

```

1 /* catalog creates summary HTML files and thumbnail JPG files for coverages
2 /* in a selected directory
3 /* 1) Creates list of coverages
4 /* 2) For each coverage
5 /*
6 /* +-Creates HTML file with +- summary of "source" metadata information
7 /* | +- summary of attributes
8 /* | +- ref to "postage stamp" map
9 /* | +- Arc/Info "DESCRIBE" summary
10 /*
11 /* +-Creates JPG file with "postage stamp" map
12 /*
13 /* Michael Silberbauer May 1995 (cf. imager.aml)
14 /* " " Nov 1995 - sorted out a few emergent pupae.
15 /* " " Aug 1996 - checked the .adf extension used by Arc 7
16 /* " " Aug 1997 - added substantial HTML capabilities
17 /* " " Sep 1998 - rooted out unused features
18 /* " " Dec 1999 - creates catalogue files in new directory
19 /* " " Mar 2000 - creates JPG thumbnails with CONVERTIMAGE
20 /* " " Jun 2001 - fixed DATE field bug from Arc/Info 7.2 -> 8.0 transition
21 /*-----
22
23 &args dir Graphics LastCov
24 &type -----
25 &type Catalog summarises the metadata for a single coverage
26 &type or all the coverages in a particular directory.
27 &type Michael Silberbauer May 1995...
28 &type -----
29 &type Usage catalog dir {Graphics} {LastCov}
30 &type Usage catalog cover {Graphics} {LastCov}
31 &type -----
32
33 &type e.g. catalog /hri/db/cover/s-africa {G} (creates JPG thumbnail map for each cover)
34 &type e.g. catalog /hri/db/cover/s-africa {C} (creates JPG thumbnail map for each cover if none already exists)
35 &type e.g. catalog /hri/db/cover/s-africa {N} (creates only HTML file for each cover, regardless of JPG files)
36 &type LastCov is only used for debugging, to limit the number of covers processed.
37 &type -----
38 &type For a simpler catalogue, which does not require an Arc/Info licence,
39 &type but without graphics and attribute information, try:
40 &type cd <the directory>
41 &type pr -f -w80 */source* | lp -dhp4si
42 &type -----
43
44 /* Housekeeping: identify user, set file names
45 &sv User [translate [substr [username] 1 1]][substr [username] 2 [calc [length [username]] - 1]]
46 &type Closing all open files:
47 &sv close = [close -all]
48
49 &sv html_dir = /hri/db/cover/catalog
50 &sv tempcatfile = %html_dir%/catlis.htm
51 &sv prov = /hri/db/cover/s-africa/spr_500
52
53 /* date in ISO format
54 &sv ISOdate = [date -year]-[substr[date -usa] 1 2]-[substr[date -vmsdate] 1 2] - [before[date -vmstime] .]
55
56 /* Create the main HTML index file:
57 &sv TxtF = [open %tempcatfile% OpenStatus -write]
58 &sv wtf [write %TxtF% [quote <HTML>]]
59 &sv wtf [write %TxtF% [quote <BODY BGCOLOR = FFFFFFFF>]]
60 &sv wtf [write %TxtF% [quote <TITLE>IWQS GIS coverage catalogue</TITLE>]]
61 &sv wtf [write %TxtF% [quote <H1 ALIGN = CENTER>Geographic Information Systems Coverages at the Institute for Water Quality Studies</H1>]]
62
63 &sv wtf [write %TxtF% [quote <H3 ALIGN = CENTER>Created by user %User% with catalog.aml on %ISOdate%</H3> ]]
64 &sv wtf [write %TxtF% [quote <HR>]]

```

```

65 &sv wtf [write %TxtF% [quote <BR>Some form of copyright exists for most of these coverages, and we only intend them for use within the Department of Water Affairs and Forestry.]]
66 &sv wtf [write %TxtF% [quote <HR>]]
67 &sv wtf [write %TxtF% [quote <BR>Please click on an underlined file name to list its metadata.]]
68
69 &if [null %dir%] | [quote %dir%] = # | [quote %dir%] = . &then
70   &sv dir = [show workspace]
71
72 &if [null %Graphics%] &then
73   &sv Graphics = C
74 &else
75   &do
76     &if [locase [substr [quote %Graphics%] 1 1]] = 'g' &then &sv Graphics = G
77     &if [locase [substr [quote %Graphics%] 1 1]] = 'c' &then &sv Graphics = C
78     &if %Graphics% <> G and %Graphics% <> C           &then &sv Graphics N
79   &end
80
81 /* Find out whether this is a single coverage, or a directory.
82 /* If a directory, how many coverages exist in the directory?
83 &if [exists %dir% -cover] &then
84 &do
85   &type Processing the single coverage %dir%:
86   &sv Single = .true.
87   &sv nCovers = 1
88   &sv LastCov = 1
89 &end
90 &else
91 &do
92   &type Now making catalogue of data in %dir%
93   &sv Single = .false.
94   &if [exists tempcatlist -file] &then &sys \rm tempcatlist
95   &sv nCovers = [filelist %dir% tempcatlist -FULL -cover -all]
96   &sv CovF = [open tempcatlist OpenStatus -read]
97
98 /* Generate HTML table of coverage names with two coverage names per line:
99 &sv wtf [write %TxtF% [quote <TABLE BORDER>]]
100 &sv StartLine = .true.
101 &do nCov = 1 &to %nCovers%
102   &sv cover = [read %CovF% ReadStatus]
103   &if %StartLine% &then &sv wtf [write %TxtF% [quote <TR>]]
104   &sv wtf ~
105   [write %TxtF% ~
106   [quote <TD>%nCov%</TD><TD>%dir%</A HREF = [entryname %cover%].htm[entryname %cover%]</A></TD>]]
107   &if %StartLine% &then &sv StartLine = .false.
108   &else
109     &do
110       &sv wtf [write %TxtF% [quote </TR>]]
111       &sv StartLine = .true.
112     &end
113   &end
114 &if ^ %StartLine% &then
115 &do
116   &sv wtf [write %TxtF% [quote </TR>]]
117   &sv StartLine = .true.
118 &end
119 &sv wtf [write %TxtF% [quote </TABLE>]]
120
121 &sv CloseCovF = [close %CovF%]
122 &sv CovF = [open tempcatlist OpenStatus -read]
123 &type %nCovers% coverages found in %dir%
124
125 &if [null %LastCov%] &then &sv LastCov = %nCovers%
126 &else
127   &if %LastCov% > %nCovers% &then &sv LastCov = %nCovers%
128

```

```

129     &sv nCovSel = ( %LastCov% )
130
131     &if %nCovers% < 1 &then &stop No coverages found in %dir%.
132 &end
133
134 /* Type confirmation details:
135 &type Date:      %ISodate%
136 &type AML:       %AML$FULLFILE%
137 &type Username   [username]
138 &type Directory: %dir%
139 &type nCovSel:   %nCovSel%
140 &type LastCov:   %LastCov%
141 &type tempcatfile: %tempcatfile%
142 &type Graphics:  %Graphics%
143 &if [exists /prjws8/users/michael/script/whatami -file] &then
144     &sys /prjws8/users/michael/script/whatami
145
146 &do nCov = 1 &to %LastCov%
147
148     &if %Single% &then
149         &sv cover = %dir%
150     &else
151         &sv cover = [read %CovF% ReadStatus]
152
153     &sv covernd = [calc [length %cover%] - 1]
154     &sv coverfile = %cover%
155
156     /* Store standard Arc/Info data about the coverage in a file for later use:
157     &if [exists describe.tmp -file] &then &sv deldesc [delete describe.tmp -file]
158     listoutput describe.tmp
159     describe %cover%
160     listoutput screen
161
162     &sv source_html %html_dir%/[entryname %cover%].htm
163     &sv SorF [open %source_html% OpenStatus -write]
164     &type HTML file = %source_html% with open status %OpenStatus%
165     &if [exists tempdesc.tmp -file] &then &sv ddes [delete tempdesc.tmp -file]
166     &sv wtf [write %SorF% [quote <HTML>]]
167     &sv wtf [write %SorF% [quote <TITLE>]]
168     &sv wtf [write %SorF% [quote IWQS GIS coverage [entryname %cover%]]]
169     &sv wtf [write %SorF% [quote </TITLE>]]
170     &sv wtf [write %SorF% [quote <BODY BGCOLOR = FFFEE>]]
171     &sv wtf [write %SorF% [quote <HR> ]]
172     &if [exists %cover%/source -file] &then
173     &do
174         &sv Close_SorF [close %SorF%]
175         &sys echo '<H2 ALIGN = CENTER>' >> %source_html%
176         &sys nawk -f /prjws8/users/michael/data/sour2desc.awk %cover%/source >> %source_html%
177         &sys echo '</H2>' >> %source_html%
178         &sv SorF [open %source_html% OpenStatus -append]
179     &end
180     &sv wtf [write %SorF% [quote <HR> ]]
181
182     &call DateStampDir
183
184     &sv cov [substr [entryname %cover%] 1 3]
185     &sv er [substr [entryname %cover%] 4 [calc [length [entryname %cover%]] - 3]]
186     &sv wtf [write %SorF% [quote <H2>(%nCov%) %dir%/<A HREF = gis3code.htm#%cov%>%cov%</A>%er%</H2>]]
187     &type %dir%
188
189     &sv wtf [write %SorF% [quote <B>Institute for Water Quality Studies metadata.</B><BR> ]
190     &sv ISodate = [date -year]-[substr[date -usa] 1 2]-[substr[date -vmsdate] 1 2] - [before[date -vmstime] .]
191     &sv wtf [write %SorF% [quote Summary created by user %User% with catalog.aml on %ISodate%]]
192

```

```

193 &type File owner: %owner%, datestamp: %date%
194 &sv wtf [write %SorF% [quote <BR>File owner: %owner%, last modified: %date%]]
195
196 ArcPlot
197
198 display 1040
199 tempcatalog
200 &type -tempcatalog
201
202 mapextent %cover%
203 shadeset color
204 pageunits cm
205 textset font
206 textsymbol 1
207
208 &call MapFrame
209
210 &if ^ [exists %cover%/source] and ^ [exists %cover%/source.adf] &then
211 &do
212 &type No source file for %cover%
213 &sv wtf [write %SorF% '<EM>No metadata file (<TT>source</TT>) available.</EM>']
214 &if [exists sources/[entryname %cover%] -file] &then
215 &do
216 &type Retrieving sources/[entryname %cover%] into %cover%/source:
217 &sys cp sources/[entryname %cover%] %cover%/source
218 &end
219 &end
220 &sv ctf [close %SorF%]
221
222 &sys echo '<P><B>Projection information (see end of this page for details)</B><BR>' >> %source_html%
223 &sys grep -i projection describe.tmp >> %source_html%
224 &sys echo '<BR>' >> %source_html%
225 &sys grep -i spheroid describe.tmp >> %source_html%
226 &sys echo '<BR>' >> %source_html%
227
228 &if [exists %cover%/source] &then
229 &sys nawk -f /prjws8/users/michael/data/sour2htm.awk %cover%/source >> %source_html%
230
231 &if [exists %cover%/source.adf] &then
232 &sys nawk -f /prjws8/users/michael/data/sour2htm.awk %cover%/source.adf >> %source_html%
233
234 &sv SorF = [open %source_html% OpenStatus -append]
235 &sv wtf [write %SorF% [quote </PRE> ]]
236 &sv wtf [write %SorF% [quote <HR> ]]
237
238 &echo &on /*Debug Bailing out of ARCPLLOT
239 &call FeatureTypeTable
240 quit
241 &echo &off /*Debug Bailing out of ARCPLLOT
242 &call ThumbNailMap
243
244 &sv ctf [close %SorF%]
245 &sys echo '<HR>' >> %source_html%
246 &sys echo '<BR><B>Map specifications:</B>' >> %source_html%
247 &sys echo '<PRE>' >> %source_html%
248 &sys cat describe.tmp >> %source_html%
249 &sys echo '</PRE>' >> %source_html%
250 &sys echo '</BODY> </HTML>' >> %source_html%
251 &end
252
253 &sv wtf [write %TxtF% [quote </BODY> </HTML>]]
254 &sv cTxtF [close %TxtF%]
255 &sv ISOdate = [date -year]-[substr[date -usa] 1 2]-[substr[date -vmsdate] 1 2] - [before[date -vmstime] .]
256 &type Completed %AML$FULLFILE% at %ISOdate%

```

```

257 &if [exists tempcatlist -file] &then &sys \rm tempcatlist
258 &if [exists tempcatalog.gra -file] &then &sys \rm tempcatalog.gra
259 &return
260
261 /*-----
262 &routine SizePage
263
264 &sv page_x = [calc [extract 3 [show mapextent]] - [extract 1 [show mapextent]] ]
265 &sv page_y = [calc [extract 4 [show mapextent]] - [extract 2 [show mapextent]] ]
266 &sv page_max = [max %page_x% %page_y%]
267
268 &sv xp = [calc 100 * ( %page_x% / %page_max% ) ]
269 &sv yp = [calc 100 * ( %page_y% / %page_max% ) ]
270
271 &sv xm = 0.1
272 &sv ym = 0.1
273
274 &return
275
276
277 /*-----
278 &routine DateStampFile
279
280 &if [exists tempdirlist -file] &then &sys \rm tempdirlist
281 &if [exists %coverfile% -file] &then
282 &do
283   &sys \ls -l %coverfile% > tempdirlist
284   &sys \cat tempdirlist
285   &sv CovI = [open tempdirlist OpenStatus -read]
286   &sv di = [unquote [read %CovI% ReadStatus]]
287   &sv close = [close %CovI%]
288   &type %di%
289   &sv owner = [extract 3 %di%]
290   /*&sv size = [calc [extract 4 %di%] / 1024]
291   &sv size = [calc [extract 5 %di%] / 1024]
292   &if %size% < 1 &then &sv filsiz = 1k
293   &else &sv filsiz = [round %size%]k
294   /*&sv td = [extract 7 %di%]
295   &sv td = [extract 8 %di%]
296   &if [quote [substr %td% 3 1]] = ':' &then
297     /*&sv date = [extract 6 %di%] [extract 5 %di%] (less than 1yr ago)
298     &sv date = [extract 7 %di%] [extract 6 %di%] (less than 1yr ago)
299   &else
300     /*&sv date = [extract 6 %di%] [extract 5 %di%] [extract 7 %di%]
301     &sv date = [extract 7 %di%] [extract 6 %di%] [extract 8 %di%]
302 &end
303 &else
304 &do
305   &sv date = ?
306   &sv owner = ?
307   &sv filsiz = ?
308   &sv close = [close %CovI%]
309 &end
310
311 &return
312
313
314 /*-----
315 &routine DateStampDir
316
317 &if [exists tempdirlist -file] &then &sys \rm tempdirlist
318 &sys ls -ld %coverfile% > tempdirlist
319 &sys cat tempdirlist
320 &sv CovI = [open tempdirlist OpenStatus -read]

```

```

321 &if %OpenStatus% = 0 &then
322 &do
323 &sv di = [unquote [read %CovI% ReadStatus]]
324 &sv close = [close %CovI%]
325 &type %di%
326 &sv owner = [extract 3 %di%]
327 /*&sv td = [extract 7 %di%]
328 &sv td = [extract 8 %di%]
329 &if [quote [substr %td% 3 1]] = ':' &then
330 /*&sv date = [extract 6 %di%] [extract 5 %di%] (less than 1yr ago)
331 &sv date = [extract 7 %di%] [extract 6 %di%] (less than 1yr ago)
332 &else
333 /*&sv date = [extract 6 %di%] [extract 5 %di%] [extract 7 %di%]
334 &sv date = [extract 7 %di%] [extract 6 %di%] [extract 8 %di%]
335 &end
336 &else
337 &do
338 &sv date = ?
339 &sv owner = ?
340 &type File error %OpenStatus% on tempdirlist: date = %date%
341 &sv close = [close %CovI%]
342 &end
343
344 &return
345
346 /*-----
347 &routine FeatureTypeTable
348
349 /* Tabulate the feature types:
350 &do type &list Annotation Arc Point Polygon
351 &if [exists %cover% -%type%] &then
352 &do
353 &if %type% = Annotation &then &sv fil = txt
354 &if %type% = Arc &then &sv fil = aat
355 &if %type% = Point &then &sv fil = pat
356 &if %type% = Polygon &then &sv fil = pat
357 &sv coverfile = %cover%/fil%
358 &call DateStampFile
359
360 &if %type% = Annotation &then
361 &do
362 &sv wtf [write %SorF% [quote <BR>%type% items, %filsiz%byte:]]
363 &sv wtf [write %SorF% [quote <BR>%date%, %owner%:]]
364 &end
365 &else
366 &do
367
368 &sv wtf [write %SorF% [quote <BR>%type% items, %filsiz%byte:]]
369 &sv wtf [write %SorF% [quote <BR>%date%, %owner%:]]
370 &sv items [listitem %cover% -%type% -all -all tempitem.lis]
371 &sys cat tempitem.lis
372 &if %items% > 1 &then &type checking %items% INFO items
373 &sv wtf [write %SorF% [quote <BR>%items% data fields]]
374 &sv InfF = [open tempitem.lis OpenStatus -read]
375
376 &sv wtf [write %SorF% [quote <TABLE BORDER>]]
377 &sv wtf [write %SorF% [quote <TR><TH>ITEM</TH><TH>EXAMPLES</TH><TH>WIDTH</TH><TH>TYPE</TH><TH>REDEFINED?</TH><TH>INDEXED?</TH></TR>]]
378
379 &if %items% > 0 &then
380 &do ItIn = 1 &to %items%
381
382 &sv itemName = [read %InfF% ReadStatus]
383 &sv itemDefn = [iteminfo %cover% -%type% %itemName% -definition]
384

```

```

385     &if [iteminfo %cover% -%type% %itemName% -redefined] &then &sv itemRedf = (ReDefined)
386     &else &sv itemRedf = [unquote '']
387     &if [iteminfo %cover% -%type% %itemName% -indexed] &then &sv itemIndx = (Indexed)
388     &else &sv itemIndx = [unquote '']
389
390     &sv fieldwidth = [extract 1 %itemDefn%]
391     &sv fieldtype = [extract 3 %itemDefn%]
392     &if %fieldtype% = D &then &sv fieldtype = date
393     &if %fieldtype% = B &then &sv fieldtype = binary
394     &if %fieldtype% = C &then &sv fieldtype = character
395     &if %fieldtype% = F &then &sv fieldtype = floatpnt
396     &if %fieldtype% = I &then &sv fieldtype = integer
397     &if %fieldtype% = N &then &sv fieldtype = numeric
398     &sv fielddecs = [extract 4 %itemDefn%]
399
400     &if %fieldtype% = character &then
401     &do
402         &sv eg = [unquote '-']
403         clearselect %cover% %type%
404         reselect %cover% %type% %itemName% ne ''
405         &sv nSelItm = [extract 1 [show select %cover% %type%]]
406         &if %nSelItm% > 0 &then
407         &do
408             &sv eg = [quote [show select %cover% %type% 1 item %itemName%]]
409             &sv eg = [quote [subst %eg% '[' '(' ]]
410             &sv eg = [subst %eg% ']' ')']
411             &sv eg = [unquote %eg% ]
412         &end
413         &sv wtf [write %SorF% [quote <TR><TD>%itemName%</TD><TD>%eg%</TD><TD>%fieldwidth%</TD> <TD>%fieldtype%</TD> <TD>%itemRedf%</TD> <TD>%itemIndx%</TD></TR>]]
414     &end
415     &else
416     &do
417         &sv zero = 0
418         &if %fieldtype% = date &then &sv zero = 1801-01-01
419         &sv eg = 0
420         clearselect %cover% %type%
421         reselect %cover% %type% %itemName% > %zero%
422         &sv nSelItm = [extract 1 [show select %cover% %type%]]
423         &if %nSelItm% > 0 &then
424         &do
425             &sv eg = [quote [show select %cover% %type% 1 item %itemName%]]
426             &sv eg = [quote [subst %eg% '[' '(' ]]
427             &sv eg = [subst %eg% ']' ')']
428             &sv eg = [unquote %eg% ]
429         &end
430         &sv wtf [write %SorF% [quote <TR><TD>%itemName%</TD><TD>%eg%</TD><TD>%fieldwidth%.%fielddecs%</TD> <TD>%fieldtype%</TD> <TD>%itemRedf%</TD> <TD>%itemIndx%</TD></TR>]]
431     &end
432
433     &end
434     &sv closeinf = [close %InfF%]
435     &sv wtf [write %SorF% [quote </TABLE>]]
436     &sv wtf [write %SorF% [quote <HR>]]
437 &end
438 &end
439 &end
440
441 &return
442
443 /*-----
444 &routine ThumbNailMap
445
446 &sv cov_jpg = %html_dir%/[entryname %cover%].jpg
447 &sv cov_jgw = %html_dir%/[entryname %cover%].jgw
448 &if %Graphics% = G and [exists %cov_jpg% -file] &then &sv djpg [delete %cov_jpg% -file]

```

```

449
450 &if %Graphics% = N or ( %Graphics% = C and [exists %cov_jpg% -file] ) &then
451     &type Not creating graphic
452 &else
453 &do
454     ArcPlot
455
456     display 1040
457     tempcatalog
458     &type -tempcatalog
459
460     mapextent %cover%
461     shadeset color
462     textset font
463     textsymbol 1
464     markerset mineral
465     mapposition cen cen
466     mapextent %cover%
467     clipmapextent off
468
469     &sv page_x = [calc [extract 3 [show mapextent]] - [extract 1 [show mapextent]] ]
470     &sv page_y = [calc [extract 4 [show mapextent]] - [extract 2 [show mapextent]] ]
471     &sv page_max = [max %page_x% %page_y%]
472     &sv xp = [calc 10 * ( %page_x% / %page_max% ) ]
473     &sv yp = [calc 10 * ( %page_y% / %page_max% ) ]
474
475     &sv disp_max = [max %xp% %yp%]
476     &sv disp_scale = [calc 256 / %disp_max%]
477     &sv disp_x = [round [calc %disp_scale% * %xp%] ]
478     &sv disp_y = [round [calc %disp_scale% * %yp%] ]
479
480     display 9999 size %disp_x% %disp_y%
481     pagesize %xp% %yp%
482     mapposition cen cen
483     mapextent %cover%
484     clipmapextent off
485
486     &type maplimits 0 0 %xp% %yp%
487     maplimits 0 0 %xp% %yp%
488
489     shadesymbol 14
490     shadecolor gray90
491     shadeput 14
492     linesymbol 1
493     box [show maplimits]
494     /*patch [show maplimits]
495
496     &if [exists %cover%/prj] or [exists %cover%/prj.adf] &then
497     &do
498         mapprojection AUTO %cover%
499         clearselect %prov% poly
500         reselect %prov% poly name = 'SEA'
501         polygonshade %prov% 5
502     &end
503
504     clearselect
505     &severity &error &ignore
506     &if [exists %cover% -arc] or [exists %cover% -polygon] or ~
507         [exists %cover%/arc -file] or [exists %cover%/arc.adf -file] ~
508     &then
509         arcs %cover%
510     &else
511         pointmarkers %cover% 323
512     &severity &error &fail

```



```

513 &if [exists tempras -file] &then &sv dt [delete tempras -file]
514 &if [exists temprasw -file] &then &sv dt [delete temprasw -file]
515 screensave tempras
516 quit
517 &type Converting graphics file %nCov% into JPG...
518 &if [exists %cov_jpg% -file] &then &sv deljpg [delete %cov_jpg% -file]
519 &if [exists %cov_jgw% -file] &then &sv deljgw [delete %cov_jgw% -file]
520 convertimage tempras %cov_jpg% jfif
521
522 &end
523
524 &return
525
526 /*-----
527 &routine JPGpage
528
529 arcplot
530 &sv disp_max = [max %xp% %yp%]
531 &sv disp_scale = [calc 256 / %disp_max%]
532 &sv disp_x = [round [calc %disp_scale% * %xp%] ]
533 &sv disp_y = [round [calc %disp_scale% * %yp%] ]
534 display 9999 size %disp_x% %disp_y%
535 pageunits cm
536 pagesize %xp% %yp%
537 plot tempcatalog box 0 0 %xp% %yp%
538 quit
539
540 &return
541
542 /*-----
543 &routine MapFrame
544
545 /* Set up a neat frame for the thumbnail map:
546
547 mapextent %cover%
548 &sv xml = [extract 1 [show mapextent] ,]
549 &sv yml = [extract 2 [show mapextent] ,]
550 &sv xm2 = [extract 3 [show mapextent] ,]
551 &sv ym2 = [extract 4 [show mapextent] ,]
552
553 /* Round off coordinates to nearest metre, or nearest 1/1000 of a degree:
554 &if [abs %xml%] < 999 &then
555 &sv xml [calc [round [calc 1000 * %xml%]] / 1000]
556 &else
557 &sv xml [round %xml% ]
558 &if [abs %ym1%] < 999 &then
559 &sv yml [calc [round [calc 1000 * %ym1%]] / 1000]
560 &else
561 &sv yml [round %ym1% ]
562 &if [abs %xm2%] < 999 &then
563 &sv xm2 [calc [round [calc 1000 * %xm2%]] / 1000]
564 &else
565 &sv xm2 [round %xm2% ]
566 &if [abs %ym2%] < 999 &then
567 &sv ym2 [calc [round [calc 1000 * %ym2%]] / 1000]
568 &else
569 &sv ym2 [round %ym2% ]
570
571 &sv %xml% = [unquote '' ]
572 &sv %xm2% = [unquote '' ]
573 &sv %ym1% = [unquote '' ]
574 &sv %ym2% = [unquote '' ]
575
576 &if %xml% < 0 &then &sv xmls W

```

```

577 &if %xml% > 0 &then &sv xmls E
578 &if %xm2% < 0 &then &sv xm2s W
579 &if %xm2% > 0 &then &sv xm2s E
580 &if %ym1% < 0 &then &sv ym1s S
581 &if %ym1% > 0 &then &sv ym1s N
582 &if %ym2% < 0 &then &sv ym2s S
583 &if %ym2% > 0 &then &sv ym2s N
584 clipmapextent off
585
586 &type (%nCov%) %cover% map extents:
587 &type %ym2%
588 &type '| '
589 &type '| '
590 &type '| '
591 &type %ym1%
592 &type . %xml% ----- %xm2%
593
594 &sv wtf [write %SorF% [quote <BR><B>Map extent:</B>]]
595 &sv wtf [write %SorF% [quote <TABLE BORDER>]]
596 &sv wtf [write %SorF% [quote <TR><TD></TD><TD>%xml%%xmls%</TD><TD>.</TD><TD>%xm2%%xm2s%</TD></TR>]]
597 &sv wtf [write %SorF% [quote <TR><TD VALIGN=TOP>%ym2%%ym2s%</TD><TD>-</TD><TD><IMG SRC=[entryname %cover%].jpg></TD><TD>-</TD><TD VALIGN=TOP>%ym2%%ym2s%</TD></TR>]]
598 &sv wtf [write %SorF% [quote <TR><TD>%ym1%%ym1s%</TD><TD>.</TD><TD>.</TD><TD>.</TD><TD>%ym1%%ym1s%</TD></TR>]]
599 &sv wtf [write %SorF% [quote <TR><TD></TD><TD>%xml%%xmls%</TD><TD>.</TD><TD>%xm2%%xm2s%</TD></TR>]]
600 &sv wtf [write %SorF% [quote </TABLE>]]
601
602 &sv ctf [close %SorF%]
603
604 &return
605
606
607

```