

```

1  pro make_model_header, outfile
2
3  ;;
4  ;;
5  ;; make_model_header: make a text format header file for the model output
6  ;;
7  ;; Inputs:
8  ;; * outfile = model output file in IDLsave format
9  ;;
10 ;; Written by Matthew Burger
11 ;; Version History:
12 ;;   3.2: 7/19/10
13 ;;   * converted to new architecture
14 ;;   3.1: 5/13/10
15 ;;   * Added num keyword
16 ;;   * Added code versions to the header
17 ;;   3.0: 5/10/10
18 ;;   * Created.
19 ;;
20 ;;
21
22 result = obj_new('IDL_Savefile', outfile)
23
24 ;; Extract identifying info
25 contents = result.contents()
26 id = {file:outfile, time:contents.date, user:contents.user, computer:contents.host}
27 t = strtrim(tag_names(id), 2)
28 idparam = strarr(n_elements(t))
29 idvalue = strarr(n_elements(t))
30 for i=0,n_elements(t)-1 do begin
31     idparam[i] = t[i]
32     idvalue[i] = string(id.(i))
33 endfor
34 idvalue = strtrim(idvalue, 2)
35
36 result.restore, 'output'
37 npackets = n_elements(*output.x)
38 result.restore, 'input'
39
40 ;; Extract geometry info
41 geometry = input.geometry
42 t = strtrim(tag_names(geometry), 2)
43 geoparam = strarr(1000)
44 geovalue = strarr(1000)
45 ct = 0
46 for i=0,n_elements(t)-1 do begin
47     if (ptr_valid(geometry.(i))) then for j=0,n_elements(*geometry.(i))-1 do begin
48         geoparam[ct] = t[i]
49         geovalue[ct] = string((*geometry.(i))[j])
50         ct++
51     endfor else begin

```

```

52   geoparam[ct] = t[i]
53   geovalue[ct] = string(geometry.(i))
54   ct++
55   endelse
56   endfor
57   geoparam = geoparam[0:ct-1]
58   geovalue = strtrim(geovalue[0:ct-1], 2)
59
60   ;; Extract Sticking_info
61   sticking_info = input.sticking_info
62   t = strtrim(tag_names(sticking_info), 2)
63   stickparam = strarr(n_elements(t))
64   stickvalue = strarr(n_elements(t))
65   for i=0,n_elements(t)-1 do begin
66     stickparam[i] = t[i]
67     stickvalue[i] = string(sticking_info.(i))
68   endfor
69   stickvalue = strtrim(stickvalue, 2)
70
71   ;; Extract Forces
72   forces = input.forces
73   t = strtrim(tag_names(forces), 2)
74   forceparam = strarr(n_elements(t))
75   forcevalue = strarr(n_elements(t))
76   for i=0,n_elements(t)-1 do begin
77     forceparam[i] = t[i]
78     forcevalue[i] = string(forces.(i))
79   endfor
80   forcevalue = strtrim(forcevalue, 2)
81
82   ;; Extract spatialdist
83   SpatialDist = input.SpatialDist
84   t = strtrim(tag_names(SpatialDist), 2)
85   spatparam = strarr(100)
86   spatvalue = strarr(100)
87   ct = 0
88   for i=0,n_elements(t)-1 do begin
89     n = n_elements(SpatialDist.(i))
90     if (n EQ 1) then begin
91       spatparam[ct] = t[i]
92       spatvalue[ct] = string(SpatialDist.(i))
93       ct++
94     endif else for j=0,n-1 do begin
95       spatparam[ct] = t[i] + strtrim(string(j),2)
96       spatvalue[ct] = string((SpatialDist.(i)[j]))
97       ct++
98     endfor
99   endfor
100   spatparam = spatparam[0:ct-1]
101   spatvalue = strtrim(spatvalue[0:ct-1], 2)
102

```

```

103 ;; Extract speeddist
104 SpeedDist = input.SpeedDist
105 t = strtrim(tag_names(SpeedDist), 2)
106 speedparam = strarr(n_elements(t))
107 speedvalue = strarr(n_elements(t))
108 for i=0,n_elements(t)-1 do begin
109   speedparam[i] = t[i]
110   speedvalue[i] = string(SpeedDist.(i))
111 endfor
112 speedvalue = strtrim(speedvalue, 2)
113
114 ;; Extract angular_dist
115 AngularDist = input.AngularDist
116 t = strtrim(tag_names(AngularDist), 2)
117 angparam = strarr(100)
118 angvalue = strarr(100)
119 ct = 0
120 for i=0,n_elements(t)-1 do begin
121   n = n_elements(AngularDist.(i))
122   if (n EQ 1) then begin
123     angparam[ct] = t[i]
124     angvalue[ct] = string(AngularDist.(i))
125     ct++
126   endif else for j=0,n-1 do begin
127     angparam[ct] = t[i] + strtrim(string(j),2)
128     angvalue[ct] = string((AngularDist.(i))[j])
129     ct++
130   endfor
131 endfor
132 angparam = angparam[0:ct-1]
133 angvalue = strtrim(angvalue[0:ct-1], 2)
134
135 ;; Extract PerturbVel
136 PerturbVel = input.PerturbVel
137 t = strtrim(tag_names(PerturbVel), 2)
138 pertparam = strarr(n_elements(t))
139 pertvalue = strarr(n_elements(t))
140 for i=0,n_elements(t)-1 do begin
141   pertparam[i] = t[i]
142   pertvalue[i] = string(PerturbVel.(i))
143 endfor
144 pertvalue = strtrim(pertvalue, 2)
145
146 ;; Extract plasma_info (if present)
147 PlasmaInfo = input.plasma_info
148 t = strtrim(tag_names(plasma_info), 2)
149 plasmparam = strarr(n_elements(t))
150 plasmavalue = strarr(n_elements(t))
151 for i=0,n_elements(t)-1 do begin
152   plasmparam[i] = t[i]
153   plasmavalue[i] = string(plasma_info.(i))

```

```

154 endfor
155 plasmavalue = strtrim(plasmavalue, 2)
156
157 ;; extract options
158 options = input.options
159 t = strtrim(tag_names(options), 2)
160 optparam = strarr(n_elements(t))
161 optvalue = strarr(n_elements(t))
162 for i=0,n_elements(t)-1 do begin
163   optparam[i] = t[i]
164   optvalue[i] = string(options.(i))
165 endfor
166 optvalue = strtrim(optvalue, 2)
167
168 ;; Extract version info
169 result.restore, 'version'
170 if (n_elements(version) LE 1) then stop
171 version = file_basename(version[1:*])
172 verparam = version
173 vervalue = version
174 q = stregex(verparam, '[0-9]\.[0-9]+$')
175 for i=0,n_elements(verparam)-1 do $
176   if (q[i] NE -1) then begin
177     verparam[i] = strmid(version[i], 0, q[i]-1)
178     vervalue[i] = strmid(version[i], q[i])
179   endif else vervalue[i] = 'XX'
180 obj_destroy, result
181
182 ;; Save header file
183 hdrfile = strmid(outputfile, 0, strlen(outputfile)-strlen('output')) + 'header'
184 print, hdrfile
185 openw, lun, hdrfile, width=100, /get_lun
186
187 form = '(A-30,A3,A-)'
188 for i=0,n_elements(idparam)-1 do printf, lun, 'id.' + idparam[i], ' = ', $
189   idvalue[i], format=form
190   printf, lun
191
192 printf, lun, 'savedpackets', ' = ', strint(npackets), format=form
193 printf, lun, 'output.totalsource', ' = ', strint(output.totalsource), format=form
194 for i=0,n_elements(*output.sourcefile)-1 do $
195   printf, lun, 'output.sourcefile', ' = ', (*output.sourcefile)[i], format=form
196   printf, lun
197
198 for i=0,n_elements(geoparam)-1 do printf, lun, 'geometry.' + geoparam[i], ' = ', $
199   geovalue[i], format=form
200   printf, lun
201
202 for i=0,n_elements(stickparam)-1 do printf, lun, 'sticking_info.' + stickparam[i], $
203   ' = ', stickvalue[i], format=form
204

```

```

205 printf, lun
206
207 for i=0,n_elements(forceparam)-1 do printf, lun, 'forces.' + forceparam[i], ' = ', $
208   forcevalue[i], format=form
209 printf, lun
210
211 for i=0,n_elements(spatparam)-1 do printf, lun, 'SpatialDist.' + spatparam[i], ' = ', $
212   spatvalue[i], format=form
213 printf, lun
214
215 for i=0,n_elements(speedparam)-1 do printf, lun, 'SpeedDist.' + speedparam[i], ' = ', $
216   speedvalue[i], format=form
217 printf, lun
218
219 for i=0,n_elements(angparam)-1 do printf, lun, 'AngularDist.' + angparam[i], ' = ', $
220   angvalue[i], format=form
221 printf, lun
222
223 for i=0,n_elements(pertparam)-1 do printf, lun, 'PerturbVel.' + pertparam[i], ' = ', $
224   pertvalue[i], format=form
225 printf, lun
226
227 for i=0,n_elements(plasparam)-1 do printf, lun, 'PlasmaInfo.' + plasmaparam[i], ' = ', $
228   plasmapvalue[i], format=form
229 printf, lun
230
231 for i=0,n_elements(optparam)-1 do printf, lun, 'options.' + optparam[i], ' = ', $
232   optvalue[i], format=form
233 printf, lun
234
235 printf, lun, 'Program Versions'
236 for i=0,n_elements(verparam)-1 do printf, lun, verparam[i], ' = ', vervalue[i], $
237   format=form
238 printf, lun
239
240 free_lun, lun
241 destroy_structure, output
242 destroy_structure, input
243
244 end

```