension(:), allocatable [parm::orig\\_ressolp](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resorgp](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resno3](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resno2](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resnh3](#)
- real \*8, dimension(:), allocatable [parm::orig\\_resorgn](#)
- real \*8, dimension(:,:), allocatable [parm::oflowmn](#)  
*minimum daily outflow for the month (read in as  $m^3/s$  and converted to  $m^3/day$ ) ( $m^3/day$ )*
- real \*8, dimension(:,:), allocatable [parm::oflowmx](#)  
*maximum daily outflow for the month (read in as  $m^3/s$  and converted to  $m^3/day$ ) ( $m^3/day$ )*
- real \*8, dimension(:,:), allocatable [parm::starg](#)  
*monthly target reservoir storage (needed if IRESCO=2) (read in as  $10^4 m^3$  and converted to  $m^3$ ) ( $m^3$ )*
- real \*8, dimension(:), allocatable [parm::psetlr1](#)  
*phosphorus settling rate for mid-year period (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*
- real \*8, dimension(:), allocatable [parm::psetlr2](#)  
*phosphorus settling rate for remainder of year (read in as  $m/year$  and converted to  $m/day$ ) ( $m/day$ )*

- real \*8, dimension(:), allocatable [parm::nsetlr1](#)  
*nitrogen settling rate for mid-year period (read in as m/year and converted to m/day) (m/day)*
- real \*8, dimension(:), allocatable [parm::nsetlr2](#)  
*nitrogen settling rate for remainder of year (read in as m/year and converted to m/day) (m/day)*
- real \*8, dimension(:,:), allocatable [parm::wuresn](#)  
*average amount of water withdrawn from reservoir each month for consumptive water use (read in as  $10^4 \text{ m}^3$  and converted to  $\text{m}^3$ ) ( $\text{m}^3$ )*
- real \*8, dimension(:,:), allocatable [parm::res\\_out](#)  
*measured average daily outflow from the reservoir for the month (needed if IRESCO=1) (read in as  $\text{m}^3/\text{s}$  and converted to  $\text{m}^3/\text{day}$ ) ( $\text{m}^3/\text{day}$ )*
- integer, dimension(:), allocatable [parm::res\\_sub](#)  
*number of subbasin reservoir is in (weather for the subbasin is used for the reservoir) (none)*
- integer, dimension(:), allocatable [parm::ires1](#)  
*beginning of mid-year nutrient settling "season" (none)*
- integer, dimension(:), allocatable [parm::ires2](#)  
*end of mid-year nutrient settling "season" (none)*
- integer, dimension(:), allocatable [parm::iresco](#)  
*outflow simulation code (none):*  
*0 compute outflow for uncontrolled reservoir with average annual release rate*  
*1 measured monthly outflow*  
*2 simulated controlled outflow-target release*  
*3 measured daily outflow*  
*4 stage/volume/outflow relationship*
- integer, dimension(:), allocatable [parm::iyres](#)  
*year of the simulation that the reservoir becomes operational (none)*
- integer, dimension(:), allocatable [parm::mores](#)  
*month the reservoir becomes operational (none)*
- integer, dimension(:), allocatable [parm::iflod1r](#)  
*beginning month of non-flood season (needed if IRESCO=2) (none)*
- integer, dimension(:), allocatable [parm::iflod2r](#)  
*ending month of non-flood season (needed if IRESCO=2) (none)*
- integer, dimension(:), allocatable [parm::ndtargr](#)  
*number of days to reach target storage from current reservoir storage (needed if IRESCO=2) (days)*
- real \*8, dimension(:), allocatable [parm::ap\\_ef](#)  
*application efficiency (0-1) (none)*
- real \*8, dimension(:), allocatable [parm::decay\\_f](#)  
*exponential of the rate constant for degradation of the pesticide on foliage (none)*
- real \*8, dimension(:), allocatable [parm::skoc](#)  
*soil adsorption coefficient normalized for soil organic carbon content  $((\text{mg}/\text{kg})/(\text{mg}/\text{L}))$*
- real \*8, dimension(:), allocatable [parm::decay\\_s](#)  
*exponential of the rate constant for degradation of the pesticide in soil (none)*
- real \*8, dimension(:), allocatable [parm::hlife\\_f](#)  
*half-life of pesticide on foliage (days)*
- real \*8, dimension(:), allocatable [parm::hlife\\_s](#)  
*half-life of pesticide in soil (days)*
- real \*8, dimension(:), allocatable [parm::pst\\_wof](#)  
*fraction of pesticide on foliage which is washed-off by a rainfall event (none)*
- real \*8, dimension(:), allocatable [parm::pst\\_wsol](#)  
*solubility of chemical in water (mg/L (ppm))*
- real \*8, dimension(:), allocatable [parm::irramt](#)  
*depth of irrigation water applied to HRU (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::phusw](#)

- real \*8, dimension(:), allocatable **parm::phusw\_nocrop**
- integer, dimension(:), allocatable **parm::pstflg**  
*flag for types of pesticide used in watershed. Array location is pesticide ID number*  
*0: pesticide not used*  
*1: pesticide used*
- integer, dimension(:), allocatable **parm::nope**  
*sequence number of pesticide in NPNO(:) (none)*
- integer, dimension(:), allocatable **parm::nop**
- integer, dimension(:), allocatable **parm::yr\_skip**
- integer, dimension(:), allocatable **parm::isweep**
- integer, dimension(:), allocatable **parm::icrmx**
- integer, dimension(:), allocatable **parm::nopmx**
- integer, dimension(:, :), allocatable **parm::mgtop**
- integer, dimension(:, :), allocatable **parm::idop**
- integer, dimension(:, :), allocatable **parm::mgt1iop**
- integer, dimension(:, :), allocatable **parm::mgt2iop**
- integer, dimension(:, :), allocatable **parm::mgt3iop**
- real \*8, dimension(:, :), allocatable **parm::mgt4op**
- real \*8, dimension(:, :), allocatable **parm::mgt5op**
- real \*8, dimension(:, :), allocatable **parm::mgt6op**
- real \*8, dimension(:, :), allocatable **parm::mgt7op**
- real \*8, dimension(:, :), allocatable **parm::mgt8op**
- real \*8, dimension(:, :), allocatable **parm::mgt9op**
- real \*8, dimension(:, :), allocatable **parm::mgt10iop**
- real \*8, dimension(:, :), allocatable **parm::phu\_op**
- real \*8, dimension(:), allocatable **parm::cnyld**  
*fraction of nitrogen in yield (kg N/kg yield)*
- real \*8, dimension(:), allocatable **parm::rsdco\_pl**  
*plant residue decomposition coefficient. The fraction of residue which will decompose in a day assuming optimal moisture, temperature, C:N ratio, and C:P ratio (none)*
- real \*8, dimension(:), allocatable **parm::wac21**  
*1st shape parameter for radiation use efficiency equation (none)*
- real \*8, dimension(:), allocatable **parm::wac22**  
*2nd shape parameter for radiation use efficiency equation (none)*
- real \*8, dimension(:), allocatable **parm::alai\_min**  
*minimum LAI during winter dormant period ( $m^2/m^2$ )*
- real \*8, dimension(:), allocatable **parm::leaf1**  
*1st shape parameter for leaf area development equation (none)*
- real \*8, dimension(:), allocatable **parm::leaf2**  
*2nd shape parameter for leaf area development equation (none)*
- real \*8, dimension(:), allocatable **parm::wsyf**  
*Value of harvest index between 0 and HVSTI which represents the lowest value expected due to water stress ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable **parm::bio\_e**  
*biomass-energy ratio. The potential (unstressed) growth rate per unit of intercepted photosynthetically active radiation ((kg/ha)/(MJ/m\*\*2))*
- real \*8, dimension(:), allocatable **parm::hvsti**  
*harvest index: crop yield/aboveground biomass ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable **parm::t\_base**  
*minimum temperature for plant growth (deg C)*
- real \*8, dimension(:), allocatable **parm::t\_opt**  
*optimal temperature for plant growth (deg C)*
- real \*8, dimension(:), allocatable **parm::chtmx**

- maximum canopy height (m)*
- real \*8, dimension(:), allocatable [parm::cvm](#)
  - natural log of USLE\_C (the minimum value of the USLE C factor for the land cover) (none)*
- real \*8, dimension(:), allocatable [parm::gsi](#)
  - maximum stomatal conductance (m/s)*
- real \*8, dimension(:), allocatable [parm::vpd2](#)
  - rate of decline in stomatal conductance per unit increase in vapor pressure deficit ((m/s)\*(1/kPa))*
- real \*8, dimension(:), allocatable [parm::wavp](#)
  - rate of decline in radiation use efficiency as a function of vapor pressure deficit (none)*
- real \*8, dimension(:), allocatable [parm::bio\\_leaf](#)
  - fraction of leaf/needle biomass that drops during dormancy (for trees only) (none)*
- real \*8, dimension(:), allocatable [parm::blai](#)
  - maximum (potential) leaf area index (none)*
- real \*8, dimension(:), allocatable [parm::cpyld](#)
  - fraction of phosphorus in yield (kg P/kg yield)*
- real \*8, dimension(:), allocatable [parm::dlai](#)
  - fraction of growing season when leaf area declines (none)*
- real \*8, dimension(:), allocatable [parm::rdmx](#)
  - maximum root depth of plant (m)*
- real \*8, dimension(:), allocatable [parm::bio\\_n1](#)
  - 1st shape parameter for plant N uptake equation (none)*
- real \*8, dimension(:), allocatable [parm::bio\\_n2](#)
  - 2nd shape parameter for plant N uptake equation (none)*
- real \*8, dimension(:), allocatable [parm::bio\\_p1](#)
  - 1st shape parameter for plant P uptake equation (none)*
- real \*8, dimension(:), allocatable [parm::bio\\_p2](#)
  - 2st shape parameter for plant P uptake equation (none)*
- real \*8, dimension(:), allocatable [parm::bm\\_dieoff](#)
  - fraction above ground biomass that dies off at dormancy (fraction)*
- real \*8, dimension(:), allocatable [parm::bmx\\_trees](#)
- real \*8, dimension(:), allocatable [parm::ext\\_coef](#)
- real \*8, dimension(:), allocatable [parm::rsr1](#)
  - initial root to shoot ratio at the beg of growing season*
- real \*8, dimension(:), allocatable [parm::rsr2](#)
  - root to shoot ratio at the end of the growing season*
- real \*8, dimension(:), allocatable [parm::pltnfr1](#)
  - nitrogen uptake parameter #1: normal fraction of N in crop biomass at emergence (kg N/kg biomass)*
- real \*8, dimension(:), allocatable [parm::pltnfr2](#)
  - nitrogen uptake parameter #2: normal fraction of N in crop biomass at 0.5 maturity (kg N/kg biomass)*
- real \*8, dimension(:), allocatable [parm::pltnfr3](#)
  - nitrogen uptake parameter #3: normal fraction of N in crop biomass at maturity (kg N/kg biomass)*
- real \*8, dimension(:), allocatable [parm::pltpfr1](#)
  - phosphorus uptake parameter #1: normal fraction of P in crop biomass at emergence (kg P/kg biomass)*
- real \*8, dimension(:), allocatable [parm::pltpfr2](#)
  - phosphorus uptake parameter #2: normal fraction of P in crop biomass at 0.5 maturity (kg P/kg biomass)*
- real \*8, dimension(:), allocatable [parm::pltpfr3](#)
  - phosphorus uptake parameter #3: normal fraction of P in crop biomass at maturity (kg P/kg biomass)*
- integer, dimension(:), allocatable [parm::idc](#)

- crop/landcover category (none):*  
 1 warm season annual legume  
 2 cold season annual legume  
 3 perennial legume  
 4 warm season annual  
 5 cold season annual  
 6 perennial  
 7 trees
- integer, dimension(:), allocatable **parm::mat\_ys**
  - real \*8, dimension(:), allocatable **parm::bactpdb**  
*concentration of persistent bacteria in manure (fertilizer) (cfu/g manure)*
  - real \*8, dimension(:), allocatable **parm::fminn**  
*fraction of fertilize/manure that is mineral nitrogen (NO<sub>3</sub> + NH<sub>3</sub>) (kg minN/kg fert)*
  - real \*8, dimension(:), allocatable **parm::forgn**  
*fraction of organic nitrogen in fertilizer/manure (kg orgN/kg fert)*
  - real \*8, dimension(:), allocatable **parm::forgp**  
*fraction of fertilizer/manure that is organic phosphorus (kg orgP/kg fert)*
  - real \*8, dimension(:), allocatable **parm::bactkddb**  
*fraction of bacteria in solution (the remaining fraction is sorbed to soil particles) (none):*  
 1: all bacteria in solution  
 0: all bacteria sorbed to soil particles
  - real \*8, dimension(:), allocatable **parm::bactlpdb**  
*concentration of less persistent bacteria in manure (fertilizer) (cfu/g manure)*
  - real \*8, dimension(:), allocatable **parm::fminp**  
*fraction of fertilizer that is mineral phosphorus in fertilizer/manure (kg minP/kg fert)*
  - real \*8, dimension(:), allocatable **parm::fnh3n**  
*fraction of mineral N content that is NH<sub>3</sub>-N in fertilizer/manure (kg NH<sub>3</sub>-N/kg minN)*
  - character(len=8), dimension(200) **parm::fertnm**  
*name of fertilizer*
  - real \*8, dimension(:), allocatable **parm::curbden**  
*curb length density in HRU (km/ha)*
  - real \*8, dimension(:), allocatable **parm::dirtmx**  
*maximum amount of solids allowed to build up on impervious surfaces (kg/curb km)*
  - real \*8, dimension(:), allocatable **parm::fimp**  
*fraction of HRU area that is impervious (both directly and indirectly connected)(fraction)*
  - real \*8, dimension(:), allocatable **parm::urbcoef**  
*wash-off coefficient for removal of constituents from an impervious surface (1/mm)*
  - real \*8, dimension(:), allocatable **parm::thalf**  
*time for the amount of solids on impervious areas to build up to 1/2 the maximum level (days)*
  - real \*8, dimension(:), allocatable **parm::tnconc**  
*concentration of total nitrogen in suspended solid load from impervious areas (mg N/kg sed)*
  - real \*8, dimension(:), allocatable **parm::tno3conc**  
*concentration of NO<sub>3</sub>-N in suspended solid load from impervious areas (mg NO<sub>3</sub>-N/kg sed)*
  - real \*8, dimension(:), allocatable **parm::tpconc**  
*concentration of total phosphorus in suspended solid load from impervious areas (mg P/kg sed)*
  - real \*8, dimension(:), allocatable **parm::fcimp**  
*fraction of HRU area that is classified as directly connected impervious (fraction)*
  - real \*8, dimension(:), allocatable **parm::urbcn2**  
*SCS curve number for moisture condition II in impervious areas (none)*
  - real \*8 **parm::fr\_curb**  
*availability factor, the fraction of the curb length that is sweepable (none)*
  - real \*8 **parm::frt\_kg**  
*amount of fertilizer applied to HRU (kg/ha)*

- real \*8 **parm::pst\_dep**  
*depth of pesticide in the soil (mm)*
- real \*8 **parm::swepeff**
- real \*8, dimension(:), allocatable **parm::ranrns\_hru**  
*random roughness for a given HRU (mm)*
- integer, dimension(:), allocatable **parm::itill**
- real \*8, dimension(:), allocatable **parm::deptil**  
*depth of mixing caused by tillage operation (mm)*
- real \*8, dimension(:), allocatable **parm::effmix**  
*mixing efficiency of tillage operation (none)*
- real \*8, dimension(:), allocatable **parm::ranrns**  
*random roughness of a given tillage operation (mm)*
- character(len=8), dimension(550) **parm::tillnm**  
*8-character name for the tillage operation*
- real \*8, dimension(:), allocatable **parm::rnum1s**  
*For ICODES equal to (none)*  
*0,1,3,5,9: not used*  
*2: Fraction of flow in channel*  
*4: amount of water transferred (as defined by INUM4S)*  
*7,8,10,11: drainage area in square kilometers associated with the record file*  
*12: reparation coefficient.*
- real \*8, dimension(:), allocatable **parm::hyd\_dakm**  
*total drainage area of hydrograph in square kilometers (km<sup>2</sup>)*
- real \*8, dimension(:,:), allocatable **parm::varoute**
- real \*8, dimension(:,:), allocatable **parm::shyd**
- real \*8, dimension(:,:), allocatable **parm::vartran**
- real \*8, dimension(:,:,:), allocatable **parm::hhvaroute**
- integer, dimension(:), allocatable **parm::icodes**  
*routing command code (none):*  
*0 = finish*  
*1 = subbasin*  
*2 = route*  
*3 = routres*  
*4 = transfer*  
*5 = add*  
*6 = rechour*  
*7 = recmon*  
*8 = recyear*  
*9 = save*  
*10 = recday*  
*11 = recnst*  
*12 = structure*  
*13 = apex*  
*14 = saveconc*  
*15 =*
- integer, dimension(:), allocatable **parm::ihouts**  
*For ICODES equal to (none)*  
*0: not used*  
*1,2,3,5,7,8,10,11: hydrograph storage location number*  
*4: departure type (1=reach, 2=reservoir)*  
*9: hydrograph storage location of data to be printed to event file*  
*14:hydrograph storage location of data to be printed to saveconc file.*
- integer, dimension(:), allocatable **parm::inum1s**  
*For ICODES equal to (none)*  
*0: not used*  
*1: subbasin number*  
*2: reach number*

- 3: reservoir number
- 4: reach or res # flow is diverted from
- 5: hydrograph storage location of 1st dataset to be added
- 7,8,9,10,11,14: file number.
- integer, dimension(:), allocatable **parm::inum2s**
  - For ICODES equal to (none)
  - 0,1,7,8,10,11: not used
  - 2,3: inflow hydrograph storage location
  - 4: destination type (1=reach, 2=reservoir)
  - 5: hydrograph storage location of 2nd dataset to be added
  - 9,14: print frequency (0=daily, 1=hourly)
- integer, dimension(:), allocatable **parm::inum3s**
  - For ICODES equal to (none)
  - 0,1,5,7,8,10,11: not used
  - 2,3: subbasin number 4: destination number. Reach or reservoir receiving water
  - 9: print format (0=normal, fixed format; 1=txt format for AV interface, recday)
- integer, dimension(:), allocatable **parm::inum4s**
  - For ICODES equal to (none)
  - 0,2,3,5,7,8,9,10,11: not used
  - 1: GIS code printed to output file (optional)
  - 4: rule code governing transfer of water (1=fraction transferred out, 2=min volume or flow left, 3=exact amount transferred)
- integer, dimension(:), allocatable **parm::inum5s**
- integer, dimension(:), allocatable **parm::inum6s**
- integer, dimension(:), allocatable **parm::inum7s**
- integer, dimension(:), allocatable **parm::inum8s**
- integer, dimension(:), allocatable **parm::subed**
- character(len=10), dimension(:), allocatable **parm::recmonps**
- character(len=10), dimension(:), allocatable **parm::reccnstps**
- character(len=5), dimension(:), allocatable **parm::subnum**
- character(len=4), dimension(:), allocatable **parm::hruno**
- real \*8, dimension(:), allocatable **parm::grwat\_n**
  - Mannings's n for grassed waterway (none)
- integer, dimension(:), allocatable **parm::grwat\_i**
  - flag for the simulation of grass waterways (none)
  - = 0 inactive
  - = 1 active
- real \*8, dimension(:), allocatable **parm::grwat\_l**
  - length of grass waterway (km)
- real \*8, dimension(:), allocatable **parm::grwat\_w**
  - average width of grassed waterway (m)
- real \*8, dimension(:), allocatable **parm::grwat\_d**
  - depth of grassed waterway from top of bank to bottom (m)
- real \*8, dimension(:), allocatable **parm::grwat\_s**
  - average slope of grassed waterway channel (m)
- real \*8, dimension(:), allocatable **parm::grwat\_spcon**
  - linear parameter for calculating sediment in grassed waterways (none)
- real \*8, dimension(:), allocatable **parm::tc\_gwat**
- real \*8, dimension(:), allocatable **parm::pot\_volmm**
- real \*8, dimension(:), allocatable **parm::pot\_tilemm**
- real \*8, dimension(:), allocatable **parm::pot\_volxmm**
- real \*8, dimension(:), allocatable **parm::pot\_fr**
  - fraction of HRU area that drains into pothole ( $\text{km}^2/\text{km}^2$ )
- real \*8, dimension(:), allocatable **parm::pot\_tile**



- average daily outflow to main channel from tile flow if drainage tiles are installed in pothole (needed only if current HRU is IPOT) ( $m^3/s$ )*
- real \*8, dimension(:), allocatable **parm::pot\_vol**  
*initial or current volume of water stored in the depression/impounded area (read in as mm and converted to  $m^3$ ) (needed only if current HRU is IPOT) (mm or  $m^3 H_2O$ )*
  - real \*8, dimension(:), allocatable **parm::potsa**  
*surface area of impounded water body (ha)*
  - real \*8, dimension(:), allocatable **parm::pot\_volx**  
*maximum volume of water stored in the depression/impounded area (read in as mm and converted to  $m^3$ ) (needed only if current HRU is IPOT) (mm)*
  - real \*8, dimension(:), allocatable **parm::wfs**  
*wetting front matric potential (average capillary suction at wetting front) (mm)*
  - real \*8, dimension(:), allocatable **parm::potflwi**
  - real \*8, dimension(:), allocatable **parm::potsedi**
  - real \*8, dimension(:), allocatable **parm::pot\_no3l**  
*nitrate decay rate in impounded area (1/day)*
  - real \*8, dimension(:), allocatable **parm::pot\_nsed**  
*normal sediment concentration in impounded water (needed only if current HRU is IPOT)(mg/L)*
  - real \*8, dimension(:), allocatable **parm::gwno3**  
*nitrate-N concentration in groundwater loading to reach (mg N/L)*
  - real \*8, dimension(:), allocatable **parm::newrti**  
*infiltration rate for last time step from the previous day (mm/hr)*
  - real \*8, dimension(:), allocatable **parm::fsred**  
*reduction in bacteria loading from filter strip (none)*
  - real \*8, dimension(:), allocatable **parm::pot\_sed**
  - real \*8, dimension(:), allocatable **parm::pot\_no3**
  - real \*8, dimension(:), allocatable **parm::tmpavp**
  - real \*8, dimension(:), allocatable **parm::dis\_stream**  
*average distance to stream (m)*
  - real \*8, dimension(:), allocatable **parm::evpot**  
*pothole evaporation coefficient (none)*
  - real \*8, dimension(:), allocatable **parm::pot\_solpl**
  - real \*8, dimension(:), allocatable **parm::sed\_con**
  - real \*8, dimension(:), allocatable **parm::orgn\_con**
  - real \*8, dimension(:), allocatable **parm::orgp\_con**
  - real \*8, dimension(:), allocatable **parm::pot\_k**  
*hydraulic conductivity of soil surface of pothole defaults to conductivity of upper soil (0.01-10.) layer*
  - real \*8, dimension(:), allocatable **parm::soln\_con**
  - real \*8, dimension(:), allocatable **parm::solp\_con**
  - real \*8, dimension(:), allocatable **parm::n\_reduc**  
*nitrogen uptake reduction factor (not currently used; defaulted 300.)*
  - real \*8, dimension(:), allocatable **parm::n\_lag**  
*lag coefficient for calculating nitrate concentration in subsurface drains (0.001 - 1.0) (dimensionless)*
  - real \*8, dimension(:), allocatable **parm::n\_ln**  
*power function exponent for calculating nitrate concentration in subsurface drains (1.0 - 3.0) (dimensionless)*
  - real \*8, dimension(:), allocatable **parm::n\_inco**  
*coefficient for power function for calculating nitrate concentration in subsurface drains (0.5 - 4.0) (dimensionless)*
  - integer, dimension(:), allocatable **parm::ioper**
  - integer, dimension(:), allocatable **parm::ngrwat**
  - real \*8, dimension(:), allocatable **parm::usle\_ls**  
*USLE equation length slope (LS) factor (none)*



- real \*8, dimension(:), allocatable [parm::filterw](#)  
*filter strip width for bacteria transport (m)*
- real \*8, dimension(:), allocatable [parm::phuacc](#)  
*fraction of plant heat units accumulated (none)*
- real \*8, dimension(:), allocatable [parm::sumix](#)  
*sum of all tillage mixing efficiencies for HRU operation (none)*
- real \*8, dimension(:), allocatable [parm::epco](#)  
*plant water uptake compensation factor (0-1) (none)*
- real \*8, dimension(:), allocatable [parm::esco](#)  
*soil evaporation compensation factor (0-1) (none)*
- real \*8, dimension(:), allocatable [parm::hru\\_slp](#)  
*average slope steepness in HRU (m/m)*
- real \*8, dimension(:), allocatable [parm::slsubbsn](#)  
*average slope length for subbasin (m)*
- real \*8, dimension(:), allocatable [parm::erorgn](#)  
*organic N enrichment ratio, if left blank the model will calculate for every event (none)*
- real \*8, dimension(:), allocatable [parm::erorgp](#)  
*organic P enrichment ratio, if left blank the model will calculate for every event (none)*
- real \*8, dimension(:), allocatable [parm::biomix](#)  
*biological mixing efficiency. Mixing of soil due to activity of earthworms and other soil biota. Mixing is performed at the end of every calendar year (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_seci](#)  
*secchi-disk depth of pond (m)*
- real \*8, dimension(:), allocatable [parm::canmx](#)  
*maximum canopy storage (mm H2O)*
- real \*8, dimension(:), allocatable [parm::divmax](#)  
*maximum daily irrigation diversion from the reach (when IRRSC=1 or IRR=3): when value is positive the units are mm H2O; when the value is negative, the units are  $(10^4 \text{ m}^3 \text{ H2O})$  (mm H2O or  $10^4 \text{ m}^3 \text{ H2O}$ )*
- real \*8, dimension(:), allocatable [parm::flowmin](#)  
*minimum instream flow for irrigation diversions when IRRSC=1, irrigation water will be diverted only when streamflow is at or above FLOWMIN ( $\text{m}^3/\text{s}$ )*
- real \*8, dimension(:), allocatable [parm::usle\\_p](#)  
*USLE equation support practice (P) factor (none)*
- real \*8, dimension(:), allocatable [parm::lat\\_sed](#)  
*sediment concentration in lateral flow (g/L)*
- real \*8, dimension(:), allocatable [parm::rch\\_dakm](#)  
*total drainage area contributing to flow at the outlet (pour point) of the reach in square kilometers ( $\text{km}^2$ )*
- real \*8, dimension(:), allocatable [parm::cn1](#)  
*SCS runoff curve number for moisture condition I (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_no3s](#)  
*amount of nitrate originating from lateral flow in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::lat\\_ttime](#)  
*lateral flow travel time or exponential of the lateral flow travel time (days or none)*
- real \*8, dimension(:), allocatable [parm::cn2](#)  
*SCS runoff curve number for moisture condition II (none)*
- real \*8, dimension(:), allocatable [parm::flowfr](#)  
*fraction of available flow in reach that is allowed to be applied to the HRU (none)*
- real \*8, dimension(:), allocatable [parm::sol\\_zmx](#)  
*maximum rooting depth (mm)*
- real \*8, dimension(:), allocatable [parm::tile\\_ttime](#)  
*exponential of the tile flow travel time (none)*

- real \*8, dimension(:), allocatable [parm::slsoil](#)  
*slope length for lateral subsurface flow (m)*
- real \*8, dimension(:), allocatable [parm::gwminp](#)  
*soluble P concentration in groundwater loading to reach (mg P/L)*
- real \*8, dimension(:), allocatable [parm::sol\\_cov](#)  
*amount of residue on soil surface (kg/ha)*
- real \*8, dimension(:), allocatable [parm::sed\\_stl](#)  
*fraction of sediment remaining suspended in impoundment after settling for one day (kg/kg)*
- real \*8, dimension(:), allocatable [parm::ov\\_n](#)  
*Manning's "n" value for overland flow (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_no3](#)  
*amount of nitrate originating from surface runoff in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::pnd\\_solp](#)  
*amount of soluble P originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::yldanu](#)  
*annual yield (dry weight) in the HRU (metric tons/ha)*
- real \*8, dimension(:), allocatable [parm::driftco](#)  
*coefficient for pesticide drift directly onto stream (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_orgn](#)  
*amount of organic N originating from surface runoff in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::pnd\\_orgp](#)  
*amount of organic P originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::cn3](#)  
*SCS runoff curve number for moisture condition III (none)*
- real \*8, dimension(:), allocatable [parm::twlpnd](#)  
*water lost through seepage from ponds on day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::twlwet](#)  
*water lost through seepage from wetlands on day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::hru\\_fr](#)  
*fraction of subbasin area contained in HRU ( $\text{km}^2/\text{km}^2$ )*
- real \*8, dimension(:), allocatable [parm::sol\\_sumul](#)  
*amount of water held in soil profile at saturation (mm H2O)*
- real \*8, dimension(:), allocatable [parm::pnd\\_chla](#)  
*amount of chlorophyll-a in pond at end of day (kg chl\_a)*
- real \*8, dimension(:), allocatable [parm::hru\\_km](#)  
*area of HRU in square kilometers ( $\text{km}^2$ )*
- real \*8, dimension(:), allocatable [parm::bio\\_ms](#)  
*land cover/crop biomass (dry weight) (kg/ha)*
- real \*8, dimension(:), allocatable [parm::sol\\_alb](#)  
*albedo when soil is moist (none)*
- real \*8, dimension(:), allocatable [parm::strsw](#)  
*fraction of potential plant growth achieved on the day where the reduction is caused by water stress (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_fr](#)  
*fraction of HRU/subbasin area that drains into ponds (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_k](#)  
*hydraulic conductivity through bottom of ponds (mm/hr)*
- real \*8, dimension(:), allocatable [parm::pnd\\_psa](#)  
*surface area of ponds when filled to principal spillway (ha)*
- real \*8, dimension(:), allocatable [parm::pnd\\_pvol](#)  
*runoff volume from catchment area needed to fill the ponds to the principal spillway (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*

- real \*8, dimension(:), allocatable [parm::pnd\\_esa](#)  
*surface area of ponds when filled to emergency spillway (ha)*
- real \*8, dimension(:), allocatable [parm::pnd\\_evol](#)  
*runoff volume from catchment area needed to fill the ponds to the emergency spillway (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [parm::pnd\\_vol](#)  
*volume of water in ponds (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [parm::yldaa](#)  
*average annual yield (dry weight) in the HRU (metric tons)*
- real \*8, dimension(:), allocatable [parm::pnd\\_nsed](#)  
*normal sediment concentration in pond water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [parm::pnd\\_sed](#)  
*sediment concentration in pond water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [parm::dep\\_imp](#)  
*depth to impervious layer (mm)*
- real \*8, dimension(:), allocatable [parm::strsa](#)
- real \*8, dimension(:), allocatable [parm::evpnd](#)
- real \*8, dimension(:), allocatable [parm::evwet](#)
- real \*8, dimension(:), allocatable [parm::wet\\_fr](#)  
*fraction of HRU/subbasin area that drains into wetlands (none)*
- real \*8, dimension(:), allocatable [parm::wet\\_k](#)  
*hydraulic conductivity of bottom of wetlands (mm/hr)*
- real \*8, dimension(:), allocatable [parm::wet\\_nsa](#)  
*surface area of wetlands in subbasin at normal water level (ha)*
- real \*8, dimension(:), allocatable [parm::wet\\_nvol](#)  
*runoff volume from catchment area needed to fill wetlands to normal water level (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- integer, dimension(:), allocatable [parm::iwetgw](#)
- integer, dimension(:), allocatable [parm::iwetile](#)
- real \*8, dimension(:), allocatable [parm::wet\\_mxsa](#)  
*surface area of wetlands at maximum water level (ha)*
- real \*8, dimension(:), allocatable [parm::wet\\_mxvol](#)  
*runoff volume from catchment area needed to fill wetlands to maximum water level (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [parm::wet\\_vol](#)  
*volume of water in wetlands (UNIT CHANGE!) ( $10^4 \text{ m}^3 \text{ H}_2\text{O}$  or  $\text{m}^3 \text{ H}_2\text{O}$ )*
- real \*8, dimension(:), allocatable [parm::wet\\_nsed](#)  
*normal sediment concentration in wetland water (UNIT CHANGE!) (mg/kg or kg/kg)*
- real \*8, dimension(:), allocatable [parm::wet\\_sed](#)  
*sediment concentration in wetland water (UNIT CHANGE!) (mg/L or kg/L)*
- real \*8, dimension(:), allocatable [parm::bp1](#)  
*1st shape parameter for pond surface area equation (none)*
- real \*8, dimension(:), allocatable [parm::bp2](#)  
*2nd shape parameter for the pond surface area equation (none)*
- real \*8, dimension(:), allocatable [parm::sci](#)  
*retention coefficient for CN method based on plant ET (none)*
- real \*8, dimension(:), allocatable [parm::smx](#)  
*retention coefficient for CN method based on soil moisture (none)*
- real \*8, dimension(:), allocatable [parm::bw1](#)  
*1st shape parameter for the wetland surface area equation (none)*
- real \*8, dimension(:), allocatable [parm::bw2](#)  
*2nd shape parameter for the wetland surface area equation (none)*

- real \*8, dimension(:), allocatable [parm::bactpq](#)  
*persistent bacteria in soil solution (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::cnday](#)  
*curve number for current day, HRU and at current soil moisture (none)*
- real \*8, dimension(:), allocatable [parm::bactlp\\_plt](#)  
*less persistent bacteria on foliage (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::bactp\\_plt](#)  
*persistent bacteria on foliage (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::auto\\_eff](#)  
*fertilizer application efficiency calculated as the amount of N applied divided by the amount of N removed at harvest (none)*
- real \*8, dimension(:), allocatable [parm::secciw](#)  
*water clarity coefficient for wetland (none)*
- real \*8, dimension(:), allocatable [parm::sol\\_sw](#)  
*amount of water stored in soil profile on current day (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::bactlpq](#)  
*less persistent bacteria in soil solution (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::chlaw](#)  
*chlorophyll-a production coefficient for wetland (none)*
- real \*8, dimension(:), allocatable [parm::tmpav](#)  
*average air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [parm::bactlps](#)  
*less persistent bacteria attached to soil particles (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::bactps](#)  
*persistent bacteria attached to soil particles (# cfu/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::sno\\_hru](#)  
*amount of water stored as snow in HRU on current day (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::wet\\_orgn](#)  
*amount of organic N originating from surface runoff in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::hru\\_ra](#)  
*solar radiation for the day in HRU (MJ/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::subp](#)  
*precipitation for the day in HRU (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::rsdin](#)  
*initial residue cover (kg/ha)*
- real \*8, dimension(:), allocatable [parm::tmn](#)  
*minimum air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [parm::tmx](#)  
*maximum air temperature on current day in HRU (deg C)*
- real \*8, dimension(:), allocatable [parm::tmp\\_hi](#)
- real \*8, dimension(:), allocatable [parm::tmp\\_lo](#)
- real \*8, dimension(:), allocatable [parm::usle\\_k](#)  
*USLE equation soil erodibility (K) factor (none)*
- real \*8, dimension(:), allocatable [parm::tconc](#)  
*time of concentration for HRU (hour)*
- real \*8, dimension(:), allocatable [parm::hru\\_rmx](#)  
*maximum possible solar radiation for the day in HRU (MJ/m<sup>2</sup>)*
- real \*8, dimension(:), allocatable [parm::rwt](#)  
*fraction of total plant biomass that is in roots (none)*
- real \*8, dimension(:), allocatable [parm::olai](#)
- real \*8, dimension(:), allocatable [parm::usle\\_cfac](#)

- real \*8, dimension(:), allocatable **parm::usle\_eifac**
- real \*8, dimension(:), allocatable **parm::sol\_sumfc**  
*amount of water held in soil profile at field capacity (mm H2O)*
- real \*8, dimension(:), allocatable **parm::t\_ov**  
*time for flow from farthest point in subbasin to enter a channel (hour)*
- real \*8, dimension(:), allocatable **parm::anano3**  
*total amount of NO3 applied during the year in auto-fertilization (kg N/ha)*
- real \*8, dimension(:), allocatable **parm::aird**  
*amount of water applied to HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable **parm::wet\_orgp**  
*amount of organic P originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable **parm::sol\_avpor**  
*average porosity for entire soil profile (none)*
- real \*8, dimension(:), allocatable **parm::usle\_mult**  
*product of USLE K, P, LS, exp(rock) (none)*
- real \*8, dimension(:), allocatable **parm::rhd**  
*relative humidity for the day in HRU (none)*
- real \*8, dimension(:), allocatable **parm::u10**  
*wind speed (measured at 10 meters above surface) for the day in HRU (m/s)*
- real \*8, dimension(:), allocatable **parm::cht**  
*canopy height (m)*
- real \*8, dimension(:), allocatable **parm::aairr**  
*average annual amount of irrigation water applied to HRU (mm H2O)*
- real \*8, dimension(:), allocatable **parm::lai\_aamx**  
*maximum leaf area index for the entire period of simulation in the HRU (none)*
- real \*8, dimension(:), allocatable **parm::deepirr**  
*amount of water removed from deep aquifer for irrigation (mm H2O)*
- real \*8, dimension(:), allocatable **parm::shallirr**  
*amount of water removed from shallow aquifer for irrigation (mm H2O)*
- real \*8, dimension(:), allocatable **parm::ch\_l1**  
*longest tributary channel length in subbasin (km)*
- real \*8, dimension(:), allocatable **parm::wet\_no3**  
*amount of nitrate originating from surface runoff in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable **parm::ovrlnd**  
*overland flow onto HRU from upstream routing unit (mm H2O)*
- real \*8, dimension(:), allocatable **parm::canstor**  
*amount of water held in canopy storage (mm H2O)*
- real \*8, dimension(:), allocatable **parm::irr\_mx**  
*maximum irrigation amount per auto application (mm)*
- real \*8, dimension(:), allocatable **parm::auto\_wstr**  
*water stress factor which triggers auto irrigation (none or mm)*
- real \*8, dimension(:), allocatable **parm::cfrt\_id**  
*fertilizer/manure identification number from database (fert.dat) (none)*
- real \*8, dimension(:), allocatable **parm::cfrt\_kg**  
*amount of fertilizer/manure applied to HRU on a given day ((kg/ha)/day)*
- real \*8, dimension(:), allocatable **parm::cpst\_id**
- real \*8, dimension(:), allocatable **parm::cpst\_kg**
- real \*8, dimension(:), allocatable **parm::irr\_asq**  
*surface runoff ratio*
- real \*8, dimension(:), allocatable **parm::irr\_eff**
- real \*8, dimension(:), allocatable **parm::irrsq**

- surface runoff ratio (0-1) .1 is 10% surface runoff (frac)*
- real \*8, dimension(:), allocatable [parm::irrsalt](#)
  - concentration of salt in irrigation water (mg/kg)*
- real \*8, dimension(:), allocatable [parm::irrefm](#)
- real \*8, dimension(:), allocatable [parm::bio\\_eat](#)
  - dry weight of biomass removed by grazing daily ((kg/ha)/day)*
- real \*8, dimension(:), allocatable [parm::bio\\_trmp](#)
  - dry weight of biomass removed by trampling daily ((kg/ha)/day)*
- integer, dimension(:), allocatable [parm::ipst\\_freq](#)
  - number of days between applications (days)*
- integer, dimension(:), allocatable [parm::ifrt\\_freq](#)
  - number of days between applications in continuous fertilizer operation (days)*
- integer, dimension(:), allocatable [parm::irr\\_noa](#)
- integer, dimension(:), allocatable [parm::irr\\_sc](#)
- integer, dimension(:), allocatable [parm::irr\\_no](#)
- integer, dimension(:), allocatable [parm::imp\\_trig](#)
  - release/impound action code (none):*
  - 0 begin impounding water*
  - 1 release impounded water*
- integer, dimension(:), allocatable [parm::fert\\_days](#)
  - number of days continuous fertilization will be simulated (none)*
- integer, dimension(:), allocatable [parm::irr\\_sca](#)
- integer, dimension(:), allocatable [parm::idplt](#)
  - land cover/crop identification code for first crop grown in HRU (the only crop if there is no rotation) (from crop.dat)*
  - (none)*
- integer, dimension(:), allocatable [parm::wstrs\\_id](#)
  - water stress identifier (none):*
  - 1 plant water demand*
  - 2 soil water deficit*
- integer, dimension(:), allocatable [parm::pest\\_days](#)
- real \*8, dimension(:,:), allocatable [parm::bio\\_aahv](#)
- real \*8, dimension(:), allocatable [parm::cumei](#)
- real \*8, dimension(:), allocatable [parm::cumeira](#)
- real \*8, dimension(:), allocatable [parm::cumrt](#)
- real \*8, dimension(:), allocatable [parm::cumrai](#)
- real \*8, dimension(:), allocatable [parm::wet\\_solp](#)
  - amount of soluble P originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::wet\\_chla](#)
  - amount of chlorophyll-a in wetland at end of day (kg chla)*
- real \*8, dimension(:), allocatable [parm::wet\\_no3s](#)
  - amount of nitrate originating from lateral flow in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::pstsol](#)
  - amount of soluble pesticide leached from bottom of soil profile on current day (kg pst/ha)*
- real \*8, dimension(:), allocatable [parm::pnd\\_no3g](#)
  - amount of nitrate originating from groundwater in pond at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::wet\\_seci](#)
  - secchi-disk depth in wetland at end of day (m)*
- real \*8, dimension(:), allocatable [parm::delay](#)
  - groundwater delay: time required for water leaving the bottom of the root zone to reach the shallow aquifer (days)*
- real \*8, dimension(:), allocatable [parm::gwht](#)
  - groundwater height (m)*
- real \*8, dimension(:), allocatable [parm::gw\\_q](#)

- groundwater contribution to streamflow from HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable [parm::pnd\\_solpg](#)  
*amount of soluble P originating from groundwater in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::alpha\\_bf](#)  
*alpha factor for groundwater recession curve (1/days)*
- real \*8, dimension(:), allocatable [parm::alpha\\_bfe](#)  
 $\exp(-\alpha_{bf})$  (none)
- real \*8, dimension(:), allocatable [parm::gw\\_spyld](#)  
*specific yield for shallow aquifer ( $m^3/m^3$ )*
- real \*8, dimension(:), allocatable [parm::alpha\\_bf\\_d](#)  
*alpha factor for groundwater recession curve of the deep aquifer (1/days)*
- real \*8, dimension(:), allocatable [parm::alpha\\_bfe\\_d](#)  
 $\exp(-\alpha_{bf_d})$  for deep aquifer (none)
- real \*8, dimension(:), allocatable [parm::gw\\_qdeep](#)  
*groundwater contribution to streamflow from deep aquifer from HRU on current day (mm H2O)*
- real \*8, dimension(:), allocatable [parm::gw\\_delaye](#)  
 $\exp(-1/delay)$  where *delay(:)* is the groundwater delay (time required for water leaving the bottom of the root zone to reach the shallow aquifer; units-days) (none)
- real \*8, dimension(:), allocatable [parm::gw\\_revap](#)  
*revap coeff: this variable controls the amount of water moving from the shallow aquifer to the root zone as a result of soil moisture depletion (none)*
- real \*8, dimension(:), allocatable [parm::rchrg\\_dp](#)  
*recharge to deep aquifer: the fraction of root zone percolation that reaches the deep aquifer (none)*
- real \*8, dimension(:), allocatable [parm::anion\\_excl](#)  
*fraction of porosity from which anions are excluded*
- real \*8, dimension(:), allocatable [parm::revapmn](#)  
*threshold depth of water in shallow aquifer required to allow revap to occur (mm H2O)*
- real \*8, dimension(:), allocatable [parm::rchrg](#)  
*amount of water recharging both aquifers on current day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::bio\\_min](#)  
*minimum plant biomass for grazing (kg/ha)*
- real \*8, dimension(:), allocatable [parm::ffc](#)  
*initial HRU soil water content expressed as fraction of field capacity (none)*
- real \*8, dimension(:), allocatable [parm::surqsolp](#)  
*amount of soluble phosphorus in surface runoff in HRU for the day (kg P/ha)*
- real \*8, dimension(:), allocatable [parm::deepst](#)  
*depth of water in deep aquifer (mm H2O)*
- real \*8, dimension(:), allocatable [parm::shallst](#)  
*depth of water in shallow aquifer in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::wet\\_solpg](#)  
*amount of soluble P originating from groundwater in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::cklsp](#)
- real \*8, dimension(:), allocatable [parm::rchrg\\_src](#)
- real \*8, dimension(:), allocatable [parm::trapeff](#)  
*filter strip trapping efficiency (used for everything but bacteria) (none)*
- real \*8, dimension(:), allocatable [parm::sol\\_avbd](#)  
*average bulk density for soil profile ( $Mg/m^3$ )*
- real \*8, dimension(:), allocatable [parm::wet\\_no3g](#)  
*amount of nitrate originating from groundwater in wetland at end of day (kg N)*
- real \*8, dimension(:), allocatable [parm::tdrain](#)  
*time to drain soil to field capacity yield used in autofertilization (hours)*



- real \*8, dimension(:), allocatable [parm::gwqmn](#)  
*threshold depth of water in shallow aquifer required before groundwater flow will occur (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::snotmp](#)  
*temperature of snow pack in HRU (deg C)*
- real \*8, dimension(:), allocatable [parm::pplnt](#)  
*plant uptake of phosphorus in HRU for the day (kg P/ha)*
- real \*8, dimension(:), allocatable [parm::gdrain](#)  
*drain tile lag time: the amount of time between the transfer of water from the soil to the drain tile and the release of the water from the drain tile to the reach (hours)*
- real \*8, dimension(:), allocatable [parm::ddrain](#)  
*depth of drain tube from the soil surface (mm)*
- real \*8, dimension(:), allocatable [parm::sol\\_crk](#)  
*crack volume potential of soil (none)*
- real \*8, dimension(:), allocatable [parm::brt](#)  
*fraction of surface runoff within the subbasin which takes 1 day or less to reach the subbasin outlet (none)*
- real \*8, dimension(:), allocatable [parm::dayl](#)  
*length of the current day (hours)*
- real \*8, dimension(:), allocatable [parm::sstmaxd](#)  
*static maximum depressional storage; read from .sdr (mm)*
- real \*8, dimension(:), allocatable [parm::re](#)  
*effective radius of drains (mm)*
- real \*8, dimension(:), allocatable [parm::sdrain](#)  
*distance between two drain tubes or tiles (mm)*
- real \*8, dimension(:), allocatable [parm::ddrain\\_hru](#)
- real \*8, dimension(:), allocatable [parm::drain\\_co](#)  
*drainage coefficient (mm/day)*
- real \*8, dimension(:), allocatable [parm::latksatf](#)  
*multiplication factor to determine  $conk(j1,j)$  from  $sol\_k(j1,j)$  for HRU (none)*
- real \*8, dimension(:), allocatable [parm::pc](#)  
*pump capacity (default pump capacity = 1.042mm/hr or 25mm/day) (mm/hr)*
- real \*8, dimension(:), allocatable [parm::stmaxd](#)  
*maximum surface depressional storage for day in a given HRU (mm)*
- real \*8, dimension(:), allocatable [parm::rnd3](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [parm::rnd2](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [parm::twash](#)
- real \*8, dimension(:), allocatable [parm::sol\\_cns](#)
- real \*8, dimension(:), allocatable [parm::doxq](#)
- real \*8, dimension(:), allocatable [parm::rnd8](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [parm::rnd9](#)  
*random number between 0.0 and 1.0 (none)*
- real \*8, dimension(:), allocatable [parm::percn](#)  
*amount of nitrate percolating past bottom of soil profile (kg N/ha)*
- real \*8, dimension(:), allocatable [parm::sol\\_sumwp](#)
- real \*8, dimension(:), allocatable [parm::qdr](#)  
*total amount of water entering main channel for day from HRU (mm H<sub>2</sub>O)*
- real \*8, dimension(:), allocatable [parm::tauton](#)  
*amount of N applied in autofert operation in year (kg N/ha)*
- real \*8, dimension(:), allocatable [parm::tautop](#)



- amount of P applied in autofert operation in year (kg N/ha)*
  - real \*8, dimension(:), allocatable **parm::cbodu**
  - real \*8, dimension(:), allocatable **parm::chl\_a**
  - real \*8, dimension(:), allocatable **parm::tfertn**
  - real \*8, dimension(:), allocatable **parm::tfertp**
  - real \*8, dimension(:), allocatable **parm::tgrazn**
  - real \*8, dimension(:), allocatable **parm::tgrazp**
  - real \*8, dimension(:), allocatable **parm::latq**
- total lateral flow in soil profile for the day in HRU (mm H2O)*
  - real \*8, dimension(:), allocatable **parm::nplnt**
- plant uptake of nitrogen in HRU for the day (kg N/ha)*
  - real \*8, dimension(:), allocatable **parm::latno3**
- amount of nitrate transported with lateral flow (kg N/ha)*
  - real \*8, dimension(:), allocatable **parm::minpgw**
  - real \*8, dimension(:), allocatable **parm::no3gw**
  - real \*8, dimension(:), allocatable **parm::tileq**
  - real \*8, dimension(:), allocatable **parm::tileno3**
  - real \*8, dimension(:), allocatable **parm::sedorgn**
- amount of organic nitrogen in surface runoff in HRU for the day (kg N/ha)*
  - real \*8, dimension(:), allocatable **parm::sedminpa**
- amount of active mineral phosphorus sorbed to sediment in surface runoff in HRU for day (kg P/ha)*
  - real \*8, dimension(:), allocatable **parm::sedminps**
- amount of stable mineral phosphorus sorbed to sediment in surface runoff in HRU for day (kg P/ha)*
  - real \*8, dimension(:), allocatable **parm::sedyld**
- soil loss caused by water erosion for day in HRU (metric tons)*
  - real \*8, dimension(:), allocatable **parm::sepbtm**
- percolation from bottom of soil profile for the day in HRU (mm H2O)*
  - real \*8, dimension(:), allocatable **parm::strsn**
- fraction of potential plant growth achieved on the day where the reduction is caused by nitrogen stress (none)*
  - real \*8, dimension(:), allocatable **parm::sedorgp**
- amount of organic phosphorus in surface runoff in HRU for the day (kg P/ha)*
  - real \*8, dimension(:), allocatable **parm::surfq**
- surface runoff generated in HRU on the current day (mm H2O)*
  - real \*8, dimension(:), allocatable **parm::strstmp**
- fraction of potential plant growth achieved on the day in HRU where the reduction is caused by temperature stress (none)*
  - real \*8, dimension(:), allocatable **parm::strsp**
- fraction of potential plant growth achieved on the day where the reduction is caused by phosphorus stress (none)*
  - real \*8, dimension(:), allocatable **parm::surqno3**
- amount of nitrate transported with surface runoff (kg N/ha)*
  - real \*8, dimension(:), allocatable **parm::hru\_ha**
- area of HRU in hectares (ha)*
  - real \*8, dimension(:), allocatable **parm::hru\_dafr**
- fraction of total watershed area contained in HRU (km2/km2)*
  - real \*8, dimension(:), allocatable **parm::tcfrtn**
  - real \*8, dimension(:), allocatable **parm::tcfrtp**
  - real \*8, dimension(:), allocatable **parm::drydep\_no3**
- atmospheric dry deposition of nitrates (kg/ha/yr)*
  - real \*8, dimension(:), allocatable **parm::drydep\_nh4**
- atmospheric dry deposition of ammonia (kg/ha/yr)*
  - real \*8, dimension(:), allocatable **parm::bio\_yrms**
- annual biomass (dry weight) in the HRU (metric tons/ha)*

- real \*8, dimension(:), allocatable [parm::phubase](#)  
*base zero total heat units (used when no land cover is growing) (heat units)*
- real \*8, dimension(:), allocatable [parm::hvstiadj](#)  
*optimal harvest index adjusted for water stress for current time during growing season ((kg/ha)/(kg/ha))*
- real \*8, dimension(:), allocatable [parm::laiday](#)  
*leaf area index for HRU ( $m^2/m^2$ )*
- real \*8, dimension(:), allocatable [parm::chlap](#)  
*chlorophyll-a production coefficient for pond (none)*
- real \*8, dimension(:), allocatable [parm::pnd\\_psed](#)  
*amount of mineral P attached to sediment originating from surface runoff in pond at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::laimxfr](#)
- real \*8, dimension(:), allocatable [parm::seccip](#)  
*water clarity coefficient for pond (none)*
- real \*8, dimension(:), allocatable [parm::plantn](#)  
*amount of nitrogen in plant biomass (kg N/ha)*
- real \*8, dimension(:), allocatable [parm::plt\\_et](#)  
*actual ET simulated during life of plant (mm H2O)*
- real \*8, dimension(:), allocatable [parm::wet\\_psed](#)  
*amount of mineral P attached to sediment originating from surface runoff in wetland at end of day (kg P)*
- real \*8, dimension(:), allocatable [parm::bio\\_aams](#)  
*average annual biomass (dry weight) in the HRU (metric tons)*
- real \*8, dimension(:), allocatable [parm::plantp](#)  
*amount of phosphorus stored in plant biomass (kg P/ha)*
- real \*8, dimension(:), allocatable [parm::plt\\_pet](#)  
*potential ET simulated during life of plant (mm H2O)*
- real \*8, dimension(:), allocatable [parm::dormhr](#)  
*time threshold used to define dormant period for plant (when daylength is within the time specified by dl from the minimum daylength for the area, the plant will go dormant) (hour)*
- real \*8, dimension(:), allocatable [parm::lai\\_yrmx](#)  
*maximum leaf area index for the year in the HRU (none)*
- real \*8, dimension(:), allocatable [parm::bio\\_aamx](#)
- real \*8, dimension(:), allocatable [parm::lat\\_pst](#)  
*amount of pesticide in lateral flow in HRU for the day (kg pst/ha)*
- real \*8, dimension(:), allocatable [parm::fld\\_fr](#)  
*fraction of HRU area that drains into floodplain ( $km^2/km^2$ )*
- real \*8, dimension(:), allocatable [parm::orig\\_snohru](#)
- real \*8, dimension(:), allocatable [parm::orig\\_potvol](#)
- real \*8, dimension(:), allocatable [parm::pltfr\\_n](#)  
*fraction of plant biomass that is nitrogen (none)*
- real \*8, dimension(:), allocatable [parm::orig\\_alai](#)
- real \*8, dimension(:), allocatable [parm::orig\\_bioms](#)
- real \*8, dimension(:), allocatable [parm::pltfr\\_p](#)  
*fraction of plant biomass that is phosphorus (none)*
- real \*8, dimension(:), allocatable [parm::orig\\_phuacc](#)
- real \*8, dimension(:), allocatable [parm::orig\\_sumix](#)
- real \*8, dimension(:), allocatable [parm::phu\\_plt](#)  
*total number of heat units to bring plant to maturity (heat units)*
- real \*8, dimension(:), allocatable [parm::orig\\_phu](#)
- real \*8, dimension(:), allocatable [parm::orig\\_shallst](#)
- real \*8, dimension(:), allocatable [parm::orig\\_deepst](#)
- real \*8, dimension(:), allocatable [parm::rip\\_fr](#)  
*fraction of HRU area that drains into riparian zone ( $km^2/km^2$ )*

- real \*8, dimension(:), allocatable **parm::orig\_pndvol**
- real \*8, dimension(:), allocatable **parm::orig\_pndsed**
- real \*8, dimension(:), allocatable **parm::orig\_pndno3**
- real \*8, dimension(:), allocatable **parm::orig\_pndsolv**
- real \*8, dimension(:), allocatable **parm::orig\_pndorgn**
- real \*8, dimension(:), allocatable **parm::orig\_pndorgp**
- real \*8, dimension(:), allocatable **parm::orig\_wetvol**
- real \*8, dimension(:), allocatable **parm::orig\_wetsed**
- real \*8, dimension(:), allocatable **parm::orig\_wetno3**
- real \*8, dimension(:), allocatable **parm::orig\_wetsolv**
- real \*8, dimension(:), allocatable **parm::orig\_wetorgn**
- real \*8, dimension(:), allocatable **parm::orig\_wetorgp**
- real \*8, dimension(:), allocatable **parm::orig\_solcov**
- real \*8, dimension(:), allocatable **parm::orig\_solsw**
- real \*8, dimension(:), allocatable **parm::orig\_potno3**
- real \*8, dimension(:), allocatable **parm::orig\_potsed**
- real \*8, dimension(:), allocatable **parm::wtab**

*water table based on 30 day antecedent climate (precip,et) (mm)*

- real \*8, dimension(:), allocatable **parm::wtab\_mn**
- real \*8, dimension(:), allocatable **parm::wtab\_mx**
- real \*8, dimension(:), allocatable **parm::shallst\_n**

*nitrate concentration in shallow aquifer converted to kg/ha (ppm NO3-N)*

- real \*8, dimension(:), allocatable **parm::gw\_nloss**
- real \*8, dimension(:), allocatable **parm::rchrg\_n**
- real \*8, dimension(:), allocatable **parm::det\_san**
- real \*8, dimension(:), allocatable **parm::det\_sil**
- real \*8, dimension(:), allocatable **parm::det\_cla**
- real \*8, dimension(:), allocatable **parm::det\_sag**
- real \*8, dimension(:), allocatable **parm::det\_lag**
- real \*8, dimension(:), allocatable **parm::afrt\_surface**

*fraction of fertilizer which is applied to top 10 mm of soil (the remaining fraction is applied to first soil layer) (none)*

- real \*8, dimension(:), allocatable **parm::tnylda**

*estimated/target nitrogen content of yield used in autofertilization (kg N/kg yield)*

- real \*8 **parm::ftr\_surface**

*fraction of fertilizer which is applied to the top 10 mm of soil (the remaining fraction is applied to the first soil layer) (none)*

- real \*8, dimension(:), allocatable **parm::auto\_nyr**

*maximum NO3-N content allowed to be applied in one year by auto-fertilization (kg NO3-N/ha)*

- real \*8, dimension(:), allocatable **parm::auto\_napp**

*maximum NO3-N content allowed in one fertilizer application (kg NO3-N/ha)*

- real \*8, dimension(:), allocatable **parm::auto\_nstrs**

*nitrogen stress factor which triggers auto fertilization (none)*

- real \*8, dimension(:), allocatable **parm::manure\_kg**

*dry weight of manure deposited on HRU daily ((kg/ha)/day)*

- real \*8, dimension(:,:), allocatable **parm::rcn\_mo**
- real \*8, dimension(:,:), allocatable **parm::rammo\_mo**
- real \*8, dimension(:,:), allocatable **parm::drydep\_no3\_mo**
- real \*8, dimension(:,:), allocatable **parm::drydep\_nh4\_mo**
- real \*8, dimension(:), allocatable **parm::rcn\_d**
- real \*8, dimension(:), allocatable **parm::rammo\_d**
- real \*8, dimension(:), allocatable **parm::drydep\_no3\_d**
- real \*8, dimension(:), allocatable **parm::drydep\_nh4\_d**
- real \*8, dimension(:,:), allocatable **parm::yldn**

- integer, dimension(:,:), allocatable **parm::gwati**
- real \*8, dimension(:,:), allocatable **parm::gwatn**
- real \*8, dimension(:,:), allocatable **parm::gwatl**
- real \*8, dimension(:,:), allocatable **parm::gwatw**
- real \*8, dimension(:,:), allocatable **parm::gwatd**
- real \*8, dimension(:,:), allocatable **parm::gwatveg**
- real \*8, dimension(:,:), allocatable **parm::gwata**
- real \*8, dimension(:,:), allocatable **parm::gwats**
- real \*8, dimension(:,:), allocatable **parm::gwatspcon**
- real \*8, dimension(:,:), allocatable **parm::rfqeo\_30d**
- real \*8, dimension(:,:), allocatable **parm::eo\_30d**
- real \*8, dimension(:), allocatable **parm::psetlp1**  
*phosphorus settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **parm::psetlp2**  
*phosphorus settling rate for 2nd season (m/day)*
- real \*8, dimension(:,:), allocatable **parm::wgnold**  
*previous value of wgncur(:,:) (none)*
- real \*8, dimension(:,:), allocatable **parm::wgncur**  
*parameter to predict the impact of precip on other weather attributes (none)*  
*wgncur(1,:) parameter which predicts impact of precip on daily maximum air temperature*  
*wgncur(2,:) parameter which predicts impact of precip on daily minimum air temperature*  
*wgncur(3,:) parameter which predicts impact of precip on daily solar radiation*
- real \*8, dimension(:), allocatable **parm::wrt1**  
*1st shape parameter for calculation of water retention (none)*
- real \*8, dimension(:), allocatable **parm::wrt2**  
*2nd shape parameter for calculation of water retention (none)*
- real \*8, dimension(:,:), allocatable **parm::pst\_enr**  
*pesticide enrichment ratio (none)*
- real \*8, dimension(:,:), allocatable **parm::pst\_surq**  
*amount of pesticide type lost in surface runoff on current day in HRU (kg/ha)*
- real \*8, dimension(:,:), allocatable **parm::zdb**  
*division term from net pesticide equation (mm)*
- real \*8, dimension(:,:), allocatable **parm::plt\_pst**  
*pesticide on plant foliage (kg/ha)*
- real \*8, dimension(:), allocatable **parm::psetlw1**  
*phosphorus settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **parm::psetlw2**  
*phosphorus settling rate for 2nd season (m/day)*
- real \*8, dimension(:,:), allocatable **parm::pst\_sed**  
*pesticide loading from HRU sorbed onto sediment (kg/ha)*
- real \*8, dimension(:,:), allocatable **parm::wupnd**  
*average daily water removal from the pond for the month ( $10^4 \text{ m}^3/\text{day}$ )*
- real \*8, dimension(:,:), allocatable **parm::phi**  
*phi(1,:) cross-sectional area of flow at bankfull depth ( $\text{m}^2$ ) phi(2,:) (none) phi(3,:) (none) phi(4,:) (none) phi(5,:) (none) phi(6,:) bottom width of main channel (m) phi(7,:) depth of water when reach is at bankfull depth (m) phi(8,:) average velocity when reach is at bankfull depth (m/s) phi(9,:) wave celerity when reach is at bankfull depth (m/s) phi(10,:) storage time constant for reach at bankfull depth (ratio of storage to discharge) (hour) phi(11,:) average velocity when reach is at 0.1 bankfull depth (low flow) (m/s) phi(12,:) wave celerity when reach is at 0.1 bankfull depth (low flow) (m/s) phi(13,:) storage time constant for reach at 0.1 bankfull depth (low flow) (ratio of storage to discharge) (hour)*
- real \*8, dimension(:,:), allocatable **parm::pcpband**  
*precipitation for the day in band in HRU (mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::tavband**

- average temperature for the day in band in HRU (deg C)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi1](#)
- cross-sectional area of flow at bankfull depth ( $m^2$ )*
- real \*8, dimension(:), allocatable [parm::wat\\_phi5](#)
- flow rate when reach is at bankfull depth ( $m^3/s$ )*
- real \*8, dimension(:), allocatable [parm::wat\\_phi6](#)
- bottom width of main channel (m)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi7](#)
- depth of water when reach is at bankfull (m)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi8](#)
- average velocity when reach is at bankfull depth (m/s)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi9](#)
- wave celerity when reach is at bankfull depth (m/s)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi10](#)
- storage time constant for reach at bankfull depth (ratio of storage to discharge) (hour)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi11](#)
- average velocity when reach is at 0.1 bankfull depth (low flow) (m/s)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi12](#)
- wave celerity when reach is at 0.1 bankfull depth (low flow) (m/s)*
- real \*8, dimension(:), allocatable [parm::wat\\_phi13](#)
- storage time constant for reach at 0.1 bankfull depth (low flow) (ratio of storage to discharge) (hour)*
- real \*8, dimension(:, :), allocatable [parm::snoeb](#)
- snow water content in elevation band on current day (mm H2O)*
- real \*8, dimension(:, :), allocatable [parm::wudeep](#)
- average daily water removal from the deep aquifer for the month ( $10^4 m^3/day$ )*
- real \*8, dimension(:, :), allocatable [parm::wushal](#)
- average daily water removal from the shallow aquifer for the month ( $10^4 m^3/day$ )*
- real \*8, dimension(:, :), allocatable [parm::tmnband](#)
- minimum temperature for the day in band in HRU (deg C)*
- real \*8, dimension(:), allocatable **parm::bss1**
- real \*8, dimension(:), allocatable **parm::bss2**
- real \*8, dimension(:), allocatable **parm::bss3**
- real \*8, dimension(:), allocatable **parm::bss4**
- real \*8, dimension(:), allocatable [parm::nsetlw1](#)
- nitrogen settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable [parm::nsetlw2](#)
- nitrogen settling rate for 2nd season (m/day)*
- real \*8, dimension(:, :), allocatable [parm::snotmpeb](#)
- temperature of snow pack in elevation band (deg C)*
- real \*8, dimension(:), allocatable [parm::surf\\_bs1](#)
- amount of surface runoff lagged over one day (mm H2O)*
- real \*8, dimension(:), allocatable **parm::surf\_bs2**
- real \*8, dimension(:), allocatable **parm::surf\_bs3**
- real \*8, dimension(:), allocatable **parm::surf\_bs4**
- real \*8, dimension(:), allocatable **parm::surf\_bs5**
- real \*8, dimension(:), allocatable **parm::surf\_bs6**
- real \*8, dimension(:), allocatable **parm::surf\_bs7**
- real \*8, dimension(:), allocatable **parm::surf\_bs8**
- real \*8, dimension(:), allocatable **parm::surf\_bs9**
- real \*8, dimension(:), allocatable **parm::surf\_bs10**
- real \*8, dimension(:), allocatable **parm::surf\_bs11**
- real \*8, dimension(:), allocatable **parm::surf\_bs12**

- real \*8, dimension(:), allocatable **parm::surf\_bs13**
- real \*8, dimension(:), allocatable **parm::surf\_bs14**
- real \*8, dimension(:), allocatable **parm::surf\_bs15**
- real \*8, dimension(:), allocatable **parm::surf\_bs16**
- real \*8, dimension(:), allocatable **parm::surf\_bs17**
- real \*8, dimension(:), allocatable **parm::nsetlp1**  
*nitrogen settling rate for 1st season (m/day)*
- real \*8, dimension(:), allocatable **parm::nsetlp2**  
*nitrogen settling rate for 2nd season (m/day)*
- real \*8, dimension(:, :), allocatable **parm::tmxband**  
*maximum temperature for the day in band in HRU (deg C)*
- real \*8, dimension(:, :), allocatable **parm::fracd**  
*fraction of solar radiation occuring during hour in day in HRU (none)*
- real \*8, dimension(:, :), allocatable **parm::rainsub**  
*precipitation for the time step during the day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable **parm::rstpbsb**
- real \*8, dimension(:, :), allocatable **parm::orig\_snoeb**
- real \*8, dimension(:, :), allocatable **parm::orig\_pltpst**
- real \*8, dimension(:, :), allocatable **parm::terr\_p**
- real \*8, dimension(:, :), allocatable **parm::terr\_cn**
- real \*8, dimension(:, :), allocatable **parm::terr\_sl**
- real \*8, dimension(:, :), allocatable **parm::drain\_d**
- real \*8, dimension(:, :), allocatable **parm::drain\_t**
- real \*8, dimension(:, :), allocatable **parm::drain\_g**
- real \*8, dimension(:, :), allocatable **parm::drain\_idep**
- real \*8, dimension(:, :), allocatable **parm::cont\_cn**
- real \*8, dimension(:, :), allocatable **parm::cont\_p**
- real \*8, dimension(:, :), allocatable **parm::filt\_w**
- real \*8, dimension(:, :), allocatable **parm::strip\_n**
- real \*8, dimension(:, :), allocatable **parm::strip\_cn**
- real \*8, dimension(:, :), allocatable **parm::strip\_c**
- real \*8, dimension(:, :), allocatable **parm::strip\_p**
- real \*8, dimension(:, :), allocatable **parm::fire\_cn**
- real \*8, dimension(:, :), allocatable **parm::cropno\_upd**
- real \*8, dimension(:, :), allocatable **parm::hi\_upd**
- real \*8, dimension(:, :), allocatable **parm::laimx\_upd**
- real \*8, dimension(:, :), allocatable **parm::phug**  
*fraction of plant heat units at which grazing begins (none)*
- real \*8, dimension(:, :), allocatable **parm::pst\_lag**
- integer, dimension(:), allocatable **parm::hrupest**  
*pesticide use flag (none)*  
*0: no pesticides used in HRU*  
*1: pesticides used in HRU*
- integer, dimension(:), allocatable **parm::nrelease**  
*sequence number of impound/release operation within the year (none)*
- integer, dimension(:), allocatable **parm::swtrg**  
*rainfall event flag (none):*  
*0: no rainfall event over midnight*  
*1: rainfall event over midnight*
- integer, dimension(:), allocatable **parm::nrot**  
*number of years of rotation (none)*
- integer, dimension(:), allocatable **parm::nfert**  
*sequence number of fertilizer application within the year (none)*

- integer, dimension(:), allocatable [parm::nro](#)  
*sequence number of year in rotation (none)*
- integer, dimension(:), allocatable [parm::igro](#)  
*land cover status code (none). This code informs the model whether or not a land cover is growing at the beginning of the simulation*  
*0 no land cover currently growing*  
*1 land cover growing*
- integer, dimension(:), allocatable [parm::ipnd1](#)  
*beginning month of nutrient settling season (none)*
- integer, dimension(:), allocatable [parm::ipnd2](#)  
*ending month of nutrient settling season (none)*
- integer, dimension(:), allocatable [parm::nair](#)  
*sequence number of auto-irrigation application within the year (none)*
- integer, dimension(:), allocatable [parm::iflod1](#)  
*beginning month of non-flood season (none)*
- integer, dimension(:), allocatable [parm::iflod2](#)  
*ending month of non-flood season (none)*
- integer, dimension(:), allocatable [parm::ndtarg](#)  
*number of days required to reach target storage from current pond storage (none)*
- integer, dimension(:), allocatable [parm::nirr](#)  
*sequence number of irrigation application within the year (none)*
- integer, dimension(:), allocatable [parm::nstress](#)  
*code for approach used to determine amount of nitrogen to HRU (none):*  
*0 nitrogen target approach*  
*1 annual max approach*
- integer, dimension(:), allocatable [parm::iafrttyp](#)
- integer, dimension(:), allocatable [parm::igrotree](#)
- integer, dimension(:), allocatable [parm::grz\\_days](#)  
*number of days grazing will be simulated (none)*
- integer, dimension(:), allocatable [parm::nmgt](#)  
*management code (for GIS output only) (none)*
- integer, dimension(:), allocatable [parm::nafert](#)  
*sequence number of auto-fert application within the year (none)*
- integer, dimension(:), allocatable [parm::nsweep](#)  
*sequence number of street sweeping operation within the year (none)*
- integer, dimension(:), allocatable [parm::icr](#)  
*sequence number of crop grown within the current year (none)*
- integer, dimension(:), allocatable [parm::ncut](#)  
*sequence number of harvest operation within a year (none)*
- integer, dimension(:), allocatable [parm::irrno](#)  
*irrigation source location (none)*  
*if IRRSC=1, IRRNO is the number of the reach*  
*if IRRSC=2, IRRNO is the number of the reservoir*  
*if IRRSC=3, IRRNO is the number of the subbasin*  
*if IRRSC=4, IRRNO is the number of the subbasin*  
*if IRRSC=5, not used*
- integer, dimension(:), allocatable [parm::sol\\_nly](#)  
*number of soil layers in HRU (none)*
- integer, dimension(:), allocatable [parm::npcp](#)  
*prior day category (none)*  
*1 dry day*  
*2 wet day*
- integer, dimension(:), allocatable [parm::irn](#)



- average annual number of irrigation applications in HRU (none)*

  - integer, dimension(:), allocatable **parm::ncf**
- sequence number of continuous fertilization operation within the year (none)*

  - integer, dimension(:), allocatable **parm::ngr**
- sequence number of grazing operation within the year (none)*

  - integer, dimension(:), allocatable **parm::igrz**
- grazing flag for HRU (none):*  
*0 HRU currently not grazed*  
*1 HRU currently grazed*

  - integer, dimension(:), allocatable **parm::ndeat**
- number of days HRU has been grazed (days)*

  - integer, dimension(:), allocatable **parm::hru\_sub**
- subbasin in which HRU is located (none)*

  - integer, dimension(:), allocatable **parm::urblu**
- urban land type identification number from urban.dat (none)*

  - integer, dimension(:), allocatable **parm::ldrain**
- soil layer where drainage tile is located (none)*

  - integer, dimension(:), allocatable **parm::idorm**
- dormancy status code (none):*  
*0 land cover growing (not dormant)*  
*1 land cover dormant*

  - integer, dimension(:), allocatable **parm::hru\_seq**
- urban simulation code (none):*  
*0 no urban sections in HRU*  
*1 urban sections in HRU, simulate using USGS regression equations*  
*2 urban sections in HRU, simulate using build up/wash off algorithm*

  - integer, dimension(:), allocatable **parm::icfrt**
- continuous fertilizer flag for HRU (none):*  
*0 HRU currently not continuously fertilized*  
*1 HRU currently continuously fertilized*

  - integer, dimension(:), allocatable **parm::iday\_fert**
- number of HRU (in subbasin) that is a floodplain (none)*

  - integer, dimension(:), allocatable **parm::ifld**
- number of HRU (in subbasin) that is a riparian zone (none)*

  - integer, dimension(:), allocatable **parm::irip**
- GIS code printed to output files (output.hru, .rch) (none)*

  - integer, dimension(:), allocatable **parm::hrugis**
- number of days HRU has been continuously fertilized (days)*

  - integer, dimension(:), allocatable **parm::ndcfrt**
- irrigation source code (none):*  
*1 divert water from reach*  
*2 divert water from reservoir*  
*3 divert water from shallow aquifer*  
*4 divert water from deep aquifer*  
*5 divert water from source outside watershed*

  - integer, dimension(:), allocatable **parm::irrsc**
- sequence number of tillage operation within current year (none)*

  - integer, dimension(:), allocatable **parm::ntil**
- high water table code (none):*  
*0 no high water table*  
*1 high water table*

  - integer, dimension(:), allocatable **parm::orig\_igro**
- high water table code (none):*  
*0 no high water table*  
*1 high water table*

  - integer, dimension(:), allocatable **parm::iwatable**



- integer, dimension(:), allocatable **parm::curyr\_mat**
- integer, dimension(:), allocatable **parm::icpst**  
*icpst = 0 do not apply*  
*icpst = 1 application period*
- integer, dimension(:), allocatable **parm::ndcpst**  
*current day within the application period (day)*
- integer, dimension(:), allocatable **parm::ncpest**
- integer, dimension(:), allocatable **parm::iday\_pest**  
*current day between applications (day)*
- integer, dimension(:), allocatable **parm::irr\_flag**
- integer, dimension(:), allocatable **parm::irra\_flag**
- integer, dimension(:,:), allocatable **parm::rndseed**  
*random number generator seeds array. The seeds in the array are used to generate random numbers for the following purposes (none):*  
*(1) wet/dry day probability*  
*(2) solar radiation*  
*(3) precipitation*  
*(4) USLE rainfall erosion index*  
*(5) wind speed*  
*(6) 0.5 hr rainfall fraction*  
*(7) relative humidity*  
*(8) maximum temperature*  
*(9) minimum temperature*  
*(10) generate new random numbers*
- integer, dimension(:,:), allocatable **parm::iterr**
- integer, dimension(:,:), allocatable **parm::iyterr**
- integer, dimension(:,:), allocatable **parm::itdrain**
- integer, dimension(:,:), allocatable **parm::iydrain**
- integer, dimension(:,:), allocatable **parm::ncrops**
- integer, dimension(:), allocatable **parm::manure\_id**  
*manure (fertilizer) identification number from fert.dat (none)*
- integer, dimension(:,:), allocatable **parm::mgt\_sdr**
- integer, dimension(:,:), allocatable **parm::idplot**
- integer, dimension(:,:), allocatable **parm::icont**
- integer, dimension(:,:), allocatable **parm::iycont**
- integer, dimension(:,:), allocatable **parm::ifilt**
- integer, dimension(:,:), allocatable **parm::iyfilt**
- integer, dimension(:,:), allocatable **parm::istrip**
- integer, dimension(:,:), allocatable **parm::iystrip**
- integer, dimension(:,:), allocatable **parm::iopday**
- integer, dimension(:,:), allocatable **parm::iopyr**
- integer, dimension(:,:), allocatable **parm::mgt\_ops**
- real \*8, dimension(:), allocatable **parm::wshd\_pstap**  
*total amount of pesticide type applied in watershed during simulation (kg/ha)*
- real \*8, dimension(:), allocatable **parm::wshd\_pstdg**  
*amount of pesticide lost through degradation in watershed (kg pst/ha)*
- integer, dimension(12) **parm::ndmo**
- integer, dimension(:), allocatable **parm::npno**  
*array of unique pesticides used in watershed (none)*
- integer, dimension(:), allocatable **parm::mcrhru**
- character(len=13), dimension(18) **parm::rfile**  
*rainfall file names (.pcp)*
- character(len=13), dimension(18) **parm::tfile**  
*temperature file names (.tmp)*
- character(len=4), dimension(1000) **parm::urbname**

- name of urban land use*
- character(len=1), dimension(:), allocatable [parm::kirr](#)
- irrigation in HRU*
- character(len=1), dimension(:), allocatable [parm::hydgrp](#)
- character(len=16), dimension(:), allocatable [parm::snam](#)
- soil series name*
- character(len=17), dimension(300) [parm::pname](#)
- name of pesticide/toxin*
- character(len=4), dimension(60) [parm::title](#)
- description lines in file.cio (1st 3 lines)*
- character(len=4), dimension(5000) [parm::cpnm](#)
- four character code to represent crop name*
- character(len=17), dimension(50) [parm::fname](#)
- real \*8, dimension(:, :, :), allocatable [parm::flomon](#)
- average daily water loading for month ( $m^3/day$ )*
- real \*8, dimension(:, :, :), allocatable [parm::solpstmon](#)
- average daily soluble pesticide loading for month (mg pst/day)*
- real \*8, dimension(:, :, :), allocatable [parm::srbpstmon](#)
- average daily sorbed pesticide loading for month (mg pst/day)*
- real \*8, dimension(:, :, :), allocatable [parm::orgnmon](#)
- average daily organic N loading for month (kg N/day)*
- real \*8, dimension(:, :, :), allocatable [parm::orgpmon](#)
- average daily organic P loading for month (kg P/day)*
- real \*8, dimension(:, :, :), allocatable [parm::sedmon](#)
- average daily sediment loading for month (metric tons/day)*
- real \*8, dimension(:, :, :), allocatable [parm::minpmon](#)
- average daily mineral P loading for month (kg P/day)*
- real \*8, dimension(:, :, :), allocatable [parm::nh3mon](#)
- average amount of NH3-N loaded to stream on a given day in the month (kg N/day)*
- real \*8, dimension(:, :, :), allocatable [parm::no3mon](#)
- average daily NO3-N loading for month (kg N/day)*
- real \*8, dimension(:, :, :), allocatable [parm::bactlpmon](#)
- average amount of less persistent bacteria loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:, :, :), allocatable [parm::bactpmon](#)
- average amount of persistent bacteria loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:, :, :), allocatable [parm::no2mon](#)
- average amount of NO2-N loaded to stream on a given day in the month (kg N/day)*
- real \*8, dimension(:, :, :), allocatable [parm::cmtl1mon](#)
- average amount of conservative metal #1 loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:, :, :), allocatable [parm::cmtl2mon](#)
- average amount of conservative metal #2 loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:, :, :), allocatable [parm::cmtl3mon](#)
- average amount of conservative metal #3 loaded to stream on a given day in the month (# bact/day)*
- real \*8, dimension(:, :, :), allocatable [parm::cbodmon](#)
- average daily loading of CBOD in month (kg/day)*
- real \*8, dimension(:, :, :), allocatable [parm::chlamon](#)
- average daily loading of chlorophyll-a in month (kg/day)*
- real \*8, dimension(:, :, :), allocatable [parm::disoxmon](#)
- average daily loading of dissolved O2 in month (kg/day)*
- real \*8, dimension(:, :), allocatable [parm::floyr](#)
- average daily water loading for year ( $m^3/day$ )*

- real \*8, dimension(:,:), allocatable [parm::orgnyr](#)  
*average daily organic N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [parm::orgpyr](#)  
*average daily organic P loading for year (kg P/day)*
- real \*8, dimension(:,:), allocatable [parm::sedyr](#)  
*average daily sediment loading for year (metric tons/day)*
- real \*8, dimension(:,:), allocatable [parm::minpyr](#)  
*average daily mineral P loading for year (kg P/day)*
- real \*8, dimension(:,:), allocatable [parm::nh3yr](#)  
*average daily NH3-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [parm::no2yr](#)  
*average daily NO2-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [parm::no3yr](#)  
*average daily NO3-N loading for year (kg N/day)*
- real \*8, dimension(:,:), allocatable [parm::bactpyr](#)  
*average daily loading of less persistent bacteria for year (# bact/day)*
- real \*8, dimension(:,:), allocatable [parm::bactpyr](#)  
*average daily loading of persistent bacteria for year (# bact/day)*
- real \*8, dimension(:,:), allocatable [parm::cmtl1yr](#)  
*average daily loading of conservative metal #1 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::chlayr](#)  
*average daily loading of chlorophyll-a in year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::cmtl2yr](#)  
*average daily loading of conservative metal #2 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::cmtl3yr](#)  
*average daily loading of conservative metal #3 for year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::cbodyr](#)  
*average daily loading of CBOD in year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::disoxyr](#)  
*average daily loading of dissolved O2 in year (kg/day)*
- real \*8, dimension(:,:), allocatable [parm::solpstyr](#)  
*average daily soluble pesticide loading for year (mg pst/day)*
- real \*8, dimension(:,:), allocatable [parm::srbpstyr](#)  
*average daily sorbed pesticide loading for year (mg pst/day)*
- real \*8, dimension(:,:), allocatable [parm::sol\\_mc](#)
- real \*8, dimension(:,:), allocatable [parm::sol\\_mn](#)
- real \*8, dimension(:,:), allocatable [parm::sol\\_mp](#)
- real \*8, dimension(:), allocatable [parm::flocnst](#)
- real \*8, dimension(:), allocatable [parm::orgncnst](#)  
*average daily organic N loading to reach (kg N/day)*
- real \*8, dimension(:), allocatable [parm::sedcnst](#)  
*average daily sediment loading for reach (metric tons/day)*
- real \*8, dimension(:), allocatable [parm::minpcnst](#)  
*average daily soluble P loading to reach (kg P/day)*
- real \*8, dimension(:), allocatable [parm::no3cnst](#)  
*average daily nitrate loading to reach (kg N/day)*
- real \*8, dimension(:), allocatable [parm::orgpcnst](#)  
*average daily organic P loading to reach (kg P/day)*
- real \*8, dimension(:), allocatable [parm::bactpcnst](#)  
*average daily persistent bacteria loading to reach (# bact/day)*
- real \*8, dimension(:), allocatable [parm::nh3cnst](#)

- average daily ammonia loading to reach (kg N/day)*
- real \*8, dimension(:), allocatable [parm::no2cnst](#)
  - average daily nitrite loading to reach (kg N/day)*
- real \*8, dimension(:), allocatable [parm::bactlpcnst](#)
  - average daily less persistent bacteria loading to reach (# bact/day)*
- real \*8, dimension(:), allocatable [parm::cmtl1cnst](#)
  - average daily conservative metal #1 loading (kg/day)*
- real \*8, dimension(:), allocatable [parm::cmtl2cnst](#)
  - average daily conservative metal #2 loading (kg/day)*
- real \*8, dimension(:), allocatable [parm::chlcnst](#)
  - average daily loading of chlorophyll-a (kg/day)*
- real \*8, dimension(:), allocatable [parm::cmtl3cnst](#)
  - average daily conservative metal #3 loading (kg/day)*
- real \*8, dimension(:), allocatable [parm::disoxcnst](#)
  - average daily loading of dissolved O2 (kg/day)*
- real \*8, dimension(:), allocatable [parm::cbodcnst](#)
  - average daily loading of CBOD to reach (kg/day)*
- real \*8, dimension(:), allocatable [parm::solpstcnst](#)
  - average daily soluble pesticide loading (mg/day)*
- real \*8, dimension(:), allocatable [parm::srbpstcnst](#)
  - average daily sorbed pesticide loading (mg/day)*
- integer [parm::nstep](#)
  - max number of time steps per day or number of lines of rainfall data for each day (none)*
- integer [parm::idt](#)
  - length of time step used to report precipitation data for sub-daily modeling (minutes)*
- real \*8, dimension(:), allocatable [parm::hrtwtr](#)
- real \*8, dimension(:), allocatable [parm::hhstor](#)
- real \*8, dimension(:), allocatable [parm::hdepth](#)
- real \*8, dimension(:), allocatable [parm::hsdti](#)
- real \*8, dimension(:), allocatable [parm::hrchwtr](#)
- real \*8, dimension(:), allocatable [parm::halgae](#)
- real \*8, dimension(:), allocatable [parm::horgn](#)
- real \*8, dimension(:), allocatable [parm::hnh4](#)
- real \*8, dimension(:), allocatable [parm::hno2](#)
- real \*8, dimension(:), allocatable [parm::hno3](#)
- real \*8, dimension(:), allocatable [parm::horgp](#)
- real \*8, dimension(:), allocatable [parm::hsolp](#)
- real \*8, dimension(:), allocatable [parm::hbod](#)
- real \*8, dimension(:), allocatable [parm::hdisox](#)
- real \*8, dimension(:), allocatable [parm::hchla](#)
- real \*8, dimension(:), allocatable [parm::hsedyld](#)
- real \*8, dimension(:), allocatable [parm::hsedst](#)
- real \*8, dimension(:), allocatable [parm::hharea](#)
- real \*8, dimension(:), allocatable [parm::hsolpst](#)
- real \*8, dimension(:), allocatable [parm::hsorpst](#)
- real \*8, dimension(:), allocatable [parm::hhqday](#)
  - surface runoff generated each timestep of day in HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::precipdt](#)
  - precipitation, or effective precipitation reaching soil surface, in time step for HRU (mm H2O)*
- real \*8, dimension(:), allocatable [parm::hhtime](#)
- real \*8, dimension(:), allocatable [parm::hbactp](#)
- real \*8, dimension(:), allocatable [parm::hbactlp](#)

- integer, dimension(10) **parm::ivar\_orig**
- real \*8, dimension(10) **parm::rvar\_orig**
- integer **parm::nsave**  
*number of save commands in .fig file*
- integer **parm::nauto**
- integer **parm::iatmodep**
- real \*8, dimension(:), allocatable **parm::wattemp**
- real \*8, dimension(:), allocatable **parm::lkipst\_mass**
- real \*8, dimension(:), allocatable **parm::lkspst\_mass**
- real \*8, dimension(:), allocatable **parm::vel\_chan**
- real \*8, dimension(:), allocatable **parm::vfscon**  
*fraction of the total runoff from the entire field entering the most concentrated 10% of the VFS (none)*
- real \*8, dimension(:), allocatable **parm::vfsrcatio**  
*field area/VFS area ratio (none)*
- real \*8, dimension(:), allocatable **parm::vfsrc**  
*fraction of flow entering the most concentrated 10% of the VFS which is fully channelized (none)*
- real \*8, dimension(:), allocatable **parm::vfsci**
- real \*8, dimension(:,:), allocatable **parm::filter\_i**
- real \*8, dimension(:,:), allocatable **parm::filter\_ratio**
- real \*8, dimension(:,:), allocatable **parm::filter\_con**
- real \*8, dimension(:,:), allocatable **parm::filter\_ch**
- real \*8, dimension(:,:), allocatable **parm::sol\_n**
- integer **parm::cswat**  
*= 0 Static soil carbon (old mineralization routines)*  
*= 1 C-FARM one carbon pool model*  
*= 2 Century model*
- real \*8, dimension(:,:), allocatable **parm::sol\_bdp**
- real \*8, dimension(:,:), allocatable **parm::tillagef**
- real \*8, dimension(:), allocatable **parm::rtfr**
- real \*8, dimension(:), allocatable **parm::stsol\_rd**  
*storing last soil root depth for use in harvestkilop/kilop (mm)*
- integer **parm::urban\_flag**
- integer **parm::dorm\_flag**
- real \*8 **parm::bf\_flg**
- real \*8 **parm::iabstr**
- real \*8, dimension(:), allocatable **parm::ubnrunoff**
- real \*8, dimension(:), allocatable **parm::ubntss**
- real \*8, dimension(:,:), allocatable **parm::sub\_ubnrunoff**
- real \*8, dimension(:,:), allocatable **parm::sub\_ubntss**
- real \*8, dimension(:,:), allocatable **parm::ovrlnd\_dt**
- real \*8, dimension(:,:), allocatable **parm::hhsurf\_bs1**
- real \*8, dimension(:,:), allocatable **parm::hhsurf\_bs2**
- integer **parm::iuh**  
*unit hydrograph method: 1=triangular UH; 2=gamma function UH;*
- integer **parm::sed\_ch**  
*channel routing for HOURLY; 0=Bagnold; 2=Brownlie; 3=Yang;*
- real \*8 **parm::eros\_expo**  
*an exponent in the overland flow erosion equation ranges 1.5-3.0*
- real \*8 **parm::eros\_spl**  
*coefficient of splash erosion varying 0.9-3.1*
- real \*8 **parm::rill\_mult**  
*Multiplier to USLE\_K for soil susceptible to rill erosion, range 0.5-2.0.*
- real \*8 **parm::sedprev**

- real \*8 **parm::c\_factor**
- real \*8 **parm::ch\_d50**  
*median particle diameter of channel bed (mm)*
- real \*8 **parm::sig\_g**  
*geometric standard deviation of particle sizes for the main channel. Mean air temperature at which precipitation is equally likely to be rain as snow/freezing rain.*
- real \*8 **parm::uhalpha**  
*alpha coefficient for estimating unit hydrograph using a gamma function (\*.bsn)*
- real \*8 **parm::abstinit**
- real \*8 **parm::abstmax**
- real \*8, dimension(:,:), allocatable **parm::hhsedy**  
*sediment yield from HRU drung a time step applied to HRU (tons)*
- real \*8, dimension(:,:), allocatable **parm::sub\_subp\_dt**
- real \*8, dimension(:,:), allocatable **parm::sub\_hhsedy**
- real \*8, dimension(:,:), allocatable **parm::sub\_atmp**
- real \*8, dimension(:), allocatable **parm::rhy**
- real \*8, dimension(:), allocatable **parm::init\_abstrc**
- real \*8, dimension(:), allocatable **parm::dratio**
- real \*8, dimension(:), allocatable **parm::hrtevp**
- real \*8, dimension(:), allocatable **parm::hrttlc**
- real \*8, dimension(:,:), allocatable **parm::rchhr**
- real \*8, dimension(:), allocatable **parm::hhresflwi**
- real \*8, dimension(:), allocatable **parm::hhresflwo**
- real \*8, dimension(:), allocatable **parm::hhressedi**
- real \*8, dimension(:), allocatable **parm::hhressedo**
- character(len=4), dimension(:), allocatable **parm::lu\_nodrain**
- integer, dimension(:), allocatable **parm::bmpdrain**
- real \*8, dimension(:), allocatable **parm::sub\_cn2**
- real \*8, dimension(:), allocatable **parm::sub\_ha\_urb**
- real \*8, dimension(:), allocatable **parm::bmp\_recharge**
- real \*8, dimension(:), allocatable **parm::sub\_ha\_imp**
- real \*8, dimension(:), allocatable **parm::subdr\_km**
- real \*8, dimension(:), allocatable **parm::subdr\_ickm**
- real \*8, dimension(:,:), allocatable **parm::sf\_im**
- real \*8, dimension(:,:), allocatable **parm::sf\_iy**
- real \*8, dimension(:,:), allocatable **parm::sp\_sa**
- real \*8, dimension(:,:), allocatable **parm::sp\_pvol**
- real \*8, dimension(:,:), allocatable **parm::sp\_pd**
- real \*8, dimension(:,:), allocatable **parm::sp\_sedi**
- real \*8, dimension(:,:), allocatable **parm::sp\_sede**
- real \*8, dimension(:,:), allocatable **parm::ft\_sa**
- real \*8, dimension(:,:), allocatable **parm::ft\_fsa**
- real \*8, dimension(:,:), allocatable **parm::ft\_dep**
- real \*8, dimension(:,:), allocatable **parm::ft\_h**
- real \*8, dimension(:,:), allocatable **parm::ft\_pd**
- real \*8, dimension(:,:), allocatable **parm::ft\_k**
- real \*8, dimension(:,:), allocatable **parm::ft\_dp**
- real \*8, dimension(:,:), allocatable **parm::ft\_dc**
- real \*8, dimension(:,:), allocatable **parm::ft\_por**
- real \*8, dimension(:,:), allocatable **parm::tss\_den**
- real \*8, dimension(:,:), allocatable **parm::ft\_alp**
- real \*8, dimension(:,:), allocatable **parm::sf\_fr**
- real \*8, dimension(:,:), allocatable **parm::sp\_qi**
- real \*8, dimension(:,:), allocatable **parm::sp\_k**

- real \*8, dimension(:,:), allocatable **parm::ft\_qpnd**
- real \*8, dimension(:,:), allocatable **parm::sp\_dp**
- real \*8, dimension(:,:), allocatable **parm::ft\_qsw**
- real \*8, dimension(:,:), allocatable **parm::ft\_qin**
- real \*8, dimension(:,:), allocatable **parm::ft\_qout**
- real \*8, dimension(:,:), allocatable **parm::ft\_sedpnd**
- real \*8, dimension(:,:), allocatable **parm::sp\_bpw**
- real \*8, dimension(:,:), allocatable **parm::ft\_bpw**
- real \*8, dimension(:,:), allocatable **parm::ft\_sed\_cumul**
- real \*8, dimension(:,:), allocatable **parm::sp\_sed\_cumul**
- integer, dimension(:), allocatable **parm::num\_sf**
- integer, dimension(:,:), allocatable **parm::sf\_typ**
- integer, dimension(:,:), allocatable **parm::sf\_dim**
- integer, dimension(:,:), allocatable **parm::ft\_qfg**
- integer, dimension(:,:), allocatable **parm::sp\_qfg**
- integer, dimension(:,:), allocatable **parm::sf\_ptp**
- integer, dimension(:,:), allocatable **parm::ft\_fc**
- real \*8 **parm::sfsedmean**
- real \*8 **parm::sfsedstdev**
- integer, dimension(:), allocatable **parm::dtp\_imo**  
*month the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **parm::dtp\_iyr**  
*year of the simulation that the reservoir becomes operational (none)*
- integer, dimension(:), allocatable **parm::dtp\_numstage**  
*total number of stages in the weir (none)*
- integer, dimension(:), allocatable **parm::dtp\_numweir**  
*total number of weirs in the BMP (none)*
- integer, dimension(:), allocatable **parm::dtp\_onoff**  
*sub-basin detention pond is associated with (none)*
- integer, dimension(:), allocatable **parm::dtp\_reltype**  
*equations for stage-discharge relationship (none):*  
*1=exponential function,*  
*2=linear,*  
*3=logarithmic,*  
*4=cubic,*  
*5=power*
- integer, dimension(:), allocatable **parm::dtp\_stagdis**  
*(none):*  
*0=use weir/orifice discharge equation to calculate outflow,*  
*1=use stage-dicharge relationship*
- integer, dimension(:), allocatable **parm::dtp\_subnum**
- real \*8, dimension(:), allocatable **parm::cf**  
*this parameter controls the response of decomposition to the combined effect of soil temperature and moisture.*
- real \*8, dimension(:), allocatable **parm::cfh**  
*maximum humification rate*
- real \*8, dimension(:), allocatable **parm::cfdec**  
*the undisturbed soil turnover rate under optimum soil water and temperature. Increasing it will increase carbon and organic N decomp.*
- real \*8, dimension(:), allocatable **parm::lat\_orgn**
- real \*8, dimension(:), allocatable **parm::lat\_orgp**
- integer, dimension(:,:), allocatable **parm::dtp\_weirdim**  
*weir dimensions (none),*  
*1=read user input,*  
*0=use model calculation*



- integer, dimension(:,:), allocatable [parm::dtp\\_weirtype](#)  
*type of weir (none):*  
*1=rectangular and*  
*2=circular*
- real \*8, dimension(:), allocatable [parm::dtp\\_coef1](#)  
*coefficient of 3rd degree in the polynomial equation (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_coef2](#)  
*coefficient of 2nd degree in the polynomial equation (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_coef3](#)  
*coefficient of 1st degree in the polynomial equation (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_evrsv](#)  
*detention pond evaporation coefficient (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_expont](#)  
*exponent used in the exponential equation (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_intcept](#)  
*intercept used in regression equations (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_lwratio](#)  
*ratio of length to width of water back up (none)*
- real \*8, dimension(:), allocatable [parm::dtp\\_totwrwid](#)  
*total constructed width of the detention wall across the creek (m)*
- real \*8, dimension(:), allocatable **parm::dtp\_inflvol**
- real \*8, dimension(:), allocatable **parm::dtp\_wdep**
- real \*8, dimension(:), allocatable **parm::dtp\_totdep**
- real \*8, dimension(:), allocatable **parm::dtp\_watdepact**
- real \*8, dimension(:), allocatable **parm::dtp\_outflow**
- real \*8, dimension(:), allocatable **parm::dtp\_totrel**
- real \*8, dimension(:), allocatable **parm::dtp\_backoff**
- real \*8, dimension(:), allocatable **parm::dtp\_seep\_sa**
- real \*8, dimension(:), allocatable **parm::dtp\_evap\_sa**
- real \*8, dimension(:), allocatable **parm::dtp\_pet\_day**
- real \*8, dimension(:), allocatable **parm::dtp\_pcpvol**
- real \*8, dimension(:), allocatable **parm::dtp\_seepvol**
- real \*8, dimension(:), allocatable **parm::dtp\_evapvol**
- real \*8, dimension(:), allocatable **parm::dtp\_flowin**
- real \*8, dimension(:), allocatable **parm::dtp\_backup\_length**
- real \*8, dimension(:), allocatable **parm::dtp\_ivol**
- real \*8, dimension(:), allocatable **parm::dtp\_ised**
- integer, dimension(:,:), allocatable **parm::so\_res\_flag**
- integer, dimension(:,:), allocatable **parm::ro\_bmp\_flag**
- real \*8, dimension(:,:), allocatable **parm::sol\_watp**
- real \*8, dimension(:,:), allocatable **parm::sol\_solp\_pre**
- real \*8, dimension(:,:), allocatable **parm::psp\_store**
- real \*8, dimension(:,:), allocatable **parm::ssp\_store**
- real \*8, dimension(:,:), allocatable **parm::so\_res**
- real \*8, dimension(:,:), allocatable **parm::sol\_cal**
- real \*8, dimension(:,:), allocatable **parm::sol\_ph**
- integer **parm::sol\_p\_model**
- integer, dimension(:,:), allocatable **parm::a\_days**
- integer, dimension(:,:), allocatable **parm::b\_days**
- real \*8, dimension(:), allocatable [parm::min\\_res](#)  
*minimum residue allowed due to implementation of residue managment in the OPS file (kg/ha)*
- real \*8, dimension(:), allocatable **parm::harv\_min**
- real \*8, dimension(:), allocatable **parm::fstap**



- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_flo**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sed**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_bac**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_pp**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sp**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_pn**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sn**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_flos**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_seds**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_bacs**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_pps**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sps**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_pns**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sns**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_flot**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_sedt**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_bact**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_ppt**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_spt**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_pnt**
- real \*8, dimension(:,:), allocatable **parm::ro\_bmp\_snt**
- real \*8, dimension(:), allocatable **parm::bmp\_flo**
- real \*8, dimension(:), allocatable **parm::bmp\_sed**
- real \*8, dimension(:), allocatable **parm::bmp\_bac**
- real \*8, dimension(:), allocatable **parm::bmp\_pp**
- real \*8, dimension(:), allocatable **parm::bmp\_sp**
- real \*8, dimension(:), allocatable **parm::bmp\_pn**
- real \*8, dimension(:), allocatable **parm::bmp\_sn**
- real \*8, dimension(:), allocatable **parm::bmp\_flag**
- real \*8, dimension(:), allocatable **parm::bmp\_flos**
- real \*8, dimension(:), allocatable **parm::bmp\_seds**
- real \*8, dimension(:), allocatable **parm::bmp\_bacs**
- real \*8, dimension(:), allocatable **parm::bmp\_pps**
- real \*8, dimension(:), allocatable **parm::bmp\_sps**
- real \*8, dimension(:), allocatable **parm::bmp\_pns**
- real \*8, dimension(:), allocatable **parm::bmp\_sns**
- real \*8, dimension(:), allocatable **parm::bmp\_flot**
- real \*8, dimension(:), allocatable **parm::bmp\_sedt**
- real \*8, dimension(:), allocatable **parm::bmp\_bact**
- real \*8, dimension(:), allocatable **parm::bmp\_ppt**
- real \*8, dimension(:), allocatable **parm::bmp\_spt**
- real \*8, dimension(:), allocatable **parm::bmp\_pnt**
- real \*8, dimension(:), allocatable **parm::bmp\_snt**
- real \*8, dimension(:,:), allocatable **parm::dtp\_addon**  
*the distance between spillway levels (m)*
- real \*8, dimension(:,:), allocatable **parm::dtp\_cdis**  
*discharge coefficient for weir/orifice flow (none)*
- real \*8, dimension(:,:), allocatable **parm::dtp\_depweir**  
*depth of rectangular weir at different stages (m)*
- real \*8, dimension(:,:), allocatable **parm::dtp\_diaweir**  
*diameter of orifice hole at different stages (m)*
- real \*8, dimension(:,:), allocatable **parm::dtp\_flowrate**  
*maximum discharge from each stage of the weir/hole (m<sup>3</sup>/s)*
- real \*8, dimension(:,:), allocatable **parm::dtp\_pcpret**

- precipitation for different return periods (not used) (mm)*
- real \*8, dimension(:,:), allocatable [parm::dtp\\_retperd](#)
- return period at different stages (years)*
- real \*8, dimension(:,:), allocatable [parm::dtp\\_wdratio](#)
- width depth ratio of rectangular weirs (none)*
- real \*8, dimension(:,:), allocatable [parm::dtp\\_wrwid](#)
- real \*8, dimension(:), allocatable [parm::ri\\_subkm](#)
- real \*8, dimension(:), allocatable [parm::ri\\_totpvol](#)
- real \*8, dimension(:), allocatable [parm::irmmdt](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_sed](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_fr](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_dim](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_im](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_iy](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_sa](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_vol](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_qi](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_k](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_dd](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_evrsv](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_dep](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_ndt](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_pmpvol](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_sed\\_cumul](#)
- real \*8, dimension(:,:), allocatable [parm::hrnopcp](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_qloss](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_pumpv](#)
- real \*8, dimension(:,:), allocatable [parm::ri\\_sedi](#)
- character(len=4), dimension(:,:), allocatable [parm::ri\\_nirr](#)
- integer, dimension(:), allocatable [parm::num\\_ri](#)
- integer, dimension(:), allocatable [parm::ri\\_luflg](#)
- integer, dimension(:), allocatable [parm::num\\_noirr](#)
- integer, dimension(:), allocatable [parm::wtp\\_subnum](#)
- integer, dimension(:), allocatable [parm::wtp\\_onoff](#)
- integer, dimension(:), allocatable [parm::wtp\\_imo](#)
- integer, dimension(:), allocatable [parm::wtp\\_iyr](#)
- integer, dimension(:), allocatable [parm::wtp\\_dim](#)
- integer, dimension(:), allocatable [parm::wtp\\_stagdis](#)
- integer, dimension(:), allocatable [parm::wtp\\_sdtype](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_pvol](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_pdepth](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdslope](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_lenwidth](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_extdepth](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_hydeff](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_evrsv](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdintc](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdexp](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdc1](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdc2](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_sdc3](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_pdia](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_plen](#)
- real \*8, dimension(:), allocatable [parm::wtp\\_pmann](#)

- real \*8, dimension(:), allocatable **parm::wtp\_ploss**
- real \*8, dimension(:), allocatable **parm::wtp\_k**
- real \*8, dimension(:), allocatable **parm::wtp\_dp**
- real \*8, dimension(:), allocatable **parm::wtp\_sedi**
- real \*8, dimension(:), allocatable **parm::wtp\_sede**
- real \*8, dimension(:), allocatable **parm::wtp\_qi**
- real \*8 **parm::lai\_init**  
*initial leaf area index of transplants*
- real \*8 **parm::bio\_init**  
*initial biomass of transplants (kg/ha)*
- real \*8 **parm::cnop**  
*SCS runoff curve number for moisture condition II (none)*
- real \*8 **parm::harveff**  
*harvest efficiency: fraction of harvested yield that is removed from HRU; the remainder becomes residue on the soil surface(none)*
- real \*8 **parm::hi\_ovr**  
*harvest index target specified at harvest ((kg/ha)/(kg/ha))*
- real \*8 **parm::frac\_harvk**
- real \*8 **parm::lid\_vgcl**  
*van Genuchten equation's coefficient, l (none)*
- real \*8 **parm::lid\_vgcm**  
*van Genuchten equation's coefficient, m (none)*
- real \*8 **parm::lid\_qsurf\_total**
- real \*8 **parm::lid\_farea\_sum**
- real \*8, dimension(:,:), allocatable **parm::lid\_cuminf\_last**  
*cumulative amount of water infiltrated into the amended soil layer at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::lid\_cumr\_last**  
*cumulative amount of rainfall at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::lid\_excum\_last**  
*cumulative amount of excess rainfall at the last time step in a day (mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::lid\_f\_last**  
*potential infiltration rate of the amended soil layer at the last time step in a day (mm/mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::lid\_sw\_last**  
*soil water content of the amended soil layer at the last time step in a day (mm/mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::lid\_qsurf**  
*depth of runoff generated on a LID in a given time interval (mm H2O)*
- real \*8, dimension(:,:), allocatable **parm::interval\_last**
- real \*8, dimension(:,:), allocatable **parm::lid\_str\_last**
- real \*8, dimension(:,:), allocatable **parm::lid\_farea**
- real \*8, dimension(:,:), allocatable **parm::lid\_sw\_add**
- real \*8, dimension(:,:), allocatable **parm::lid\_cumqperc\_last**
- real \*8, dimension(:,:), allocatable **parm::lid\_cumirr\_last**
- integer, dimension(:,:), allocatable **parm::gr\_onoff**
- integer, dimension(:,:), allocatable **parm::gr\_imo**
- integer, dimension(:,:), allocatable **parm::gr\_iyr**
- real \*8, dimension(:,:), allocatable **parm::gr\_farea**  
*fractional area of a green roof to the HRU (none)*
- real \*8, dimension(:,:), allocatable **parm::gr\_solop**
- real \*8, dimension(:,:), allocatable **parm::gr\_etcoef**
- real \*8, dimension(:,:), allocatable **parm::gr\_fc**
- real \*8, dimension(:,:), allocatable **parm::gr\_wp**
- real \*8, dimension(:,:), allocatable **parm::gr\_ksat**

- real \*8, dimension(:,:), allocatable **parm::gr\_por**
- real \*8, dimension(:,:), allocatable **parm::gr\_hydeff**
- real \*8, dimension(:,:), allocatable **parm::gr\_soldpt**
- integer, dimension(:,:), allocatable **parm::rg\_onoff**
- integer, dimension(:,:), allocatable **parm::rg\_imo**
- integer, dimension(:,:), allocatable **parm::rg\_iyr**
- real \*8, dimension(:,:), allocatable **parm::rg\_farea**
- real \*8, dimension(:,:), allocatable **parm::rg\_solop**
- real \*8, dimension(:,:), allocatable **parm::rg\_etcoef**
- real \*8, dimension(:,:), allocatable **parm::rg\_fc**
- real \*8, dimension(:,:), allocatable **parm::rg\_wp**
- real \*8, dimension(:,:), allocatable **parm::rg\_ksat**
- real \*8, dimension(:,:), allocatable **parm::rg\_por**
- real \*8, dimension(:,:), allocatable **parm::rg\_hydeff**
- real \*8, dimension(:,:), allocatable **parm::rg\_soldpt**
- real \*8, dimension(:,:), allocatable **parm::rg\_dimop**
- real \*8, dimension(:,:), allocatable **parm::rg\_sarea**
- real \*8, dimension(:,:), allocatable **parm::rg\_vol**
- real \*8, dimension(:,:), allocatable **parm::rg\_sth**
- real \*8, dimension(:,:), allocatable **parm::rg\_sdia**
- real \*8, dimension(:,:), allocatable **parm::rg\_bdia**
- real \*8, dimension(:,:), allocatable **parm::rg\_sts**
- real \*8, dimension(:,:), allocatable **parm::rg\_orifice**
- real \*8, dimension(:,:), allocatable **parm::rg\_oheight**
- real \*8, dimension(:,:), allocatable **parm::rg\_odia**
- integer, dimension(:,:), allocatable **parm::cs\_onoff**
- integer, dimension(:,:), allocatable **parm::cs\_imo**
- integer, dimension(:,:), allocatable **parm::cs\_iyr**
- integer, dimension(:,:), allocatable **parm::cs\_grcon**
- real \*8, dimension(:,:), allocatable **parm::cs\_farea**
- real \*8, dimension(:,:), allocatable **parm::cs\_vol**
- real \*8, dimension(:,:), allocatable **parm::cs\_rdepth**
- integer, dimension(:,:), allocatable **parm::pv\_onoff**
- integer, dimension(:,:), allocatable **parm::pv\_imo**
- integer, dimension(:,:), allocatable **parm::pv\_iyr**
- integer, dimension(:,:), allocatable **parm::pv\_solop**
- real \*8, dimension(:,:), allocatable **parm::pv\_grvdep**
- real \*8, dimension(:,:), allocatable **parm::pv\_grvpor**
- real \*8, dimension(:,:), allocatable **parm::pv\_farea**
- real \*8, dimension(:,:), allocatable **parm::pv\_drcoef**
- real \*8, dimension(:,:), allocatable **parm::pv\_fc**
- real \*8, dimension(:,:), allocatable **parm::pv\_wp**
- real \*8, dimension(:,:), allocatable **parm::pv\_ksat**
- real \*8, dimension(:,:), allocatable **parm::pv\_por**
- real \*8, dimension(:,:), allocatable **parm::pv\_hydeff**
- real \*8, dimension(:,:), allocatable **parm::pv\_soldpt**
- integer, dimension(:,:), allocatable **parm::lid\_onoff**
- real \*8, dimension(:,:), allocatable **parm::sol\_hsc**  
*mass of C present in slow humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_hsn**  
*mass of N present in slow humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_hpc**  
*mass of C present in passive humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_hpn**

- mass of N present in passive humus (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lm**
- mass of metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lmc**
- mass of C in metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lmn**
- mass of N in metabolic litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_ls**
- mass of structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lsc**
- mass of C in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lsl**
- mass of lignin in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_lsn**
- mass of N in structural litter (kg ha-1)*
- real \*8, dimension(:,:), allocatable **parm::sol\_bmc**
- real \*8, dimension(:,:), allocatable **parm::sol\_bmn**
- real \*8, dimension(:,:), allocatable **parm::sol\_rnmn**
- real \*8, dimension(:,:), allocatable **parm::sol\_lslc**
- real \*8, dimension(:,:), allocatable **parm::sol\_lslnc**
- real \*8, dimension(:,:), allocatable **parm::sol\_rspc**
- real \*8, dimension(:,:), allocatable **parm::sol\_woc**
- real \*8, dimension(:,:), allocatable **parm::sol\_won**
- real \*8, dimension(:,:), allocatable **parm::sol\_hp**
- real \*8, dimension(:,:), allocatable **parm::sol\_hs**
- real \*8, dimension(:,:), allocatable **parm::sol\_bm**
- real \*8, dimension(:,:), allocatable **parm::sol\_cac**
- real \*8, dimension(:,:), allocatable **parm::sol\_cec**
- real \*8, dimension(:,:), allocatable **parm::sol\_percc**
- real \*8, dimension(:,:), allocatable **parm::sol\_latc**
- real \*8, dimension(:), allocatable **parm::sedc\_d**
- amount of C lost with sediment pools (kg C/ha)*
- real \*8, dimension(:), allocatable **parm::surfqc\_d**
- real \*8, dimension(:), allocatable **parm::latc\_d**
- real \*8, dimension(:), allocatable **parm::percc\_d**
- real \*8, dimension(:), allocatable **parm::foc\_d**
- real \*8, dimension(:), allocatable **parm::nppc\_d**
- real \*8, dimension(:), allocatable **parm::rsdc\_d**
- real \*8, dimension(:), allocatable **parm::grainc\_d**
- real \*8, dimension(:), allocatable **parm::stoverc\_d**
- real \*8, dimension(:), allocatable **parm::soc\_d**
- real \*8, dimension(:), allocatable **parm::rspc\_d**
- real \*8, dimension(:), allocatable **parm::emitc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_sedc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_surfqc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_latc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_percc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_foc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_nppc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_rsdc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_grainc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_stoverc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_emitc\_d**

- real \*8, dimension(:), allocatable **parm::sub\_soc\_d**
- real \*8, dimension(:), allocatable **parm::sub\_rspc\_d**
- real \*8, dimension(:), allocatable **parm::sedc\_m**
- real \*8, dimension(:), allocatable **parm::surfqc\_m**
- real \*8, dimension(:), allocatable **parm::latc\_m**
- real \*8, dimension(:), allocatable **parm::percc\_m**
- real \*8, dimension(:), allocatable **parm::foc\_m**
- real \*8, dimension(:), allocatable **parm::nppc\_m**
- real \*8, dimension(:), allocatable **parm::rsdc\_m**
- real \*8, dimension(:), allocatable **parm::grainc\_m**
- real \*8, dimension(:), allocatable **parm::stoverc\_m**
- real \*8, dimension(:), allocatable **parm::emitc\_m**
- real \*8, dimension(:), allocatable **parm::soc\_m**
- real \*8, dimension(:), allocatable **parm::rspc\_m**
- real \*8, dimension(:), allocatable **parm::sedc\_a**
- real \*8, dimension(:), allocatable **parm::surfqc\_a**
- real \*8, dimension(:), allocatable **parm::latc\_a**
- real \*8, dimension(:), allocatable **parm::percc\_a**
- real \*8, dimension(:), allocatable **parm::foc\_a**
- real \*8, dimension(:), allocatable **parm::nppc\_a**
- real \*8, dimension(:), allocatable **parm::rsdc\_a**
- real \*8, dimension(:), allocatable **parm::grainc\_a**
- real \*8, dimension(:), allocatable **parm::stoverc\_a**
- real \*8, dimension(:), allocatable **parm::emitc\_a**
- real \*8, dimension(:), allocatable **parm::soc\_a**
- real \*8, dimension(:), allocatable **parm::rspc\_a**
- integer, dimension(:), allocatable **parm::tillage\_switch**
- real \*8, dimension(:), allocatable **parm::tillage\_depth**
- integer, dimension(:), allocatable **parm::tillage\_days**
- real \*8, dimension(:), allocatable **parm::tillage\_factor**
- real \*8 **parm::dthy**  
*time interval for subdaily routing*
- integer, dimension(4) **parm::ihx**
- integer, dimension(:), allocatable **parm::nhy**
- real \*8, dimension(:), allocatable **parm::rchx**
- real \*8, dimension(:), allocatable **parm::rcss**
- real \*8, dimension(:), allocatable **parm::qcap**
- real \*8, dimension(:), allocatable **parm::chxa**
- real \*8, dimension(:), allocatable **parm::chxp**
- real \*8, dimension(:, :, :), allocatable **parm::qhy**
- real \*8 **parm::ff1**
- real \*8 **parm::ff2**

### 7.73.1 Detailed Description

file containing the module parm

Author

modified by Javier Burguete Tolosa

## 7.74 ndenit.f90 File Reference

### Functions/Subroutines

- subroutine [ndenit](#) (k, j, cdg, wdn, void)  
*this subroutine computes denitrification*

#### 7.74.1 Detailed Description

file containing the subroutine ndenit

#### Author

modified by Javier Burguete

#### 7.74.2 Function/Subroutine Documentation

##### 7.74.2.1 ndenit()

```
subroutine ndenit (
    integer, intent(in) k,
    integer, intent(in) j,
    real*8, intent(in) cdg,
    real*8, intent(out) wdn,
    real*8, intent(in) void )
```

this subroutine computes denitrification

#### Parameters

|     |             |  |
|-----|-------------|--|
| in  | <i>k</i>    |  |
| in  | <i>j</i>    |  |
| in  | <i>cdg</i>  |  |
| in  | <i>wdn</i>  |  |
| out | <i>void</i> |  |

## 7.75 newtillmix.f90 File Reference

### Functions/Subroutines

- subroutine [newtillmix](#) (j, bmix)  
*this subroutine mixes residue and nutrients during tillage and biological mixing. Mixing was extended to all layers. A subroutine to simulate stimulation of organic matter decomposition was added. March 2009: testing has been minimal and further adjustments are expected. Use with caution!*

### 7.75.1 Detailed Description

file containing the subroutine newtillmix

#### Author

Armen R. Kemanian,  
Stefan Julich,  
Cole Rossi  
modified by Javier Burguete

### 7.75.2 Function/Subroutine Documentation

#### 7.75.2.1 newtillmix()

```
subroutine newtillmix (
    integer, intent(in) j,
    real*8, intent(in) bmix )
```

this subroutine mixes residue and nutrients during tillage and biological mixing. Mixing was extended to all layers. A subroutine to simulate stimulation of organic matter decomposition was added. March 2009: testing has been minimal and further adjustments are expected. Use with caution!

#### Parameters

|    |             |   |
|----|-------------|---|
| in | <i>j</i>    | HRU number (none)   |
| in | <i>bmix</i> | biological mixing efficiency: this number is zero for tillage operations (none) |

## 7.76 nfix.f90 File Reference

### Functions/Subroutines

- subroutine [nfix](#) (j)  
*this subroutine estimates nitrogen fixation by legumes*

#### 7.76.1 Detailed Description

file containing the subroutine nfix

#### Author

modified by Javier Burguete



## 7.76.2 Function/Subroutine Documentation

### 7.76.2.1 nfix()

```
subroutine nfix (
    integer, intent(in) j )
```

this subroutine estimates nitrogen fixation by legumes

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.77 nitvol.f90 File Reference

### Functions/Subroutines

- subroutine [nitvol](#) (j)  
*this subroutine estimates daily mineralization (NH3 to NO3) and volatilization of NH3*

### 7.77.1 Detailed Description

file containing the subroutine nitvol

#### Author

modified by Javier Burguete

## 7.77.2 Function/Subroutine Documentation

### 7.77.2.1 nitvol()

```
subroutine nitvol (
    integer, intent(in) j )
```

this subroutine estimates daily mineralization (NH3 to NO3) and volatilization of NH3

#### Parameters

|   |            |
|---|------------|
| j | HRU number |
|---|------------|

## 7.78 nlch.f90 File Reference

### Functions/Subroutines

- subroutine [nlch](#) (j)

*this subroutine simulates the loss of nitrate via surface runoff, lateral flow, tile flow, and percolation out of the profile*

#### 7.78.1 Detailed Description

file containing the subroutine nlch

##### Author

modified by Javier Burguete

#### 7.78.2 Function/Subroutine Documentation

##### 7.78.2.1 nlch()

```
subroutine nlch (
    integer, intent(in) j )
```

this subroutine simulates the loss of nitrate via surface runoff, lateral flow, tile flow, and percolation out of the profile

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.79 nminrl.f90 File Reference

### Functions/Subroutines

- subroutine [nminrl](#) (j)

*this subroutine estimates daily nitrogen and phosphorus mineralization and immobilization considering fresh organic material (plant residue) and active and stable humus material*

#### 7.79.1 Detailed Description

file containing the subroutine nminrl

##### Author

modified by Javier Burguete

## 7.79.2 Function/Subroutine Documentation

### 7.79.2.1 nminrl()

```
subroutine nminrl (
    integer, intent(in) j )
```

this subroutine estimates daily nitrogen and phosphorus mineralization and immobilization considering fresh organic material (plant residue) and active and stable humus material

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.80 npup.f90 File Reference

### Functions/Subroutines

- subroutine [npup](#) (j)  
*this subroutine calculates plant phosphorus uptake*

### 7.80.1 Detailed Description

file containing the subroutine npup

#### Author

modified by Javier Burguete

## 7.80.2 Function/Subroutine Documentation

### 7.80.2.1 npup()

```
subroutine npup (
    integer, intent(in) j )
```

this subroutine calculates plant phosphorus uptake

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.81 nrain.f90 File Reference

### Functions/Subroutines

- subroutine `nrain` (j)  
*this subroutine adds nitrate from rainfall to the soil profile*

#### 7.81.1 Detailed Description

file containing the subroutine nrain

##### Author

modified by Javier Burguete

#### 7.81.2 Function/Subroutine Documentation

##### 7.81.2.1 nrain()

```
subroutine nrain (
    integer, intent(in) j )
```

this subroutine adds nitrate from rainfall to the soil profile

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.82 nup.f90 File Reference

### Functions/Subroutines

- subroutine `nup` (j)  
*this subroutine calculates plant nitrogen uptake*

#### 7.82.1 Detailed Description

file containing the subroutine nup

##### Author

modified by Javier Burguete

## 7.82.2 Function/Subroutine Documentation

### 7.82.2.1 nup()

```
subroutine nup (
    integer, intent(in) j )
```

this subroutine calculates plant nitrogen uptake

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.83 nuts.f90 File Reference

### Functions/Subroutines

- subroutine [nuts](#) (u1, u2, uu)

*this function calculates the plant stress factor caused by limited supply of nitrogen or phosphorus*

### 7.83.1 Detailed Description

file containing the subroutine nuts

#### Author

modified by Javier Burguete

## 7.83.2 Function/Subroutine Documentation

### 7.83.2.1 nuts()

```
subroutine nuts (
    real*8, intent(in) u1,
    real*8, intent(in) u2,
    real*8, intent(out) uu )
```

this function calculates the plant stress factor caused by limited supply of nitrogen or phosphorus

## Parameters

|     |           |  |
|-----|-----------|--|
| in  | <i>u1</i> | actual amount of element in plant (kg/ha)  |
| in  | <i>u2</i> | optimal amount of element in plant (kg/ha)   |
| out | <i>uu</i> | fraction of optimal plant growth achieved where reduction is caused by plant element deficiency (none) |

## 7.84 openwth.f90 File Reference

### Functions/Subroutines

- subroutine [openwth](#)  
*this subroutine opens the precipitation, temperature, solar radiation, relative humidity and wind speed files for simulations using measured weather data*

#### 7.84.1 Detailed Description

file containing the subroutine openwth

#### Author

modified by Javier Burguete

## 7.85 operatn.f90 File Reference

### Functions/Subroutines

- subroutine [operatn](#) (*j*)  
*this subroutine performs all management operations*

#### 7.85.1 Detailed Description

file containing the subroutine operatn

#### Author

modified by Javier Burguete

#### 7.85.2 Function/Subroutine Documentation

##### 7.85.2.1 operatn()

```
subroutine operatn (
    integer, intent(in) j )
```

this subroutine performs all management operations

## Parameters

|            |          |            |
|------------|----------|------------|
| <i>i</i> n | <i>j</i> | HRU number |
|------------|----------|------------|

## 7.86 orgn.f90 File Reference

### Functions/Subroutines

- subroutine `orgn` (*iwave*, *j*)  
*this subroutine calculates the amount of organic nitrogen removed in surface runoff*

#### 7.86.1 Detailed Description

file containing the subroutine `orgn`

## Author

modified by Javier Burguete

#### 7.86.2 Function/Subroutine Documentation

##### 7.86.2.1 `orgn()`

```
subroutine orgn (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

this subroutine calculates the amount of organic nitrogen removed in surface runoff

## Parameters

|            |              |   |
|------------|--------------|---|
| <i>i</i> n | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| <i>i</i> n | <i>j</i>     | HRU number  |

## 7.87 orgncswat.f90 File Reference

### Functions/Subroutines

- subroutine `orgncswat` (*iwave*, *j*)  
*this subroutine calculates the amount of organic nitrogen removed in surface runoff - when using CSWAT it excludes sol\_aorgn, uses only sol\_n = sol\_orgn, and includes sol\_mn (nitrogen in manure)*

### 7.87.1 Detailed Description

file containing the subroutine orgncswat

Author

modified by Javier Burguete

### 7.87.2 Function/Subroutine Documentation

#### 7.87.2.1 orgncswat()

```
subroutine orgncswat (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

this subroutine calculates the amount of organic nitrogen removed in surface runoff - when using CSWAT it excludes sol\_aorgn, uses only sol\_n = sol\_orgn, and includes sol\_mn (nitrogen in manure)

Parameters

|    |              |   |
|----|--------------|---|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| in | <i>j</i>     | HRU number  |

## 7.88 orgncswat2.f90 File Reference

### Functions/Subroutines

- subroutine [orgncswat2](#) (iwave, j)  
*this subroutine calculates the amount of organic nitrogen removed in surface runoff - when using CSWAT==2 it*

#### 7.88.1 Detailed Description

file containing the subroutine orgncswat2

Author

modified by Javier Burguete

#### 7.88.2 Function/Subroutine Documentation



### 7.88.2.1 orgncswat2()

```
subroutine orgncswat2 (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

this subroutine calculates the amount of organic nitrogen removed in surface runoff - when using CSWAT==2 it

#### Parameters

|    |              |   |
|----|--------------|---|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| in | <i>j</i>     | HRU number  |

## 7.89 origtile.f90 File Reference

### Functions/Subroutines

- subroutine [origtile](#) (d, j)

*this subroutine computes tile drainage using basic tile equations developed by Saleh et al.(2005)*

### 7.89.1 Detailed Description

file containing the subroutine origtile

#### Author

modified by Javier Burguete

### 7.89.2 Function/Subroutine Documentation

#### 7.89.2.1 origtile()

```
subroutine origtile (
    real*8, intent(in) d,
    integer, intent(in) j )
```

this subroutine computes tile drainage using basic tile equations developed by Saleh et al.(2005)

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>d</i> |            |
| in | <i>j</i> | HRU number |

## 7.90 ovr\_sed.f90 File Reference

### Functions/Subroutines

- subroutine `ovr_sed` ()  
*this subroutine computes splash erosion by raindrop impact and flow erosion by overland flow*

#### 7.90.1 Detailed Description

file containing the subroutine `ovr_sed`

##### Author

modified by Javier Burguete

## 7.91 percmacro.f90 File Reference

### Functions/Subroutines

- subroutine `percmacro` (j)  
*this subroutine computes percolation by crack flow*

#### 7.91.1 Detailed Description

file containing the subroutine `percmacro`

##### Author

modified by Javier Burguete

#### 7.91.2 Function/Subroutine Documentation

##### 7.91.2.1 percmacro()

```
subroutine percmacro (
    integer, intent(in) j )
```

this subroutine computes percolation by crack flow

##### Parameters

|                 |                |            |
|-----------------|----------------|------------|
| <code>in</code> | <code>j</code> | HRU number |
|-----------------|----------------|------------|

## 7.92 percmmain.f90 File Reference

### Functions/Subroutines

- subroutine [percmmain](#) (*j*)  
*this subroutine is the master soil percolation component*

#### 7.92.1 Detailed Description

file containing the subroutine percmmain

##### Author

modified by Javier Burguete

#### 7.92.2 Function/Subroutine Documentation

##### 7.92.2.1 percmmain()

```
subroutine percmmain (
    integer, intent(in) j )
```

this subroutine is the master soil percolation component

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.93 percmicro.f90 File Reference

### Functions/Subroutines

- subroutine [percmicro](#) (*ly1*, *j*)  
*this subroutine computes percolation and lateral subsurface flow from a soil layer when field capacity is exceeded*

#### 7.93.1 Detailed Description

file containing the subroutine percmicro

##### Author

modified by Javier Burguete

## 7.93.2 Function/Subroutine Documentation

### 7.93.2.1 percmicro()

```
subroutine percmicro (
    integer, intent(in) ly1,
    integer, intent(in) j )
```

this subroutine computes percolation and lateral subsurface flow from a soil layer when field capacity is exceeded

#### Parameters

|    |            |                   |
|----|------------|-------------------|
| in | <i>ly1</i> | soil layer number |
| in | <i>j</i>   | HRU number        |

## 7.94 pestlch.f90 File Reference

### Functions/Subroutines

- subroutine [pestlch](#) (*j*)

*this subroutine calculates pesticides leached through each layer, pesticide transported with lateral subsurface flow, and pesticide transported with surface runoff*

### 7.94.1 Detailed Description

file containing the subroutine pestlch

#### Author

modified by Javier Burguete

## 7.94.2 Function/Subroutine Documentation

### 7.94.2.1 pestlch()

```
subroutine pestlch (
    integer, intent(in) j )
```

this subroutine calculates pesticides leached through each layer, pesticide transported with lateral subsurface flow, and pesticide transported with surface runoff

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.95 pesty.f90 File Reference

### Functions/Subroutines

- subroutine [pesty](#) (iwave, j)

#### 7.95.1 Detailed Description

file containing the subroutine pesty

#### Author

modified by Javier Burguete

#### 7.95.2 Function/Subroutine Documentation

##### 7.95.2.1 pesty()

```
subroutine pesty (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

## Parameters

|    |       |   |
|----|-------|---|
| in | iwave | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| in | j     | HRU number  |

## 7.96 pgen.f90 File Reference

### Functions/Subroutines

- subroutine [pgen](#) (j)

*this subroutine generates precipitation data when the user chooses to simulate or when data is missing for particular days in the weather file*

### 7.96.1 Detailed Description

file containing the subroutine pgen

Author

modified by Javier Burguete

### 7.96.2 Function/Subroutine Documentation

#### 7.96.2.1 pgen()

```
subroutine pgen (
    integer, intent(in) j )
```

this subroutine generates precipitation data when the user chooses to simulate or when data is missing for particular days in the weather file

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.97 pgenhr.f90 File Reference

### Functions/Subroutines

- subroutine [pgenhr](#) (jj)  
*this subroutine distributes daily rainfall exponentially within the day @parameter[in] jj HRU number*

#### 7.97.1 Detailed Description

file containing the subroutine pgenhr

Author

modified by Javier Burguete

## 7.98 pkq.f90 File Reference

### Functions/Subroutines

- subroutine [pkq](#) (iwave, j)  
*this subroutine computes the peak runoff rate for each HRU and the entire subbasin using a modification of the rational formula*

### 7.98.1 Detailed Description

file containing the subroutine pkq

#### Author

J. Jeong,  
C. Santhi,  
modified by Javier Burguete

file containing the subroutine pkq

#### Author

modified by Javier Burguete

### 7.98.2 Function/Subroutine Documentation

#### 7.98.2.1 pkq()

```
subroutine pkq (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

this subroutine computes the peak runoff rate for each HRU and the entire subbasin using a modification of the rational formula

#### Parameters

|    |              |  |
|----|--------------|--|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU MUSLE(sedyld) each hru is calculated independently using hru area and adjusted channel length<br>iwave = 1 subbasin # for subbasin MUSLE is computed for entire subbasin using hru weighted KLSCP |
| in | <i>j</i>     | HRU number (none)  |

## 7.99 plantmod.f90 File Reference

### Functions/Subroutines

- subroutine [plantmod](#) (j)

*this subroutine predicts daily potential growth of total plant biomass and roots and calculates leaf area index. Incorporates residue for tillage functions and decays residue on ground surface. Adjusts daily dry matter based on water stress.*

### 7.99.1 Detailed Description

file containing the subroutine plantmod

Author

modified by Javier Burguete

### 7.99.2 Function/Subroutine Documentation

#### 7.99.2.1 plantmod()

```
subroutine plantmod (
    integer, intent(in) j )
```

this subroutine predicts daily potential growth of total plant biomass and roots and calculates leaf area index. Incorporates residue for tillage functions and decays residue on ground surface. Adjusts daily dry matter based on water stress.

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.100 plantop.f90 File Reference

### Functions/Subroutines

- subroutine [plantop](#) (j)  
*this subroutine performs the plant operation*

#### 7.100.1 Detailed Description

file containing the subroutine plantop

Author

modified by Javier Burguete

### 7.100.2 Function/Subroutine Documentation

#### 7.100.2.1 plantop()

```
subroutine plantop (
    integer, intent(in) j )
```

this subroutine performs the plant operation



## Parameters

|                       |          |            |
|-----------------------|----------|------------|
| <i>i</i> <sub>n</sub> | <i>j</i> | HRU number |
|-----------------------|----------|------------|

## 7.101 pmeas.f90 File Reference

### Functions/Subroutines

- subroutine [pmeas](#) (*i*)  
*this subroutine reads in precipitation data and assigns it to the proper subbasins*

#### 7.101.1 Detailed Description

file containing the subroutine pmeas

#### Author

modified by Javier Burguete

#### 7.101.2 Function/Subroutine Documentation

##### 7.101.2.1 pmeas()

```
subroutine pmeas (
    integer, intent(in) i )
```

this subroutine reads in precipitation data and assigns it to the proper subbasins

## Parameters

|                       |          |   |
|-----------------------|----------|---|
| <i>i</i> <sub>n</sub> | <i>i</i> | current day of simulation (julian date) |
|-----------------------|----------|---|

## 7.102 pminrl.f90 File Reference

### Functions/Subroutines

- subroutine [pminrl](#) (*j*)  
*this subroutine computes p flux between the labile, active mineral and stable mineral p pools.*

### 7.102.1 Detailed Description

file containing the subroutine pminrl

Author

modified by Javier Burguete

### 7.102.2 Function/Subroutine Documentation

#### 7.102.2.1 pminrl()

```
subroutine pminrl (
    integer, intent(in) j )
```

this subroutine computes p flux between the labile, active mineral and stable mineral p pools.

Parameters

|          |            |
|----------|------------|
| <i>j</i> | HRU number |
|----------|------------|

## 7.103 pminrl2.f90 File Reference

### Functions/Subroutines

- subroutine [pminrl2](#) (*j*)  
*this subroutine computes p flux between the labile, active mineral and stable mineral p pools. this is the alternate phosphorus model described in [5]*

#### 7.103.1 Detailed Description

file containing the subroutine pminrl2

Author

modified by Javier Burguete

#### 7.103.2 Function/Subroutine Documentation

##### 7.103.2.1 pminrl2()

```
subroutine pminrl2 (
    integer, intent(in) j )
```

this subroutine computes p flux between the labile, active mineral and stable mineral p pools. this is the alternate phosphorus model described in [\[5\]](#)

## Parameters

|          |            |
|----------|------------|
| <i>j</i> | HRU number |
|----------|------------|

## 7.104 psed.f90 File Reference

### Functions/Subroutines

- subroutine `psed` (*iwave*, *j*)

#### 7.104.1 Detailed Description

file containing the subroutine `psed`

## Author

modified by Javier Burguete

#### 7.104.2 Function/Subroutine Documentation

##### 7.104.2.1 `psed()`

```
subroutine psed (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

## Parameters

|    |              |   |
|----|--------------|---|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br><i>iwave</i> = 0 for HRU<br><i>iwave</i> = subbasin # for subbasin |
| in | <i>j</i>     | HRU number  |

## 7.105 qman.f90 File Reference

### Functions/Subroutines

- real \*8 function `qman` (*x1*, *x2*, *x3*, *x4*)

*this subroutine calculates flow rate or flow velocity using Manning's equation. If *x1* is set to 1, the velocity is calculated. If *x1* is set to cross-sectional area of flow, the flow rate is calculated.*

### 7.105.1 Detailed Description

file containing the function qman

Author

modified by Javier Burguete

### 7.105.2 Function/Subroutine Documentation

#### 7.105.2.1 qman()

```
real*8 function qman (
    real*8, intent(in) x1,
    real*8, intent(in) x2,
    real*8, intent(in) x3,
    real*8, intent(in) x4 )
```

this subroutine calculates flow rate or flow velocity using Manning's equation. If x1 is set to 1, the velocity is calculated. If x1 is set to cross-sectional area of flow, the flow rate is calculated.

##### Parameters

|    |    |   |
|----|----|---|
| in | x1 | cross-sectional flow area or 1 (m <sup>2</sup> or none) |
| in | x2 | hydraulic radius (m)                                    |
| in | x3 | Manning's "n" value for channel (none)                  |
| in | x4 | average slope of channel (m/m)                          |

##### Returns

flow rate or flow velocity (m<sup>3</sup>/s or m/s)

## 7.106 rchaa.f90 File Reference

### Functions/Subroutines

- subroutine [rchaa](#) (years)  
*this subroutine writes the average annual reach output to the .rch file*

#### 7.106.1 Detailed Description

file containing the subroutine rchaa

Author

modified by Javier Burguete

## 7.106.2 Function/Subroutine Documentation

### 7.106.2.1 rchaa()

```
subroutine rchaa (  
    real*8, intent(in) years )
```

this subroutine writes the average annual reach output to the .rch file

#### Parameters

|    |       |                              |
|----|-------|------------------------------|
| in | years | length of simulation (years) |
|----|-------|------------------------------|

## 7.107 rchday.f90 File Reference

### Functions/Subroutines

- subroutine [rchday](#)  
*this subroutine writes the daily reach output to the .rch file*

### 7.107.1 Detailed Description

file containing the subroutine rchday

#### Author

modified by Javier Burguete

## 7.108 rchmon.f90 File Reference

### Functions/Subroutines

- subroutine [rchmon](#) (mdays)  
*this subroutine writes the monthly reach output to the .rch file*

### 7.108.1 Detailed Description

file containing the subroutine rchmon

#### Author

modified by Javier Burguete

## 7.108.2 Function/Subroutine Documentation

### 7.108.2.1 rchmon()

```
subroutine rchmon (
    integer, intent(in) mdays )
```

this subroutine writes the monthly reach output to the .rch file

#### Parameters

|    |       |                                   |
|----|-------|-----------------------------------|
| in | mdays | number of days simulated in month |
|----|-------|-----------------------------------|

## 7.109 rchyr.f90 File Reference

### Functions/Subroutines

- subroutine [rchyr](#) (i)  
*this subroutine writes the annual reach output to the .rch file*

### 7.109.1 Detailed Description

file containing the subroutine rchyr

#### Author

modified by Javier Burguete

## 7.109.2 Function/Subroutine Documentation

### 7.109.2.1 rchyr()

```
subroutine rchyr (
    integer, intent(in) i )
```

this subroutine writes the annual reach output to the .rch file

#### Parameters

|    |         |                                 |
|----|---------|---------------------------------|
| in | current | day of simulation (julian date) |
|----|---------|---------------------------------|

## 7.110 readatmodep.f90 File Reference

### Functions/Subroutines

- subroutine [readatmodep](#)  
*this subroutine reads the atmospheric deposition values*

#### 7.110.1 Detailed Description

file containing the subroutine readatmodep

Author

modified by Javier Burguete

## 7.111 readbsn.f90 File Reference

### Functions/Subroutines

- subroutine [readbsn](#)  
*this subroutine reads data from the basin input file (.bsn). This file contains information related to processes modeled or defined at the watershed level*

#### 7.111.1 Detailed Description

file containing the suborutine readbsn

Author

modified by Javier Burguete

## 7.112 readchm.f90 File Reference

### Functions/Subroutines

- subroutine [readchm](#) (l)  
*This subroutine reads data from the HRU/subbasin soil chemical input file (.chm). This file contains initial amounts of pesticides/nutrients in the first soil layer. (Specifics about the first soil layer are given in the .sol file.) All data in the .chm file is optional input.*

#### 7.112.1 Detailed Description

file containing the subroutine readchm

Author

modified by Javier Burguete

## 7.112.2 Function/Subroutine Documentation

### 7.112.2.1 readchm()

```
subroutine readchm (
    integer, intent(in) i )
```

This subroutine reads data from the HRU/subbasin soil chemical input file (.chm). This file contains initial amounts of pesticides/nutrients in the first soil layer. (Specifics about the first soil layer are given in the .sol file.) All data in the .chm file is optional input.

#### Parameters

|    |   |                   |
|----|---|-------------------|
| in | / | HRU number (none) |
|----|---|-------------------|

## 7.113 readcnst.f90 File Reference

### Functions/Subroutines

- subroutine [readcnst](#) (jj)  
*reads in the loading information for the reccnst command*

### 7.113.1 Detailed Description

file containing the subroutine [readcnst.f90](#)

#### Author

modified by Javier Burguete

## 7.113.2 Function/Subroutine Documentation

### 7.113.2.1 readcnst()

```
subroutine readcnst (
    integer, intent(in) jj )
```

reads in the loading information for the reccnst command



## Parameters

|    |    |   |
|----|----|---|
| in | jj | file number associated with recnst command (none) |
|----|----|---|

## 7.114 readfcst.f90 File Reference

### Functions/Subroutines

- subroutine [readfcst](#)  
*this subroutine reads the HRU forecast weather generator parameters from the .cst file*

#### 7.114.1 Detailed Description

file containing the subroutine readfcst

## Author

modified by Javier Burguete

## 7.115 readfert.f90 File Reference

### Functions/Subroutines

- subroutine [readfert](#)  
*this subroutine reads input parameters from the fertilizer/manure (i.e. nutrient) database (fert.dat)*

#### 7.115.1 Detailed Description

file containing the subroutine readfert

## Author

modified by Javier Burguete

## 7.116 readfig.f90 File Reference

### Functions/Subroutines

- subroutine [readfig](#)  
*reads in the routing information from the watershed configuration input file (.fig) and calculates the number of sub-basins, reaches, and reservoirs*

### 7.116.1 Detailed Description

file containing the subroutine readfig

Author

modified by Javier Burguete

## 7.117 readfile.f90 File Reference

### Functions/Subroutines

- subroutine [readfile](#)

*this subroutine opens the main input and output files and reads watershed information from the file.cio*

### 7.117.1 Detailed Description

file containing the subroutine readfile

Author

modified by Javier Burguete

## 7.118 readgw.f90 File Reference

### Functions/Subroutines

- subroutine [readgw](#) (i, j)

*this subroutine reads the parameters from the HRU/subbasin groundwater input file (.gw)*

### 7.118.1 Detailed Description

file containing the subroutine readgw

Author

modified by Javier Burguete

### 7.118.2 Function/Subroutine Documentation

#### 7.118.2.1 readgw()

```
subroutine readgw (  
    integer, intent(in) i,  
    integer, intent(in) j )
```

this subroutine reads the parameters from the HRU/subbasin groundwater input file (.gw)

## Parameters

|    |          |                        |
|----|----------|------------------------|
| in | <i>i</i> | subbasin number (none) |
| in | <i>j</i> | HRU number (none)      |

## 7.119 readhru.f90 File Reference

### Functions/Subroutines

- subroutine [readhru](#) (*i*, *j*)

*this subroutine reads data from the HRU general input file (.hru). This file contains data related to general processes modeled at the HRU level.*

#### 7.119.1 Detailed Description

file containing the subroutine readhru

#### Author

modified by Javier Burguete

#### 7.119.2 Function/Subroutine Documentation

##### 7.119.2.1 readhru()

```
subroutine readhru (
    integer, intent(in) i,
    integer, intent(in) j )
```

this subroutine reads data from the HRU general input file (.hru). This file contains data related to general processes modeled at the HRU level.

## Parameters

|    |          |                        |
|----|----------|------------------------|
| in | <i>i</i> | subbasin number (none) |
| in | <i>j</i> | HRU number (none)      |

## 7.120 readinpt.f90 File Reference

### Functions/Subroutines

- subroutine [readinpt](#)

*this subroutine calls subroutines which read input data for the databases and the HRUs*

### 7.120.1 Detailed Description

file containing the subroutine readinpt

Author

modified by Javier Burguete

## 7.121 readlup.f90 File Reference

### Functions/Subroutines

- subroutine [readlup](#)  
*this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.*

### 7.121.1 Detailed Description

file containing the subroutine readlup

Author

modified by Javier Burguete

## 7.122 readlwq.f90 File Reference

### Functions/Subroutines

- subroutine [readlwq](#) (ii)  
*this subroutine reads data from the lake water quality input file (.lwq). This file contains data related to initial pesticide and nutrient levels in the lake/reservoir and transformation processes occurring within the lake/reservoir. Data in the lake water quality input file is assumed to apply to all reservoirs in the watershed.*

### 7.122.1 Detailed Description

file containing the subroutine readlwq

Author

modified by Javier Burguete

### 7.122.2 Function/Subroutine Documentation

#### 7.122.2.1 readlwq()

```
subroutine readlwq (
    integer, intent(in) ii )
```

this subroutine reads data from the lake water quality input file (.lwq). This file contains data related to initial pesticide and nutrient levels in the lake/reservoir and transformation processes occurring within the lake/reservoir. Data in the lake water quality input file is assumed to apply to all reservoirs in the watershed.

## Parameters

|    |    |                         |
|----|----|-------------------------|
| in | ii | reservoir number (none) |
|----|----|-------------------------|

## 7.123 readmgt.f90 File Reference

### Functions/Subroutines

- subroutine [readmgt](#) (k)

*this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.*

#### 7.123.1 Detailed Description

file containing the subroutine readmgt

#### Author

modified by Javier Burguete

#### 7.123.2 Function/Subroutine Documentation

##### 7.123.2.1 readmgt()

```
subroutine readmgt (
    integer, intent(in) k )
```

this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.

## Parameters

|    |   |                   |
|----|---|-------------------|
| in | k | HRU number (none) |
|----|---|-------------------|

## 7.124 readmon.f90 File Reference

### Functions/Subroutines

- subroutine [readmon](#) (i)

*reads in the input data for the recmon command*

### 7.124.1 Detailed Description

file containing the subroutine readmon

Author

modified by Javier Burguete

## 7.125 readops.f90 File Reference

### Functions/Subroutines

- subroutine [readops](#) (k)

*this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.*

### 7.125.1 Detailed Description

file containing the subroutine readops

Author

modified by Javier Burguete

### 7.125.2 Function/Subroutine Documentation

#### 7.125.2.1 readops()

```
subroutine readops (
    integer, intent(in) k )
```

this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.

Parameters

|    |   |                   |
|----|---|-------------------|
| in | k | HRU number (none) |
|----|---|-------------------|

## 7.126 readpest.f90 File Reference

### Functions/Subroutines

- subroutine [readpest](#)

*this subroutine reads parameters from the toxin/pesticide database (pest.dat)*

### 7.126.1 Detailed Description

file containing the subroutine readpest

Author

modified by Javier Burguete

## 7.127 readplant.f90 File Reference

### Functions/Subroutines

- subroutine [readplant](#)

*this subroutine reads input parameters from the landuse/landcover database (plant.dat)*

### 7.127.1 Detailed Description

file containing the subroutine readplant

Author

modified by Javier Burguete

## 7.128 readpnd.f90 File Reference

### Functions/Subroutines

- subroutine [readpnd](#) (i)

*This subroutine reads data from the HRU/subbasin pond input file (.pnd). This file contains data related to ponds and wetlands in the HRUs/subbasins.*

### 7.128.1 Detailed Description

file containing the subroutine readpnd

Author

modified by Javier Burguete

### 7.128.2 Function/Subroutine Documentation

#### 7.128.2.1 readpnd()

```
subroutine readpnd (  
    integer, intent(in) i )
```

This subroutine reads data from the HRU/subbasin pond input file (.pnd). This file contains data related to ponds and wetlands in the HRUs/subbasins.

## Parameters

|    |   |                        |
|----|---|------------------------|
| in | i | subbasin number (none) |
|----|---|------------------------|

## 7.129 readres.f90 File Reference

### Functions/Subroutines

- subroutine [readres](#) (i)  
*the purpose of this subroutine is to read in data from the reservoir input file (.res)*

#### 7.129.1 Detailed Description

file containing the subroutine readres

#### Author

modified by Javier Burguete

#### 7.129.2 Function/Subroutine Documentation

##### 7.129.2.1 readres()

```
subroutine readres (
    integer, intent(in) i )
```

the purpose of this subroutine is to read in data from the reservoir input file (.res)

## Parameters

|    |   |                         |
|----|---|-------------------------|
| in | i | reservoir number (none) |
|----|---|-------------------------|

## 7.130 readrte.f90 File Reference

### Functions/Subroutines

- subroutine [readrte](#)  
*this subroutine reads data from the reach (main channel) input file (.rte). This file contains data related to channel attributes. Only one reach file should be made for each subbasin. If multiple HRUs are modeled within a subbasin, the same .rte file should be listed for all HRUs in file.cio*



### 7.130.1 Detailed Description

file containing the subroutine readrte

Author

modified by Javier Burguete

## 7.131 readru.f90 File Reference

### Functions/Subroutines

- subroutine [readru](#) (i)  
*this subroutine reads data from the sub input file (.sub). This file contains data related to routing*

### 7.131.1 Detailed Description

file containing the subroutine readru

Author

modified by Javier Burguete

### 7.131.2 Function/Subroutine Documentation

#### 7.131.2.1 readru()

```
subroutine readru (
    integer, intent(in) i )
```

this subroutine reads data from the sub input file (.sub). This file contains data related to routing

Parameters

|    |   |                 |
|----|---|-----------------|
| in | i | subbasin number |
|----|---|-----------------|

## 7.132 readsdr.f90 File Reference

### Functions/Subroutines

- subroutine [readsdr](#) (j)  
*this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.*

### 7.132.1 Detailed Description

file containing the subroutine readsdr

Author

modified by Javier Burguete

### 7.132.2 Function/Subroutine Documentation

#### 7.132.2.1 readsdr()

```
subroutine readsdr (
    integer j )
```

this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.

Parameters

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.133 readsepticbz.f90 File Reference

### Functions/Subroutines

- subroutine [readsepticbz](#) (j)

*this subroutine reads data from the septic input file (.sep). This file contains information related to septic tanks modeled or defined at the watershed level*

### 7.133.1 Detailed Description

file containing the subroutine readsepticbz

Author

modified by Javier Burguete

### 7.133.2 Function/Subroutine Documentation

### 7.133.2.1 readsepticbz()

```
subroutine readsepticbz (  
    integer, intent(in) j )
```

this subroutine reads data from the septic input file (.sep). This file contains information related to septic tanks modeled or defined at the watershed level

## Parameters

|                 |                |                   |
|-----------------|----------------|-------------------|
| <code>in</code> | <code>j</code> | HRU number (none) |
|-----------------|----------------|-------------------|

## 7.134 readseptwq.f90 File Reference

### Functions/Subroutines

- subroutine [readseptwq](#)

*this subroutine reads input parameters from the sept wq database (septwq.dat). Information is used when a hru has septic tank.*

#### 7.134.1 Detailed Description

file containing the subroutine readseptwq

#### Author

C. Santhi, modified by Javier Burguete

### 7.134.2 Function/Subroutine Documentation

#### 7.134.2.1 readseptwq()

```
subroutine readseptwq ( )
```

this subroutine reads input parameters from the sept wq database (septwq.dat). Information is used when a hru has septic tank.

This routine was developed by C. Santhi. Inputs for this routine are provided in septwq.dat of septic documentation. Data were compiled from [\[4\]](#) and [\[3\]](#).

## 7.135 readsno.f90 File Reference

### Functions/Subroutines

- subroutine [readsno](#) (i)

*this subroutine reads snow data from the HRU/subbasin soil chemical input*

### 7.135.1 Detailed Description

file containing the subroutine readsno

Author

modified by Javier Burguete

### 7.135.2 Function/Subroutine Documentation

#### 7.135.2.1 readsno()

```
subroutine readsno (  
    integer, intent(in) i )
```

this subroutine reads snow data from the HRU/subbasin soil chemical input

Parameters

|    |   |                        |
|----|---|------------------------|
| in | i | subbasin number (none) |
|----|---|------------------------|

## 7.136 readsol.f90 File Reference

### Functions/Subroutines

- subroutine [readsol](#) (k)

*this subroutine reads data from the HRU/subbasin soil properties file (.sol). This file contains data related to soil physical properties and general chemical properties*

### 7.136.1 Detailed Description

file containing the subroutine readsol

Author

modified by Javier Burguete

### 7.136.2 Function/Subroutine Documentation

#### 7.136.2.1 readsol()

```
subroutine readsol (  
    integer, intent(in) k )
```

this subroutine reads data from the HRU/subbasin soil properties file (.sol). This file contains data related to soil physical properties and general chemical properties

## Parameters

|    |   |            |
|----|---|------------|
| in | k | HRU number |
|----|---|------------|

## 7.137 readsub.f90 File Reference

### Functions/Subroutines

- subroutine [readsub](#) (i)

*this subroutine reads data from the HRU/subbasin general input file (.sub). This file contains data related to general processes modeled at the HRU/subbasin level.*

#### 7.137.1 Detailed Description

file containing the subroutine readsub

## Author

modified by Javier Burguete

#### 7.137.2 Function/Subroutine Documentation

##### 7.137.2.1 readsub()

```
subroutine readsub (
    integer, intent(in) i )
```

this subroutine reads data from the HRU/subbasin general input file (.sub). This file contains data related to general processes modeled at the HRU/subbasin level.

## Parameters

|    |   |                        |
|----|---|------------------------|
| in | i | subbasin number (none) |
|----|---|------------------------|

## 7.138 readswq.f90 File Reference

### Functions/Subroutines

- subroutine [readswq](#)

*this subroutine reads parameters from the subbasin instream water quality file (.swq) and initializes the QUAL2E variables which apply to the individual subbasins*

### 7.138.1 Detailed Description

file containing the subroutine readswq

Author

modified by Javier Burguete

## 7.139 readtill.f90 File Reference

### Functions/Subroutines

- subroutine [readtill](#)  
*this subroutine reads input data from tillage database (till.dat)*

### 7.139.1 Detailed Description

file containing the subroutine readtill

Author

modified by Javier Burguete

## 7.140 readurban.f90 File Reference

### Functions/Subroutines

- subroutine [readurban](#)  
*this subroutine reads input parameters from the urban database (urban.dat). Information from this database is used only if the urban buildup/washoff routines are selected for the modeling of urban areas*

### 7.140.1 Detailed Description

file containing the subroutine readurban

Author

modified by Javier Burguete

## 7.141 readwgn.f90 File Reference

### Functions/Subroutines

- subroutine [readwgn](#) (ii)  
*this subroutine reads the HRU weather generator parameters from the .wgn file*

### 7.141.1 Detailed Description

file containing the subroutine readwgn

Author

modified by Javier Burguete

### 7.141.2 Function/Subroutine Documentation

#### 7.141.2.1 readwgn()

```
subroutine readwgn (
    integer, intent(in) ii )
```

this subroutine reads the HRU weather generator parameters from the .wgn file

Parameters

|    |    |                        |
|----|----|------------------------|
| in | ii | subbasin number (none) |
|----|----|------------------------|

## 7.142 readwus.f90 File Reference

### Functions/Subroutines

- subroutine [readwus](#) (i)

*This subroutine reads data from the HRU/subbasin water use input file (.wus). The water use file extracts water from the subbasin and it is considered to be lost from the watershed. These variables should be used to remove water transported outside the watershed.*

#### 7.142.1 Detailed Description

file containing the subroutine readwus

Author

modified by Javier Burguete

#### 7.142.2 Function/Subroutine Documentation



**7.142.2.1 readwus()**

```
subroutine readwus (
    integer, intent(in) i )
```

This subroutine reads data from the HRU/subbasin water use input file (.wus). The water use file extracts water from the subbasin and it is considered to be lost from the watershed. These variables should be used to remove water transported outside the watershed.

**Parameters**

|    |   |                 |
|----|---|-----------------|
| in | i | subbasin number |
|----|---|-----------------|

**7.143 readwwq.f90 File Reference****Functions/Subroutines**

- subroutine [readwwq](#)  
*this subroutine reads the watershed stream water quality input data (.wwq file) and initializes the QUAL2E variables which apply to the entire watershed*

**7.143.1 Detailed Description**

file containing the subroutine readwwq

**Author**

modified by Javier Burguete

**7.144 readyr.f90 File Reference****Functions/Subroutines**

- subroutine [readyr](#) (i)  
*reads in the input data for the recyear command*

**7.144.1 Detailed Description**

file containing the subroutine readyr

**Author**

modified by Javier Burguete

**7.144.2 Function/Subroutine Documentation****7.144.2.1 readyr()**

```
subroutine readyr (
    integer, intent(in) i )
```

reads in the input data for the recyear command

## Parameters

|    |   |                         |
|----|---|-------------------------|
| in | i | reservoir number (none) |
|----|---|-------------------------|

## 7.145 resetlu.f90 File Reference

### Functions/Subroutines

- subroutine [resetlu](#)

*this subroutine reads data from the HRU/subbasin management input file (.mgt). This file contains data related to management practices used in the HRU/subbasin.*

#### 7.145.1 Detailed Description

file containing the subroutine resetlu

#### Author

modified by Javier Burguete

## 7.146 rhgen.f90 File Reference

### Functions/Subroutines

- subroutine [rhgen](#) (j)

*this subroutine generates weather relative humidity, solar radiation, and wind speed.*

#### 7.146.1 Detailed Description

file containing the subroutine rhgen

#### Author

modified by Javier Burguete

## 7.147 rootfr.f90 File Reference

### Functions/Subroutines

- subroutine [rootfr](#) (j)

*this subroutine distributes dead root mass through the soil profile*

### 7.147.1 Detailed Description

file containing the subroutine rootfr

#### Author

Armen R. Kemanian,  
modified by Javier Burguete

### 7.147.2 Function/Subroutine Documentation

#### 7.147.2.1 rootfr()

```
subroutine rootfr (
    integer, intent(in) j )
```

this subroutine distributes dead root mass through the soil profile

#### Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.148 rsedaa.f90 File Reference

### Functions/Subroutines

- subroutine [rsedaa](#) (years)  
*this subroutine writes the annual reach output to the .sed file*

### 7.148.1 Detailed Description

file containing the subroutine rsedaa

#### Author

modified by Javier Burguete

### 7.148.2 Function/Subroutine Documentation

#### 7.148.2.1 rsedaa()

```
subroutine rsedaa (
    real*8, intent(in) years )
```

this subroutine writes the annual reach output to the .sed file

## Parameters

|              |                              |
|--------------|------------------------------|
| <i>years</i> | length of simulation (years) |
|--------------|------------------------------|

## 7.149 **rseday.f90** File Reference

### Functions/Subroutines

- subroutine **rseday**

#### 7.149.1 Detailed Description

file containing the subroutine **rseday**

## Author

modified by Javier Burguete

## 7.150 **rsedmon.f90** File Reference

### Functions/Subroutines

- subroutine **rsedmon** (mdays)  
*this subroutine writes the monthly reach output to the .sed file*

#### 7.150.1 Detailed Description

file containing the subroutine **rsedmon**

## Author

modified by Javier Burguete

#### 7.150.2 Function/Subroutine Documentation

##### 7.150.2.1 **rsedmon()**

```
subroutine rsedmon (  
    integer, intent(in) mdays )
```

this subroutine writes the monthly reach output to the .sed file

## Parameters

|    |       |                                   |
|----|-------|-----------------------------------|
| in | mdays | number of days simulated in month |
|----|-------|-----------------------------------|

## 7.151 rsedyr.f90 File Reference

### Functions/Subroutines

- subroutine [rsedyr](#)  
*this subroutine writes the yearly reach output to the .sed file*

#### 7.151.1 Detailed Description

file containing the subroutine rsedyr

## Author

modified by Javier Burguete

## 7.152 rteinit.f90 File Reference

### Functions/Subroutines

- subroutine [rteinit](#)  
*This subroutine reads in the areas associated with files processed with the recday, recepic, recmon and recyear commands, calculates subbasin areas, calculates reach and hydrograph node drainage areas.*

#### 7.152.1 Detailed Description

file containing the subroutine rteinit

## Author

modified by Javier Burguete

## 7.153 sat\_excess.f90 File Reference

### Functions/Subroutines

- subroutine [sat\\_excess](#) (j1, j)  
*this subroutine is the master soil percolation component*

### 7.153.1 Detailed Description

file containing the subroutine sat\_excess

Author

modified by Javier Burguete

### 7.153.2 Function/Subroutine Documentation

#### 7.153.2.1 sat\_excess()

```
subroutine sat_excess (
    integer, intent(in) j1,
    integer, intent(in) j )
```

this subroutine is the master soil percolation component

Parameters

|    |           |            |
|----|-----------|------------|
| in | <i>j1</i> | counter    |
| in | <i>j</i>  | HRU number |

## 7.154 sched\_mgt.f90 File Reference

### Functions/Subroutines

- subroutine [sched\\_mgt](#) (j)  
*this subroutine performs all management operations*

#### 7.154.1 Detailed Description

file containing the subroutine sched\_mgt

Author

modified by Javier Burguete

### 7.154.2 Function/Subroutine Documentation

#### 7.154.2.1 sched\_mgt()

```
subroutine sched_mgt (
    integer, intent(in) j )
```

this subroutine performs all management operations

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.155 schedule\_ops.f90 File Reference

### Functions/Subroutines

- subroutine [schedule\\_ops](#) (j)  
*this subroutine controls the simulation of the land phase of the hydrologic cycle*

#### 7.155.1 Detailed Description

file containing the subroutine schedule\_ops

#### Author

modified by Javier Burguete

#### 7.155.2 Function/Subroutine Documentation

##### 7.155.2.1 schedule\_ops()

```
subroutine schedule_ops (
    integer, intent(in) j )
```

this subroutine controls the simulation of the land phase of the hydrologic cycle

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.156 sim\_inityr.f90 File Reference

### Functions/Subroutines

- subroutine [sim\\_inityr](#)  
*this subroutine initializes variables at the beginning of the year*

### 7.156.1 Detailed Description

file containing the subroutine `sim_inityr`

Author

modified by Javier Burguete

## 7.157 `simulate.f90` File Reference

### Functions/Subroutines

- subroutine [simulate](#)  
*this subroutine contains the loops governing the modeling of processes in the watershed*

### 7.157.1 Detailed Description

file containing the subroutine `simulate`

Author

modified by Javier Burguete

## 7.158 `slrgen.f90` File Reference

### Functions/Subroutines

- subroutine [slrgen](#) (*j*)  
*this subroutine generates solar radiation*

### 7.158.1 Detailed Description

file containing the subroutine `slrgen`

Author

modified by Javier Burguete

### 7.158.2 Function/Subroutine Documentation

#### 7.158.2.1 `slrgen()`

```
subroutine slrgen (  
    integer, intent(in) j )
```

this subroutine generates solar radiation



## Parameters

|                 |                |            |
|-----------------|----------------|------------|
| <code>in</code> | <code>j</code> | HRU number |
|-----------------|----------------|------------|

## 7.159 smeas.f90 File Reference

### Functions/Subroutines

- subroutine [smeas](#)

*this subroutine reads in daily solar radiation data and assigns the values to the proper HRUs*

#### 7.159.1 Detailed Description

file containing the subroutine smeas

#### Author

modified by Javier Burguete

## 7.160 snom.f90 File Reference

### Functions/Subroutines

- subroutine [snom](#) (`j`)

*this subroutine predicts daily snom melt when the average air temperature exceeds 0 degrees Celsius*

#### 7.160.1 Detailed Description

file containing the subroutine snom

#### Author

modified by Javier Burguete

#### 7.160.2 Function/Subroutine Documentation

##### 7.160.2.1 snom()

```
subroutine snom (
    integer, intent(in) j )
```

this subroutine predicts daily snom melt when the average air temperature exceeds 0 degrees Celsius

## Parameters

|                 |                |            |
|-----------------|----------------|------------|
| <code>in</code> | <code>j</code> | HRU number |
|-----------------|----------------|------------|

## 7.161 soil\_chem.f90 File Reference

### Functions/Subroutines

- subroutine [soil\\_chem](#) (ii)  
*this subroutine initializes soil chemical properties*

#### 7.161.1 Detailed Description

file containing the subroutine `soil_chem`

## Author

modified by Javier Burguete

#### 7.161.2 Function/Subroutine Documentation

##### 7.161.2.1 soil\_chem()

```
subroutine soil_chem (  
    integer, intent(in) ii )
```

this subroutine initializes soil chemical properties

## Parameters

|                 |                 |            |
|-----------------|-----------------|------------|
| <code>in</code> | <code>ii</code> | HRU number |
|-----------------|-----------------|------------|

## 7.162 soil\_phys.f90 File Reference

### Functions/Subroutines

- subroutine [soil\\_phys](#) (ii)  
*this subroutine initializes soil physical properties*

### 7.162.1 Detailed Description

file containing the subroutine soil\_phys

Author

modified by Javier Burguete

### 7.162.2 Function/Subroutine Documentation

#### 7.162.2.1 soil\_phys()

```
subroutine soil_phys (
    integer, intent(in) ii )
```

this subroutine initializes soil physical properties

Parameters

|    |    |            |
|----|----|------------|
| in | ii | HRU number |
|----|----|------------|

## 7.163 soil\_write.f90 File Reference

### Functions/Subroutines

- subroutine [soil\\_write](#) (i)  
*this subroutine writes output to the output.sol file*

#### 7.163.1 Detailed Description

file containing the subroutine soil\_write

Author

modified by Javier Burguete

### 7.163.2 Function/Subroutine Documentation

#### 7.163.2.1 soil\_write()

```
subroutine soil_write (
    integer, intent(in) i )
```

this subroutine writes output to the output.sol file

## Parameters

|    |   |  |
|----|---|--|
| in | i | current day in simulation - loop counter (julian date) |
|----|---|--|

## 7.164 solp.f90 File Reference

### Functions/Subroutines

- subroutine [solp](#) (j)

*this subroutine calculates the amount of phosphorus lost from the soil profile in runoff and the movement of soluble phosphorus from the first to the second layer via percolation*

#### 7.164.1 Detailed Description

file containing the subroutine solp

## Author

modified by Javier Burguete

#### 7.164.2 Function/Subroutine Documentation

##### 7.164.2.1 solp()

```
subroutine solp (
    integer, intent(in) j )
```

this subroutine calculates the amount of phosphorus lost from the soil profile in runoff and the movement of soluble phosphorus from the first to the second layer via percolation

## Parameters

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.165 solt.f90 File Reference

### Functions/Subroutines

- subroutine [solt](#) (j)

*this subroutine estimates daily average temperature at the bottom of each soil layer @parameter[in] j HRU number*

### 7.165.1 Detailed Description

file containing the subroutine solt

Author

modified by Javier Burguete

## 7.166 std1.f90 File Reference

### Functions/Subroutines

- subroutine [std1](#)

*this subroutine writes general information to the standard output file and header lines to miscellaneous output files*

### 7.166.1 Detailed Description

file containing the subroutine std1

Author

modified by Javier Burguete

## 7.167 std2.f90 File Reference

### Functions/Subroutines

- subroutine [std2](#)

*this subroutine writes general information to the standard output file and to miscellaneous output files*

### 7.167.1 Detailed Description

file containing the subroutine std2

Author

modified by Javier Burguete

## 7.168 std3.f90 File Reference

### Functions/Subroutines

- subroutine [std3](#)

*this subroutine writes the annual table header to the standard output file*

### 7.168.1 Detailed Description

file containing the subroutine std3

Author

modified by Javier Burguete

## 7.169 storeinitial.f90 File Reference

### Functions/Subroutines

- subroutine [storeinitial](#)  
*this subroutine saves initial values for variables that must be reset to rerun the simulation for different real time weather scenarios*

### 7.169.1 Detailed Description

file containing the subroutine storeinitial

Author

modified by Javier Burguete

## 7.170 subbasin.f90 File Reference

### Functions/Subroutines

- subroutine [subbasin](#) (i)  
*this subroutine controls the simulation of the land phase of the hydrologic cycle*

### 7.170.1 Detailed Description

file containing the subroutine subbasin

Author

modified by Javier Burguete

### 7.170.2 Function/Subroutine Documentation

#### 7.170.2.1 subbasin()

```
subroutine subbasin (  
    integer, intent(in) i )
```

this subroutine controls the simulation of the land phase of the hydrologic cycle

## Parameters

|    |   |  |
|----|---|--|
| in | i | current day in simulation—loop counter (julian date) |
|----|---|--|

## 7.171 submon.f90 File Reference

### Functions/Subroutines

- subroutine [submon](#)  
*this subroutine writes monthly subbasin output to the output.sub file*

#### 7.171.1 Detailed Description

file containing the subroutine submon

## Author

modified by Javier Burguete

## 7.172 subyr.f90 File Reference

### Functions/Subroutines

- subroutine [subyr](#)  
*this subroutine writes annual subbasin output to the output.sub file*

#### 7.172.1 Detailed Description

file containing the subroutine subyr

## Author

modified by Javier Burguete

## 7.173 surface.f90 File Reference

### Functions/Subroutines

- subroutine [surface](#) (i, j)  
*this subroutine models surface hydrology at any desired time step*

### 7.173.1 Detailed Description

file containing the subroutine surface

Author

modified by Javier Burguete

### 7.173.2 Function/Subroutine Documentation

#### 7.173.2.1 surface()

```
subroutine surface (
    integer, intent(in) i,
    integer, intent(in) j )
```

this subroutine models surface hydrology at any desired time step

Parameters

|    |          |  |
|----|----------|--|
| in | <i>i</i> | current day in simulation-loop counter (julian date) |
| in | <i>j</i> | HRU number (none)                                    |

## 7.174 surfst\_h2o.f90 File Reference

### Functions/Subroutines

- subroutine [surfst\\_h2o](#) (j)

*this subroutine determines the net surface runoff reaching the main channel on a given day. The net amount of water reaching the main channel can include water in surface runoff from the previous day and will exclude surface runoff generated on the current day which takes longer than one day to reach the main channel*

#### 7.174.1 Detailed Description

file containing the subroutine surfst\_h2o

Author

modified by Javier Burguete

#### 7.174.2 Function/Subroutine Documentation



**7.174.2.1 surfst\_h2o()**

```
subroutine surfst_h2o (
    integer, intent(in) j )
```

this subroutine determines the net surface runoff reaching the main channel on a given day. The net amount of water reaching the main channel can include water in surface runoff from the previous day and will exclude surface runoff generated on the current day which takes longer than one day to reach the main channel

**Parameters**

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

**7.175 surq\_daycn.f90 File Reference****Functions/Subroutines**

- subroutine [surq\\_daycn](#) (j)  
*predicts daily runoff given daily precipitation and snow melt using a modified SCS curve number approach*

**7.175.1 Detailed Description**

file containing the subroutine surq\_daycn

**Author**

modified by Javier Burguete

**7.175.2 Function/Subroutine Documentation****7.175.2.1 surq\_daycn()**

```
subroutine surq_daycn (
    integer, intent(in) j )
```

predicts daily runoff given daily precipitation and snow melt using a modified SCS curve number approach

**Parameters**

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.176 surq\_greenampt.f90 File Reference

### Functions/Subroutines

- subroutine [surq\\_greenampt](#) (j)

*predicts daily runoff given breakpoint precipitation and snow melt using the Green & Ampt technique*

#### 7.176.1 Detailed Description

file containing the subroutine `surq_greenampt`

##### Author

modified by Javier Burguete

#### 7.176.2 Function/Subroutine Documentation

##### 7.176.2.1 `surq_greenampt()`

```
subroutine surq_greenampt (
    integer, intent(in) j )
```

predicts daily runoff given breakpoint precipitation and snow melt using the Green & Ampt technique

##### Parameters

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.177 swbl.f90 File Reference

### Functions/Subroutines

- subroutine [swbl](#) (snow, irrg)

*this subroutine checks the soil water balance at the end of the simulation*

#### 7.177.1 Detailed Description

file containing the subroutine `swbl`

##### Author

modified by Javier Burguete

## 7.177.2 Function/Subroutine Documentation

### 7.177.2.1 swbl()

```
subroutine swbl (
    real*8, intent(in) snow,
    real*8, intent(in) irrg )
```

this subroutine checks the soil water balance at the end of the simulation

#### Parameters

|    |             |  |
|----|-------------|--|
| in | <i>snow</i> | snow in watershed at end of simulation |
| in | <i>irrg</i> | irrigation water applied to watershed  |

## 7.178 swu.f90 File Reference

### Functions/Subroutines

- subroutine [swu](#) (j)

*this subroutine distributes potential plant evaporation through the root zone and calculates actual plant water use based on soil water availability. Also estimates water stress factor*

### 7.178.1 Detailed Description

file containing the subroutine swu

#### Author

modified by Javier Burguete

## 7.178.2 Function/Subroutine Documentation

### 7.178.2.1 swu()

```
subroutine swu (
    integer, intent(in) j )
```

this subroutine distributes potential plant evaporation through the root zone and calculates actual plant water use based on soil water availability. Also estimates water stress factor

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.179 tgen.f90 File Reference

### Functions/Subroutines

- subroutine [tgen](#) (j)

*this subroutine generates temperature data when the user chooses to simulate or when data is missing for particular days in the weather file*

#### 7.179.1 Detailed Description

file containing the subroutine tgen

## Author

modified by Javier Burguete

#### 7.179.2 Function/Subroutine Documentation

##### 7.179.2.1 tgen()

```
subroutine tgen (
    integer, intent(in) j )
```

this subroutine generates temperature data when the user chooses to simulate or when data is missing for particular days in the weather file

## Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.180 tillfactor.f90 File Reference

### Functions/Subroutines

- subroutine [tillfactor](#) (j, bmix, emix, dtil, sol\_thick)

*this procedure increases tillage factor (tillagef(l,j) per layer for each operation. The tillage factor settling will depend of soil moisture (tentatively) and must be called every day. For simplicity the settling is calculated now at the soil carbon sub because soil water content is available.*

### 7.180.1 Detailed Description

file containing the subroutine tillfactor

Author

modified by Javier Burguete

### 7.180.2 Function/Subroutine Documentation

#### 7.180.2.1 tillfactor()

```
subroutine tillfactor (
    integer, intent(in) j,
    real*8, intent(in) bmix,
    real*8, intent(inout) emix,
    real*8, intent(in) dtil,
    real*8, dimension(sol_nly(j)), intent(in) sol_thick )
```

this procedure increases tillage factor (tillagef(l,j) per layer for each operation. The tillage factor settling will depend of soil moisture (tentatively) and must be called every day. For simplicity the settling is calculated now at the soil carbon sub because soil water content is available.

#### Parameters

|         |                  |  |
|---------|------------------|--|
| in      | <i>j</i>         | HRU number (none)  |
| in      | <i>bmix</i>      | biological mixing efficiency: this number is zero for tillage operations (none)  |
| in, out | <i>emix</i>      | mixing efficiency (none)   |
| in      | <i>dtil</i>      | depth of mixing (mm)   |
| in      | <i>sol_thick</i> | <p>The tillage factor depends on the cumulative soil disturbance rating = csdr For simplicity, csdr is a function of emix. First step is to calculate "current" csdr by inverting tillage factor function. The effect of texture on tillage factor (ZZ) is removed first (and recovered at the end of the procedure).</p> $YY = \text{tillagef}(l, j) / ZZ$ <p>Since the tillage factor function is non linear, iterations are needed. <math>XX = 0.5</math> is the initial value that works OK for the range of values observed. If a layer is only partially tilled then emix is corrected accordingly</p> |

## 7.181 tmeas.f90 File Reference

### Functions/Subroutines

- subroutine [tmeas](#)

*this subroutine reads in temperature data and assigns it to the HRUs*

### 7.181.1 Detailed Description

file containing the subroutine tmeas

Author

modified by Javier Burguete

## 7.182 tran.f90 File Reference

### Functions/Subroutines

- subroutine [tran](#) (j)  
*this subroutine computes tributary channel transmission losses*

### 7.182.1 Detailed Description

file containing the subroutine tran

Author

modified by Javier Burguete

### 7.182.2 Function/Subroutine Documentation

#### 7.182.2.1 tran()

```
subroutine tran (
    integer, intent(in) j )
```

this subroutine computes tributary channel transmission losses

Parameters

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.183 tstr.f90 File Reference

### Functions/Subroutines

- subroutine [tstr](#) (j)  
*computes temperature stress for crop growth - strtmp*

### 7.183.1 Detailed Description

file containing the subroutine tstr

Author

modified by Javier Burguete

### 7.183.2 Function/Subroutine Documentation

#### 7.183.2.1 tstr()

```
subroutine tstr (
    integer, intent(in) j )
```

computes temperature stress for crop growth - strtmp

Parameters

|    |   |            |
|----|---|------------|
| in | j | HRU number |
|----|---|------------|

## 7.184 ttcoef.f90 File Reference

### Functions/Subroutines

- subroutine [ttcoef](#) (k)  
*this subroutine computes travel time coefficients for routing along the main channel*

### 7.184.1 Detailed Description

file containing the subroutine ttcoef

Author

modified by Javier Burguete

### 7.184.2 Function/Subroutine Documentation

#### 7.184.2.1 ttcoef()

```
subroutine ttcoef (
    integer, intent(in) k )
```

this subroutine computes travel time coefficients for routing along the main channel

## Parameters

|    |   |            |
|----|---|------------|
| in | k | HRU number |
|----|---|------------|

## 7.185 ttcoef\_wway.f90 File Reference

### Functions/Subroutines

- subroutine [ttcoef\\_wway](#) (j)  
*this subroutine computes travel time coefficients for routing along the main channel - grassed waterways*

#### 7.185.1 Detailed Description

file containing the subroutine ttcoef\_wway

##### Author

modified by Javier Burguete

## 7.186 varinit.f90 File Reference

### Functions/Subroutines

- subroutine [varinit](#) (j)  
*this subroutine initializes variables for the daily simulation of the land phase of the hydrologic cycle (the subbasin command loop)*

#### 7.186.1 Detailed Description

file containing the subroutine varinit

##### Author

modified by Javier Burguete

#### 7.186.2 Function/Subroutine Documentation

##### 7.186.2.1 varinit()

```
subroutine varinit (
    integer, intent(in) j )
```

this subroutine initializes variables for the daily simulation of the land phase of the hydrologic cycle (the subbasin command loop)



## Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.187 vbl.f90 File Reference

### Functions/Subroutines

- subroutine [vbl](#) (evx, spx, pp, qin, ox, vx1, vy, yi, yo, ysx, vf, vyf, aha)  
*this subroutine checks the water and sediment balance for ponds and reservoirs at the end of a simulation*

#### 7.187.1 Detailed Description

file containing the subroutine vbl

## Author

modified by Javier Burguete

#### 7.187.2 Function/Subroutine Documentation

##### 7.187.2.1 vbl()

```
subroutine vbl (
    real*8, intent(in) evx,
    real*8, intent(in) spx,
    real*8, intent(in) pp,
    real*8, intent(in) qin,
    real*8, intent(in) ox,
    real*8, intent(inout) vx1,
    real*8, intent(inout) vy,
    real*8, intent(in) yi,
    real*8, intent(in) yo,
    real*8, intent(in) ysx,
    real*8, intent(in) vf,
    real*8, intent(in) vyf,
    real*8, intent(in) aha )
```

this subroutine checks the water and sediment balance for ponds and reservoirs at the end of a simulation

## Parameters

|    |            |                             |
|----|------------|-----------------------------|
| in | <i>evx</i> | evaporation from water body |
| in | <i>spx</i> | seepage from water body     |
| in | <i>pp</i>  | precipitation on water body |
| in | <i>qin</i> | water entering water body   |

**Parameters**

|         |     |  |
|---------|-----|--|
| in      | ox  | water leaving water body   |
| in, out | vx1 | (in) volume of water in water body at beginning of simulation<br>(out) dfw expressed as depth over drainage area         |
| in, out | vy  | (in) sediment in water body at beginning of simulation<br>(out) dfy expressed as loading per unit area for drainage area |
| in      | yi  | sediment entering water body   |
| in      | yo  | sediment leaving water body  |
| in      | ysx | change in sediment level in water body   |
| in      | vf  | volume of water in water body at end of simulation   |
| in      | vyf | sediment in water body at end of simulation  |
| in      | aha | area draining into water body  |

## 7.188 volq.f90 File Reference

### Functions/Subroutines

- subroutine [volq](#) (j)

*call subroutines to calculate the current day's CN for the HRU and to calculate surface runoff*

#### 7.188.1 Detailed Description

file containing the subroutine volq

**Author**

modified by Javier Burguete

#### 7.188.2 Function/Subroutine Documentation

##### 7.188.2.1 volq()

```
subroutine volq (
    integer, intent(in) j )
```

call subroutines to calculate the current day's CN for the HRU and to calculate surface runoff

**Parameters**

|    |   |                   |
|----|---|-------------------|
| in | j | HRU number (none) |
|----|---|-------------------|

## 7.189 water\_hru.f90 File Reference

### Functions/Subroutines

- subroutine [water\\_hru](#) (j)

*this subroutine compute pet and et using Priestly-Taylor and a coefficient*

### 7.189.1 Detailed Description

file containing the subroutine water\_hru

#### Author

modified by Javier Burguete

### 7.189.2 Function/Subroutine Documentation

#### 7.189.2.1 water\_hru()

```
subroutine water_hru (
    integer, intent(in) j )
```

this subroutine compute pet and et using Priestly-Taylor and a coefficient

#### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.190 wattable.f90 File Reference

### Functions/Subroutines

- subroutine [wattable](#) (j)

*this subroutine is the master soil percolation component. param[in] j HRU number*

### 7.190.1 Detailed Description

file containing the subroutine wattable

#### Author

modified by Javier Burguete

## 7.191 weatgn.f90 File Reference

### Functions/Subroutines

- subroutine [weatgn](#) (*j*)

*this subroutine generates weather parameters used to simulate the impact of precipitation on the other climatic processes*

#### 7.191.1 Detailed Description

file containing the subroutine weatgn

##### Author

modified by Javier Burguete

#### 7.191.2 Function/Subroutine Documentation

##### 7.191.2.1 weatgn()

```
subroutine weatgn (
    integer, intent(in) j )
```

this subroutine generates weather parameters used to simulate the impact of precipitation on the other climatic processes

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.192 wmeas.f90 File Reference

### Functions/Subroutines

- subroutine [wmeas](#)

*this subroutine reads in wind speed data from file and assigns the data to HRUs*

#### 7.192.1 Detailed Description

file containing the subroutine wmeas

##### Author

modified by Javier Burguete

## 7.193 wndgen.f90 File Reference

### Functions/Subroutines

- subroutine [wndgen](#) (j)  
*this subroutine generates wind speed*

#### 7.193.1 Detailed Description

file containing the subroutine wndgen

##### Author

modified by Javier Burguete

#### 7.193.2 Function/Subroutine Documentation

##### 7.193.2.1 wndgen()

```
subroutine wndgen (
    integer, intent(in) j )
```

this subroutine generates wind speed

##### Parameters

|    |          |            |
|----|----------|------------|
| in | <i>j</i> | HRU number |
|----|----------|------------|

## 7.194 writea.f90 File Reference

### Functions/Subroutines

- subroutine [writea](#) (i)  
*this subroutine writes annual output*

#### 7.194.1 Detailed Description

file containing the subroutine writea

##### Author

modified by Javier Burguete

## 7.194.2 Function/Subroutine Documentation

### 7.194.2.1 writea()

```
subroutine writea (
    integer, intent(in) i )
```

this subroutine writes annual output

#### Parameters

|    |   |   |
|----|---|---|
| in | i | current day of simulation (julian date) |
|----|---|---|

## 7.195 writed.f90 File Reference

### Functions/Subroutines

- subroutine [writed](#)  
*this subroutine contains the daily output writes*

### 7.195.1 Detailed Description

file containing the subroutine writed

#### Author

modified by Javier Burguete

## 7.196 writem.f90 File Reference

### Functions/Subroutines

- subroutine [writem](#) (i)  
*this subroutine writes monthly output*

### 7.196.1 Detailed Description

file containing the subroutine writem

#### Author

modified by Javier Burguete

## 7.196.2 Function/Subroutine Documentation

### 7.196.2.1 writem()

```
subroutine writem (
    integer, intent(in) i )
```

this subroutine writes monthly output

#### Parameters

|    |   |   |
|----|---|---|
| in | i | current day of simulation (julian date) |
|----|---|---|

## 7.197 xmon.f90 File Reference

### Functions/Subroutines

- subroutine [xmon](#)  
*this subroutine determines the month, given the julian date and leap year flag*

### 7.197.1 Detailed Description

file containing the subroutine xmon

#### Author

modified by Javier Burguete

## 7.198 ysed.f90 File Reference

### Functions/Subroutines

- subroutine [ysed](#) (iwave, j)  
*this subroutine predicts daily soil loss caused by water erosion using the modified universal soil loss equation*

### 7.198.1 Detailed Description

file containing the subroutine ysed

#### Author

modified by Javier Burguete

## 7.198.2 Function/Subroutine Documentation

### 7.198.2.1 ysed()

```
subroutine ysed (
    integer, intent(in) iwave,
    integer, intent(in) j )
```

this subroutine predicts daily soil loss caused by water erosion using the modified universal soil loss equation

#### Parameters

|    |              |   |
|----|--------------|---|
| in | <i>iwave</i> | flag to differentiate calculation of HRU and subbasin sediment calculation (none)<br>iwave = 0 for HRU<br>iwave = subbasin # for subbasin |
| in | <i>j</i>     | HRU number  |

## 7.199 zero0.f90 File Reference

### Functions/Subroutines

- subroutine [zero0](#)  
*this subroutine initializes the values for some of the arrays*

### 7.199.1 Detailed Description

file containing the subroutine zero0

#### Author

modified by Javier Burguete

## 7.200 zero1.f90 File Reference

### Functions/Subroutines

- subroutine [zero1](#)  
*this subroutine initializes the values for some of the arrays*

### 7.200.1 Detailed Description

file containing the subroutine zero1

#### Author

modified by Javier Burguete



## 7.201 zero2.f90 File Reference

### Functions/Subroutines

- subroutine [zero2](#)  
*this subroutine zeros all array values*

#### 7.201.1 Detailed Description

file containing the subroutine zero2

Author

modified by Javier Burguete

## 7.202 zero\_urbn.f90 File Reference

### Functions/Subroutines

- subroutine [zero\\_urbn](#)  
*this subroutine zeros all array values used in urban modeling*

#### 7.202.1 Detailed Description

file containing the subroutine zero\_urbn

Author

modified by Javier Burguete

## 7.203 zeroini.f90 File Reference

### Functions/Subroutines

- subroutine [zeroini](#)  
*this subroutine zeros values for single array variables*

#### 7.203.1 Detailed Description

file containing the subroutine zeroini

Author

modified by Javier Burguete



# Bibliography

- [1] P Bratley, B L Fox, and L E Schrage. *A Guide to Simulation*. Springer-Verlag, New York, USA, 1983. [110](#)
- [2] Armen R Kemanian and Claudio O Stöckle. C-farm: A simple model to evaluate the carbon balance of soil profiles. *European Journal of Agronomy*, 32(1):22–29, 2010. [115](#)
- [3] J. E. McCray, S. L. Kirkland, R. L. Siegrist, and G. D. Thyne. Model parameters for simulating fate and transport of on-site wastewater nutrients. *Ground Water*, 43(4):628–639, 2005. [272](#)
- [4] R. L. Siegrist, J. McCray, L. Weintraub, C. Chen, J. Bagdol, P. Lemonds, S. Van Cuyk, K. Lowe, R. Goldstein, and J. Rada. Quantifying site-scale processes and watershed-scale cumulative effects of decentralized wastewater systems, project no. wu-ht-00-27. Prepared for the National Decentralized Water Resources Capacity Development Project, Washington University, St. Louis, MO, by the Colorado School of Mines, 2005. [272](#)
- [5] P.A. Vadas and Michael White. Validating soil phosphorus routines in the swat model. *Transactions of ASABE*, 53, 09 2010. [254](#)



# Index

albedo  
    albedo.f90, 105  
albedo.f90, 105  
    albedo, 105  
allocate\_parms.f90, 106  
alph  
    alph.f90, 106  
alph.f90, 106  
    alph, 106  
anfert  
    anfert.f90, 107  
anfert.f90, 107  
    anfert, 107  
apply  
    apply.f90, 108  
apply.f90, 107  
    apply, 108  
ascrv  
    ascrv.f90, 108  
ascrv.f90, 108  
    ascrv, 108  
atri  
    atri.f90, 109  
atri.f90, 109  
    atri, 109  
aunif  
    aunif.f90, 110  
aunif.f90, 110  
    aunif, 110  
autoirr  
    autoirr.f90, 111  
autoirr.f90, 111  
    autoirr, 111  
  
bacteria  
    bacteria.f90, 112  
bacteria.f90, 111  
    bacteria, 112  
bmpinit  
    bmpinit.f90, 112  
bmpinit.f90, 112  
    bmpinit, 112  
burnop  
    burnop.f90, 113  
burnop.f90, 113  
    burnop, 113  
  
canopyint.f90, 113  
caps  
    caps.f90, 114  
caps.f90, 114  
    caps, 114  
carbon  
    carbon\_new.f90, 115  
carbon\_new.f90, 114  
    carbon, 115  
carbon\_zhang2  
    carbon\_zhang2.f90, 116  
carbon\_zhang2.f90, 116  
    carbon\_zhang2, 116  
cfactor.f90, 116  
clgen  
    clgen.f90, 117  
clgen.f90, 117  
    clgen, 117  
clicon  
    clicon.f90, 118  
clicon.f90, 117  
    clicon, 118  
command  
    command.f90, 118  
command.f90, 118  
    command, 118  
conapply  
    conapply.f90, 119  
conapply.f90, 119  
    conapply, 119  
confert  
    confert.f90, 120  
confert.f90, 119  
    confert, 120  
crackflow.f90, 120  
crackvol.f90, 120  
curno  
    curno.f90, 121  
curno.f90, 121  
    curno, 121  
  
dailycn.f90, 122  
decay  
    decay.f90, 122  
decay.f90, 122  
    decay, 122  
depstor  
    depstor.f90, 123  
depstor.f90, 123  
    depstor, 123  
dormant  
    dormant.f90, 124  
dormant.f90, 123

- dormant, 124
- drains
  - drains.f90, 124
- drains.f90, 124
- drains, 124
- dstn1
  - dstn1.f90, 125
- dstn1.f90, 125
- dstn1, 125
- ee
  - ee.f90, 126
- ee.f90, 125
- ee, 126
- eiusle.f90, 126
- enrsb
  - enrsb.f90, 127
- enrsb.f90, 126
- enrsb, 127
- estimate\_ksat
  - estimate\_ksat.f90, 127
- estimate\_ksat.f90, 127
- estimate\_ksat, 127
- etact.f90, 128
- etpot
  - etpot.f90, 129
- etpot.f90, 128
- etpot, 129
- expo
  - expo.f90, 129
- expo.f90, 129
- expo, 129
- fert
  - fert.f90, 130
- fert.f90, 130
- fert, 130
- finalbal.f90, 130
- gcycl.f90, 131
- getallo.f90, 131
- graze
  - graze.f90, 132
- graze.f90, 131
- graze, 132
- grow
  - grow.f90, 132
- grow.f90, 132
- grow, 132
- gwmmod
  - gwmmod.f90, 133
- gwmmod.f90, 133
- gwmmod, 133
- gwmmod\_deep
  - gwmmod\_deep.f90, 134
- gwmmod\_deep.f90, 133
- gwmmod\_deep, 134
- h2omgt\_init.f90, 134
- harvestop
  - harvestop.f90, 135
- harvestop.f90, 134
- harvestop, 135
- harvkillop
  - harvkillop.f90, 135
- harvkillop.f90, 135
- harvkillop, 135
- headout.f90, 136
- hmeas.f90, 136
- hruaa
  - hruaa.f90, 137
- hruaa.f90, 136
- hruaa, 137
- hruallo.f90, 137
- hrumon.f90, 137
- hruyr.f90, 138
- hydroinit.f90, 138
- icl
  - icl.f90, 139
- icl.f90, 138
- icl, 139
- igropt
  - parm, 101
- impnd\_init.f90, 139
- impndmon.f90, 139
- impndyr.f90, 140
- irrigate
  - irrigate.f90, 140
- irrigate.f90, 140
- irrigate, 140
- irrsb
  - irrsb.f90, 141
- irrsb.f90, 141
- irrsb, 141
- jdt
  - jdt.f90, 142
- jdt.f90, 141
- jdt, 142
- killop
  - killop.f90, 142
- killop.f90, 142
- killop, 142
- lid\_cistern
  - lid\_cistern.f90, 143
- lid\_cistern.f90, 143
- lid\_cistern, 143
- lid\_greenroof
  - lid\_greenroof.f90, 144
- lid\_greenroof.f90, 144
- lid\_greenroof, 144
- lid\_por pavement
  - lid\_por pavement.f90, 145
- lid\_por pavement.f90, 144
- lid\_por pavement, 145

lid\_raingarden  
     lid\_raingarden.f90, 146  
 lid\_raingarden.f90, 145  
     lid\_raingarden, 146  
 lidinit  
     lidinit.f90, 146  
 lidinit.f90, 146  
     lidinit, 146  
 lids  
     lids.f90, 147  
 lids.f90, 147  
     lids, 147  
 lwqdef  
     lwqdef.f90, 148  
 lwqdef.f90, 148  
     lwqdef, 148  
  
 main.f90, 148  
 modparm.f90, 149  
  
 ndenit  
     ndenit.f90, 235  
 ndenit.f90, 235  
     ndenit, 235  
 newtillmix  
     newtillmix.f90, 236  
 newtillmix.f90, 235  
     newtillmix, 236  
 nfix  
     nfix.f90, 237  
 nfix.f90, 236  
     nfix, 237  
 nitvol  
     nitvol.f90, 237  
 nitvol.f90, 237  
     nitvol, 237  
 nlch  
     nlch.f90, 238  
 nlch.f90, 238  
     nlch, 238  
 nminrl  
     nminrl.f90, 239  
 nminrl.f90, 238  
     nminrl, 239  
 npup  
     npup.f90, 239  
 npup.f90, 239  
     npup, 239  
 nrain  
     nrain.f90, 240  
 nrain.f90, 240  
     nrain, 240  
 nup  
     nup.f90, 241  
 nup.f90, 240  
     nup, 241  
 nuts  
     nuts.f90, 241  
 nuts.f90, 241  
  
     nuts, 241  
 openwth.f90, 242  
 operatn  
     operatn.f90, 242  
 operatn.f90, 242  
     operatn, 242  
 orgn  
     orgn.f90, 243  
 orgn.f90, 243  
     orgn, 243  
 orgncswat  
     orgncswat.f90, 244  
 orgncswat.f90, 243  
     orgncswat, 244  
 orgncswat2  
     orgncswat2.f90, 244  
 orgncswat2.f90, 244  
     orgncswat2, 244  
 origtile  
     origtile.f90, 245  
 origtile.f90, 245  
     origtile, 245  
 ovr\_sed.f90, 246  
  
 parm, 15  
     igropt, 101  
 parm::fcgd, 103  
 parm::regres, 103  
 parm::tair, 103  
 parm::theta, 103  
 percmacro  
     percmacro.f90, 246  
 percmacro.f90, 246  
     percmacro, 246  
 percmain  
     percmain.f90, 247  
 percmain.f90, 247  
     percmain, 247  
 percmicro  
     percmicro.f90, 248  
 percmicro.f90, 247  
     percmicro, 248  
 pestlch  
     pestlch.f90, 248  
 pestlch.f90, 248  
     pestlch, 248  
 pesty  
     pesty.f90, 249  
 pesty.f90, 249  
     pesty, 249  
 pgen  
     pgen.f90, 250  
 pgen.f90, 249  
     pgen, 250  
 pgenhr.f90, 250  
 pkq  
     pkq.f90, 251  
 pkq.f90, 250

- pkq, 251
- plantmod
  - plantmod.f90, 252
- plantmod.f90, 251
  - plantmod, 252
- plantop
  - plantop.f90, 252
- plantop.f90, 252
  - plantop, 252
- pmeas
  - pmeas.f90, 253
- pmeas.f90, 253
  - pmeas, 253
- pminrl
  - pminrl.f90, 254
- pminrl.f90, 253
  - pminrl, 254
- pminrl2
  - pminrl2.f90, 254
- pminrl2.f90, 254
  - pminrl2, 254
- psed
  - psed.f90, 255
- psed.f90, 255
  - psed, 255
- qman
  - qman.f90, 256
- qman.f90, 255
  - qman, 256
- rcha
  - rcha.f90, 257
- rcha.f90, 256
  - rcha, 257
- rchday.f90, 257
- rchmon
  - rchmon.f90, 258
- rchmon.f90, 257
  - rchmon, 258
- rchyr
  - rchyr.f90, 258
- rchyr.f90, 258
  - rchyr, 258
- readatmodep.f90, 259
- readbsn.f90, 259
- readchm
  - readchm.f90, 260
- readchm.f90, 259
  - readchm, 260
- readcnst
  - readcnst.f90, 260
- readcnst.f90, 260
  - readcnst, 260
- readfcst.f90, 261
- readfert.f90, 261
- readfig.f90, 261
- readfile.f90, 262
- readgw
  - readgw.f90, 262
- readgw.f90, 262
  - readgw, 262
- readhru
  - readhru.f90, 263
- readhru.f90, 263
  - readhru, 263
- readinpt.f90, 263
- readlup.f90, 264
- readlwq
  - readlwq.f90, 264
- readlwq.f90, 264
  - readlwq, 264
- readmgt
  - readmgt.f90, 265
- readmgt.f90, 265
  - readmgt, 265
- readmon.f90, 265
- readops
  - readops.f90, 266
- readops.f90, 266
  - readops, 266
- readpest.f90, 266
- readplant.f90, 267
- readpnd
  - readpnd.f90, 267
- readpnd.f90, 267
  - readpnd, 267
- readres
  - readres.f90, 268
- readres.f90, 268
  - readres, 268
- readrte.f90, 268
- readru
  - readru.f90, 269
- readru.f90, 269
  - readru, 269
- readsdr
  - readsdr.f90, 270
- readsdr.f90, 269
  - readsdr, 270
- readsepticbz
  - readsepticbz.f90, 270
- readsepticbz.f90, 270
  - readsepticbz, 270
- readseptwq
  - readseptwq.f90, 272
- readseptwq.f90, 272
  - readseptwq, 272
- readsno
  - readsno.f90, 273
- readsno.f90, 272
  - readsno, 273
- readsol
  - readsol.f90, 273
- readsol.f90, 273
  - readsol, 273
- readsub



- readsub.f90, 274
- readsub.f90, 274
  - readsub, 274
- readswq.f90, 274
- readtill.f90, 275
- readurban.f90, 275
- readwgn
  - readwgn.f90, 276
- readwgn.f90, 275
  - readwgn, 276
- readwus
  - readwus.f90, 276
- readwus.f90, 276
  - readwus, 276
- readwwq.f90, 277
- readyr
  - readyr.f90, 277
- readyr.f90, 277
  - readyr, 277
- resetlu.f90, 278
- rhgen.f90, 278
- rootfr
  - rootfr.f90, 279
- rootfr.f90, 278
  - rootfr, 279
- rsedaa
  - rsedaa.f90, 279
- rsedaa.f90, 279
  - rsedaa, 279
- rseday.f90, 280
- rsedmon
  - rsedmon.f90, 280
- rsedmon.f90, 280
  - rsedmon, 280
- rsedyr.f90, 281
- rteinit.f90, 281
- sat\_excess
  - sat\_excess.f90, 282
- sat\_excess.f90, 281
  - sat\_excess, 282
- sched\_mgt
  - sched\_mgt.f90, 282
- sched\_mgt.f90, 282
  - sched\_mgt, 282
- schedule\_ops
  - schedule\_ops.f90, 283
- schedule\_ops.f90, 283
  - schedule\_ops, 283
- sim\_inityr.f90, 283
- simulate.f90, 284
- slrgen
  - slrgen.f90, 284
- slrgen.f90, 284
  - slrgen, 284
- smeas.f90, 285
- snom
  - snom.f90, 285
- snom.f90, 285
  - snom, 285
- soil\_chem
  - soil\_chem.f90, 286
- soil\_chem.f90, 286
  - soil\_chem, 286
- soil\_phys
  - soil\_phys.f90, 287
- soil\_phys.f90, 286
  - soil\_phys, 287
- soil\_write
  - soil\_write.f90, 287
- soil\_write.f90, 287
  - soil\_write, 287
- solp
  - solp.f90, 288
- solp.f90, 288
  - solp, 288
- solt.f90, 288
- std1.f90, 289
- std2.f90, 289
- std3.f90, 289
- storeinitial.f90, 290
- subbasin
  - subbasin.f90, 290
- subbasin.f90, 290
  - subbasin, 290
- submon.f90, 291
- subyr.f90, 291
- surface
  - surface.f90, 292
- surface.f90, 291
  - surface, 292
- surfst\_h2o
  - surfst\_h2o.f90, 292
- surfst\_h2o.f90, 292
  - surfst\_h2o, 292
- surq\_daycn
  - surq\_daycn.f90, 293
- surq\_daycn.f90, 293
  - surq\_daycn, 293
- surq\_greenampt
  - surq\_greenampt.f90, 294
- surq\_greenampt.f90, 294
  - surq\_greenampt, 294
- swbl
  - swbl.f90, 295
- swbl.f90, 294
  - swbl, 295
- swu
  - swu.f90, 295
- swu.f90, 295
  - swu, 295
- tgen
  - tgen.f90, 296
- tgen.f90, 296
  - tgen, 296
- tillfactor
  - tillfactor.f90, 297

tillfactor.f90, [296](#)  
    tillfactor, [297](#)  
tmeas.f90, [297](#)  
tran  
    tran.f90, [298](#)  
tran.f90, [298](#)  
    tran, [298](#)  
tstr  
    tstr.f90, [299](#)  
tstr.f90, [298](#)  
    tstr, [299](#)  
ttcoef  
    ttcoef.f90, [299](#)  
ttcoef.f90, [299](#)  
    ttcoef, [299](#)  
ttcoef\_wway.f90, [300](#)  
  
varinit  
    varinit.f90, [300](#)  
varinit.f90, [300](#)  
    varinit, [300](#)  
vbl  
    vbl.f90, [301](#)  
vbl.f90, [301](#)  
    vbl, [301](#)  
volq  
    volq.f90, [302](#)  
volq.f90, [302](#)  
    volq, [302](#)  
  
water\_hru  
    water\_hru.f90, [303](#)  
water\_hru.f90, [303](#)  
    water\_hru, [303](#)  
wattable.f90, [303](#)  
weatgn  
    weatgn.f90, [304](#)  
weatgn.f90, [304](#)  
    weatgn, [304](#)  
wmeas.f90, [304](#)  
wndgen  
    wndgen.f90, [305](#)  
wndgen.f90, [305](#)  
    wndgen, [305](#)  
writea  
    writea.f90, [306](#)  
writea.f90, [305](#)  
    writea, [306](#)  
writed.f90, [306](#)  
writem  
    writem.f90, [307](#)  
writem.f90, [306](#)  
    writem, [307](#)  
  
xmon.f90, [307](#)  
  
ysed  
    ysed.f90, [308](#)  
ysed.f90, [307](#)  
    ysed, [308](#)  
zero0.f90, [308](#)  
zero1.f90, [308](#)  
zero2.f90, [309](#)  
zero\_urban.f90, [309](#)  
zeroini.f90, [309](#)