

```

1 function headername, outputfile
2
3 return, reform((strmid(outputfile, 0, strlen(outputfile)-6) + 'header')[0,*])
4
5 end
6
7
8
9 function outputname, headerfile
10
11 return, reform((strmid(headerfile, 0, strlen(headerfile)-6) + 'output')[0,*])
12
13 end
14
15
16
17 function output_filename, inputtemp, path=path, file=file
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```

Create a unique output filename
 These are not user readable filenames to keep them short. The header files are user readable.
 path = '/Volumes/DroboData/burger/modeloutputs/PLANET/' +
 TAA_DEG/ATOM/SPEEDDIST/SPATIALDIST/' or
 MOON/PHI_DEG/ATOM/SPEEDDIST/SPATIALDIST/'
 file = 'USER.HOSTNAME.###.output'
 Version History
 3.4: 9/28/2011
 * adding option to search locally
 3.3: 3 Jan 2011
 * changing path to DroboData
 3.1: 26 August 2010
 * Making a bit more of a directory tree
 * Making filenames a bit more unique
 * adding computer identifier to avoid an ambiguity
 3.0: Original
 common constants
 input = (isa(inputtemp, 'string')) ? inputs_restore(inputtemp) : inputtemp
 if (SystemConsts EQ !null) then SystemConstants, input.geometry.planet, SystemConsts
 Determine if startpoint = planet
 s = (where(strcmp(*SystemConsts.objects, input.geometry.startpoint, /fold)))[0]

```

52 ;; Create the path
53 path = input.geometry.planet + '/'
54
55 if (s EQ 0) then begin
56   taastr = strint(round(input.geometry.taa!/dtor))
57   path += taastr + '/'
58 endif else begin
59   path += input.geometry.startpoint + '/'
60
61   phistr = strint(round((*input.geometry.phi)[s!/dtor]))
62   path += phistr + '/'
63 endelse
64
65 path += input.options.atom + '/'
66 path += strlowcase(input.speeddist.type) + '/'
67 path += strlowcase(input.spatialdist.type) + '/'
68 path = strcompress(path, /remove_all)
69
70 sh = file_test(!model.SharedOutputPath)
71 loc = file_test(!model.LocalOutputPath)
72
73 case (1) of
74   (stuff EQ !null) and (sh): path = !model.SharedOutputPath + path
75   (stuff.local) and (loc): path = !model.LocalOutputPath + path
76   ~(stuff.local) and (sh): path = !model.SharedOutputPath + path
77   else: stop
78 endcase
79
80 if ~(file_test(path)) then file_mkdir, path
81
82 ;; Create the filename
83 filest = !model.user + '.' + !model.hostname + '.'
84
85 q = file_search(path, filest+'*.output', count=nq)
86
87 if (nq EQ 0) $
88   then filename = filest + '0000.output' $
89   else begin
90     file = file_basename(q, '.output')
91     num = max(fix(strexex(file, '[0-9]+$'), /extract))) + 1
92     case (1) of
93       (num LT 10): nn = '000' + strint(num)
94       (num LT 100): nn = '00' + strint(num)
95       (num LT 1000): nn = '0' + strint(num)
96       else: nn = strint(num)
97     endcase
98     filename = filest + nn + '.output'
99   endelse
100   if file_test(path+filename) then stop
101
102 return, path+filename

```

103
104 end