

```

1 | #include "cppdefs.h"
2 | -----
3 | !BOP
4 | !
5 | ! MODULE: input_netcdf
6 | !
7 | ! INTERFACE:
8 |   module input_netcdf
9 | !
10 | ! DESCRIPTION:
11 | !
12 | ! USES:
13 |   use time, only: julianday, secondsofday
14 |   use time, only: read_time_string, write_time_string
15 | #ifdef NETCDF_FMT
16 |   use netcdf
17 |   implicit none
18 | !
19 | ! default: all is private.
20 | private
21 | !
22 | ! PUBLIC MEMBER FUNCTIONS:
23 |   public open_restart, close_restart, check_restart_time, read_restart_data
24 | !
25 | ! REVISION HISTORY:
26 | !   Original author(s): Karsten Bolding and Jorn Bruggeman
27 | !
28 | !EOP
29 | -----
30 | !BOC
31 | !
32 | ! PRIVATE TYPES
33 |   integer, parameter, public :: maxpathlen=256
34 | !
35 | ! Information on an observed variable
36 | !
37 | ! PRIVATE DATA MEMBERS
38 | !   Pointers to first files with observed profiles and observed scalars.
39 | !
40 | ! PRIVATE PARAMETERS
41 |   integer                      :: ncid=-1
42 | !
43 | ! -----
44 | !
45 | ! contains
46 | ! -----
47 | ! -----
48 | !BOP
49 | !
50 | ! IROUTINE: Initialize input
51 | !
52 | ! INTERFACE:
53 |   subroutine init_input_netcdf(n)
54 | !
55 | ! DESCRIPTION:
56 | !
57 | ! INPUT PARAMETERS:
58 |   integer, intent(in), optional :: n
59 | !
60 | ! REVISION HISTORY:
61 | !   Original author(s): Karsten Bolding and Jorn Bruggeman
62 | !
63 | !EOP
64 | !
65 | ! -----
66 | !BOC
67 |   LEVEL1 'init_input'
68 |   LEVEL1 'done'
69 |
70 |   end subroutine init_input_netcdf
71 | !EOC
72 | !
73 | ! -----
74 | !BOP
75 | !
76 | ! IROUTINE:
77 | !
78 | ! INTERFACE:
79 |   subroutine open_restart(fn)
80 | !
81 | ! DESCRIPTION:
82 | !
83 | ! INPUT PARAMETERS:
84 |   character(len=*), intent(in) :: fn
85 | !
86 | ! REVISION HISTORY:
87 | !   Original author(s): Karsten Bolding and Jorn Bruggeman
88 | !
89 | ! LOCAL VARIABLES:
90 |   integer                      :: ierr
91 | !EOP
92 | ! -----
93 | !BOC
94 | !
95 |   ierr = nf90_open(trim(fn) // '.nc', NF90_NOWRITE, ncid)
96 |   if (ierr /= NF90_NOERR) call handle_err(ierr)
97 |
98 |   end subroutine open_restart

```

```

99 |!EOC
100 |
101 |-----
102 |!BOP
103 |!
104 |!IROUTINE:
105 |!
106 |!INTERFACE:
107 |  subroutine close_restart()
108 |!
109 |!DESCRIPTION:
110 |!
111 |!REVISION HISTORY:
112 |!  Original author(s): Karsten Bolding and Jorn Bruggeman
113 |!
114 |!LOCAL VARIABLES:
115 |  integer          :: ierr
116 |!EOP
117 |-----
118 |!BOC
119 |  ierr = nf90_close(ncid)
120 |  if (ierr /= NF90_NOERR) call handle_err(ierr)
121 |
122 |  end subroutine close_restart
123 |!EOC
124 |
125 |-----
126 |!BOP
127 |!
128 |!IROUTINE:
129 |!
130 |!INTERFACE:
131 |  subroutine check_restart_time(var_name)
132 |!
133 |!DESCRIPTION:
134 |!
135 |!INPUT PARAMETERS:
136 |  character(len=*), intent(in)          :: var_name
137 |!
138 |!REVISION HISTORY:
139 |!  Original author(s): Karsten Bolding and Jorn Bruggeman
140 |!
141 |!LOCAL VARIABLES:
142 |  integer          :: ierr, id
143 |  integer          :: jd,secs
144 |  character(len=256) :: units,timestr_out
145 |!EOP
146 |-----
147 |!BOC
148 |  ierr = nf90_inq_varid(ncid, trim(var_name), id)
149 |  if (ierr /= NF90_NOERR) then
150 |    FATAL 'Could not find time-variable in restart file'
151 |    stop 'restart_file()'
152 |  else
153 |    ierr = nf90_get_att(ncid, id, 'units', units)
154 |    call read_time_string(trim(units(15:)),jd,secs)
155 |  end if
156 |
157 |  if (jd.ne. julianday.or. secs.ne. secondsofday) then
158 |    FATAL 'start time given in namelist does not match time'
159 |    FATAL 'read from restart file'
160 |    FATAL 'from namelist: ',julianday,secondsofday
161 |    call write_time_string(julianday,secondsofday,timestr_out)
162 |    LEVEL3 trim(timestr_out)
163 |    FATAL 'from hotstart: ',jd,secs
164 |    call write_time_string(jd,secs,timestr_out)
165 |    LEVEL3 trim(timestr_out)
166 |    stop 'restart_file()'
167 |  else
168 |    call write_time_string(jd,secs,timestr_out)
169 |    LEVEL2 'restart time ',trim(timestr_out)
170 |  end if
171 |  return
172 |
173 |  end subroutine check_restart_time
174 |!EOC
175 |
176 |-----
177 |!BOP
178 |!
179 |!IROUTINE:
180 |!
181 |!INTERFACE:
182 |  subroutine read_restart_data(var_name,allow_missing_variable,data_0d,data_1d)
183 |!
184 |!DESCRIPTION:
185 |!
186 |!INPUT PARAMETERS:
187 |  character(len=*), intent(in)          :: var_name
188 |  logical                               :: allow_missing_variable
189 |  REALTYPE, optional                    :: data_0d
190 |  REALTYPE, optional                    :: data_1d(:)
191 |!
192 |!REVISION HISTORY:
193 |!  Original author(s): Karsten Bolding and Jorn Bruggeman
194 |!
195 |!LOCAL VARIABLES:
196 |  integer          :: ierr, id

```

```

197| integer :: start(4), edges(4)
198| !EOP
199| !-----
200| !BOC
201| ierr = nf90_inq_varid(ncid, trim(var_name), id)
202| if (ierr /= NF90_NOERR) then
203|   if (.not. allow_missing_variable) then
204|     call handle_err(ierr,var_name)
205|   else
206|     LEVEL2 'variable ',trim(var_name),' not found - continuing'
207|     return
208|   end if
209| end if
210|
211| if (present(data_0d)) then
212|   ierr = nf90_get_var(ncid, id, data_0d)
213|   if (ierr /= NF90_NOERR) then
214|     call handle_err(ierr)
215|   end if
216| end if
217|
218| if (present(data_1d)) then
219|   start = 1 ; edges = 1; edges(3) = size(data_1d)
220|   ierr = nf90_get_var(ncid,id,data_1d,start,edges)
221|   if (ierr /= NF90_NOERR) then
222|     call handle_err(ierr)
223|   end if
224| end if
225|
226| end subroutine read_restart_data
227| !EOC
228|
229| subroutine handle_err(ierr,msg)
230| integer, intent(in) :: ierr
231| character(len=*), optional :: msg
232| LEVEL2 'read_restart_data(): error'
233| if (present(msg)) then
234|   LEVEL3 trim(nf90_strerror(ierr)),' - ',trim(msg)
235| else
236|   LEVEL3 trim(nf90_strerror(ierr))
237| end if
238| stop
239| end subroutine handle_err
240|
241| !-----
242| #endif
243| end module input_netcdf
244|
245| !-----
246| ! Copyright by the GOTM-team under the GNU Public License - www.gnu.org
247| !-----
248|

```