```
1       * ACTION.S -- Action Procedures for Small Runoff; Dec 6, 1981
2       BEGIN ACTION;
3       *--------------
4         ENT BRK,PUTOUT,SPACE;
5         ENT GETVAL,SETVAL,LEADBL,UNDERL;
6         ENT INSERT,DELETE,SPREAD;
7         ENT GETWRD,GOTWRD;
8         ENT NADA,PLUS,MINUS;
9         ENT POS,NEG;
10      *--------------
11        EXT TRUE,FALSE,NULL;
12        EXT OUTFILE;
13        EXT INBUF,OUTBUF;
14        EXT PSEUDO;
15        EXT M1VAL,M2VAL,M3VAL,M4VAL;
16        EXT FILLVAL,CURPAG,NEWPAG;
17        EXT LINENO,TIVAL,LSVAL,INVAL;
18        EXT BOTTOM,OUTP,OUTW;
19        EXT HEAD,FOOT;
20        EXT DIRRT,OUTWRDS,NEXTRA,MLINEW;
21        EXT WRDBUF,WRDLEN;
22      *--------------
23        EXT PROC WRITE,PROC STREQ,PROC CAT2;
24        EXT PROC PWRITE,PROC QWRITE,PROC WRITELN;
25      *--------------
26        DCL IZ1;
27        DCL PLUS='+',MINUS='-',BLANK=' ';
28        MSG BLINE=' ';                          * Blank line
29      *--------------
30      PROC SKIPLINES(NZ1);                      * Skip Lines
31  1     IZ1=1;
32  1     DO WHILE IZ1 LE NZ1;
33  2       CALL WRITE(OUTFILE,BLINE);
34  2       IZ1=IZ1+1;
35  2       ENDDO
36  1     RETURN
37  1     ENDPROC
38  0   *--------------
39  0     DCL LX1;                                * Line Length
40  0     DCL CZ1;                                * Temp char buffer
41  0   *--------------
42  0   PROC PUTTL(SZ3);                          * Output Title with Page Number
43  1     LX1=SZ3;                                * Length of arg string
44  1     IZ1=1;
45  1     DO WHILE IZ1 LE LX1;
46  2       CZ1=SZ3(IZ1);
47  2       IF CZ1 EQ PSEUDO;
48  3         THEN CALL QWRITE(OUTFILE,CURPAG);
49  3         ELSE CALL PWRITE(OUTFILE,CZ1);
50  3         ENDIF
51  2       IZ1=IZ1+1;
52  2       ENDDO
53  1     CALL WRITELN(OUTFILE);
54  1     RETURN
55  1     ENDPROC
56  0   *--------------
57  0     DCL IZ2;
58  0   *--------------
59  0   PROC PHEAD;                       * Write header at top of page
60  1     CURPAG=NEWPAG;
```

```
61   1    NEWPAG=NEWPAG+1;
62   1    IF M1VAL GT 0;
63   2      THEN
64   2        IZ2=M1VAL-1;
65   2        CALL SKIPLINES(IZ2);
66   2        CALL PUTTL(HEAD);
67   2      ENDIF
68   1    CALL SKIPLINES(M2VAL);
69   1    LINENO=M1VAL+M2VAL;
70   1    RETURN
71   1    ENDPROC
72   0  *---------------
73   0  PROC PFOOT;                        * Write footer at bottom of page
74   1    CALL SKIPLINES(M3VAL);
75   1    IF M4VAL GT 0;
76   2      THEN
77   2        CALL PUTTL(FOOT);
78   2        IZ2=M4VAL-1;
79   2        CALL SKIPLINES(IZ2);
80   2      ENDIF
81   1    LINENO=0;
82   1    RETURN
83   1    ENDPROC
84   0  *---------------
85   0    DCL IZ3;
86   0    DCL OUTP1;
87   0  *---------------
88   0  PROC PUTOUT;                      * Write out the currant line
89   1    IF LINENO EQ 0;
90   2      THEN CALL PHEAD;
91   2      ELSE
92   2        IF LINENO GE BOTTOM;
93   3          THEN CALL PHEAD;
94   3          ENDIF
95   2      ENDIF
96   1    IZ3=1;
97   1    DO WHILE IZ3 LE TIVAL;
98   2      CALL PWRITE(OUTFILE,BLANK);
99   2      IZ3=IZ3+1;
100  2      ENDDO
101  1    TIVAL=INVAL;
102  1    OUTP1=OUTP-1;
103  1    IZ3=1;
104  1    DO WHILE IZ3 LE OUTP1;
105  2      CZ1=OUTBUF(IZ3);
106  2      IF CZ1 EQ PSEUDO;
107  3        THEN CALL PWRITE(OUTFILE,BLANK);
108  3        ELSE CALL PWRITE(OUTFILE,CZ1);
109  3        ENDIF
110  2      IZ3=IZ3+1;
111  2      ENDDO
112  1    CALL WRITELN(OUTFILE);
113  1    LINENO=LINENO+1;
114  1    OUTP=0;
115  1    OUTW=0;
116  1    IF LINENO GT (BOTTOM-LSVAL);
117  2      THEN IZ3=BOTTOM-LINENO;
118  2      ELSE IZ3=LSVAL-1;
119  2      ENDIF
120  1    CALL SKIPLINES(IZ3);
```

```
121  1    LINENO=LINENO+LSVAL-1;
122  1    IF LINENO GE BOTTOM;
123  2      THEN CALL PFOOT;
124  2      ENDIF
125  1    RETURN
126  1    ENDPROC
127  0  *--------------
128  0    DCL IZ4;
129  0  *--------------
130  0  PROC BRK;                          * Write out current line using PUTOUT
131  1    IF OUTP GT 0;
132  2      THEN CALL PUTOUT;
133  2      ENDIF
134  1    OUTP=0;
135  1    OUTW=0;
136  1    OUTWRDS=0;
137  1    RETURN
138  1    ENDPROC
139  0  *--------------
140  0    DCL IZ5,OLDLSVAL;
141  0  *--------------
142  0  PROC SPACE(NZ2);                   * Write out NZ2 blank lines
143  1    CALL BRK;
144  1    IZ4=NZ2;
145  1    IF IZ4 GT BOTTOM-LINENO;
146  2      THEN IZ4=BOTTOM-LINENO;
147  2      ENDIF
148  1    OLDLSVAL=LSVAL;
149  1    LSVAL=1;
150  1    IZ5=1;
151  1    DO WHILE IZ5 LE IZ4;
152  2      CALL PUTOUT;
153  2      IZ5=IZ5+1;
154  2      ENDDO
155  1    LSVAL=OLDLSVAL;
156  1    RETURN
157  1    ENDPROC
158  0  *--------------
159  0  PROC INDIG(CH8);                   * True if arg is a digit
160  1    IF CH8 LT '0';
161  2      THEN RETURN FALSE;
162  2      ELSE
163  2        IF CH8 LE '9';
164  3          THEN RETURN TRUE;
165  3          ELSE RETURN FALSE;
166  3          ENDIF
167  2      ENDIF
168  1    ENDPROC
169  0  *--------------
170  0  PROC INTVAL(CH7);                  * True if arg is +, - or digit
171  1    IF CH7 EQ PLUS;
172  2      THEN RETURN TRUE;
173  2      ELSE
174  2        IF CH7 EQ MINUS;
175  3          THEN RETURN TRUE;
176  3          ELSE RETURN INDIG(CH7);
177  3          ENDIF
178  2      ENDIF
179  1    ENDPROC
180  0  *--------------
```

```
181  0    DCL CH;
182  0    DCL NADA,POS,NEG;
183  0    DCL IZ6,LX6,GETV;
184  0  *--------------
185  0  PROC GETVAL(GETVA);                  * Get value of parameter
186  1    NADA=FALSE;
187  1    POS=FALSE;
188  1    NEG=FALSE;
189  1    LX6=INBUF;                        * Length of INBUF
190  1    IZ6=1;
191  1    DO WHILE IZ6 LT LX6;              * Skip over non-blanks
192  2      IF INBUF(IZ6) EQ BLANK;
193  3        THEN EXIT;
194  3        ENDIF
195  2      IZ6=IZ6+1;
196  2      ENDDO
197  1    DO WHILE IZ6 LT LX6;              * Now skip over blanks
198  2      IF INBUF(IZ6) NE BLANK;
199  3        THEN EXIT;
200  3        ENDIF
201  2      IZ6=IZ6+1;
202  2      ENDDO
203  1    CH=INBUF(IZ6);
204  1    IF NOT INTVAL(CH);
205  2      THEN NADA=TRUE;
206  2      ELSE
207  2        IF CH EQ PLUS;
208  3          THEN POS=TRUE;
209  3          ELSE
210  3            IF CH EQ MINUS;
211  4              THEN NEG=TRUE;
212  4              ENDIF
213  3          ENDIF
214  2      ENDIF
215  1    IF POS OR NEG;
216  2      THEN IZ6=IZ6+1;
217  2      ENDIF
218  1    GETV=0;
219  1    DO WHILE IZ6 LE LX6;
220  2      CH=INBUF(IZ6);
221  2      IF INDIG(CH);
222  3        THEN GETV=10*GETV+CH-'0';
223  3        ELSE EXIT;
224  3        ENDIF
225  2      IZ6=IZ6+1;
226  2      ENDDO
227  1    GETVA=GETV;
228  1    RETURN
229  1    ENDPROC
230  0  *--------------
231  0  * Set the value of the command parameter within limits
232  0  PROC SETVAL(PARAM,VAL,MINVAL,MAXVAL);
233  1    IF POS;
234  2      THEN PARAM=PARAM+VAL;
235  2      ELSE
236  2        IF NEG;
237  3          THEN PARAM=PARAM-VAL;
238  3          ELSE PARAM=VAL;
239  3          ENDIF
240  2      ENDIF
```

```
241  1    IF PARAM GT MAXVAL;
242  2      THEN PARAM=MAXVAL;
243  2      ELSE
244  2        IF PARAM LT MINVAL;
245  3          THEN PARAM=MINVAL;
246  3          ENDIF
247  2      ENDIF
248  1    RETURN
249  1    ENDPROC
250  0  *---------------
251  0    DCL IY1,LY1;
252  0    DCL STILLBLANK;
253  0    DCL ONE=1;
254  0  *---------------
255  0  * Delete leading blanks from INBUF, set TIVAL
256  0  PROC LEADBL;
257  1    LY1=INBUF;                          * Length of INBUF
258  1    IF LY1 GT 0;
259  2      THEN
260  2        IY1=1;
261  2        STILLBLANK=TRUE;
262  2        DO WHILE IY1 LE LY1;
263  3          IF NOT STILLBLANK;
264  4            THEN EXIT;
265  4            ENDIF
266  3          IF INBUF(IY1) NE BLANK;
267  4            THEN STILLBLANK=FALSE;
268  4            ELSE IY1=IY1+1;
269  4            ENDIF
270  3          ENDDO
271  2        IY1=IY1-1;
272  2        IF IY1 NE LY1;
273  3          THEN
274  3            IF NOT FILLVAL;
275  4              THEN TIVAL=IY1+INVAL;
276  4              ENDIF
277  3          ENDIF
278  2        IF IY1 GT 0;
279  3          THEN CALL DELETE(INBUF,ONE,IY1);
280  3          ENDIF
281  2      ENDIF
282  1      RETURN
283  1      ENDPROC
284  0  *---------------
285  0    DCL LOWERA=97,LOWERZ=122;
286  0  *---------------
287  0  PROC INALPH(CH9);                     * Return true if CH9 is a letter
288  1    IF CH9 LT 'A';
289  2      THEN RETURN FALSE;
290  2      ELSE
291  2        IF CH9 LE 'Z';
292  3          THEN RETURN TRUE;
293  3          ELSE
294  3            IF CH9 LT LOWERA;
295  4              THEN RETURN FALSE;
296  4              ELSE
297  4                IF CH9 LE LOWERZ;
298  5                  THEN RETURN TRUE;
299  5                  ENDIF
300  4              ENDIF
```

```
301  3          ENDIF
302  2        ENDIF
303  1      RETURN FALSE;
304  1      ENDPROC
305  0    *---------------
306  0      DCL IY2,LY2;
307  0      DCL UNDERS=(2,8,95);          * Backspace, Underscore
308  0    *---------------
309  0    PROC UNDERL;
310  1      IY2=1;
311  1      DO WHILE IY2 LE INBUF;
312  2        CH=INBUF(IY2);
313  2        IF INALPH(CH);
314  3          THEN
315  3            IF IY2 EQ INBUF;
316  4              THEN CALL CAT2(INBUF,UNDERS);
317  4              ELSE
318  4                LY2=IY2+1;
319  4                CALL INSERT(UNDERS,INBUF,LY2);
320  4              ENDIF
321  3            IY2=IY2+3;
322  3          ELSE IY2=IY2+1;
323  3          ENDIF
324  2        ENDDO
325  1      RETURN
326  1      ENDPROC
327  0    *---------------
328  0      DCL IS2,IY3,LY3;
329  0    *---------------
330  0    PROC INSERT(NEWS,OLDS,ATS);
331  1      IS2=OLDS;
332  1      IY3=NEWS+OLDS;
333  1      OLDS=IY3;
334  1      DO WHILE IS2 GE ATS;
335  2        OLDS(IY3)=OLDS(IS2);
336  2        IY3=IY3-1;
337  2        IS2=IS2-1;
338  2        ENDDO
339  1      IS2=1;
340  1      IY3=ATS;
341  1      DO WHILE IS2 LE NEWS;
342  2        OLDS(IY3)=NEWS(IS2);
343  2        IY3=IY3+1;
344  2        IS2=IS2+1;
345  2        ENDDO
346  1      RETURN
347  1      ENDPROC
348  0    *---------------
349  0      DCL LD8,ID8,JD8;
350  0    *---------------
351  0    * Delete L characters from DSTR starting at I
352  0    PROC DELETE(DSTR,ID9,LD9);
353  1      ID8=ID9+LD9;
354  1      JD8=ID9;
355  1      LD8=DSTR;
356  1      DO WHILE JD8 LE LD8;
357  2        DSTR(JD8)=DSTR(ID8);
358  2        JD8=JD8+1;
359  2        ID8=ID8+1;
360  2        ENDDO
```

```
361  1    DSTR=DSTR-LD9;
362  1    RETURN
363  1    ENDPROC
364  0  *--------------
365  0    DCL IU1,JU1,LU1;
366  0    DCL FINWRD,GOTWRD;
367  0  *--------------
368  0  PROC GETWRD(IPOS);
369  1    LU1=INBUF;
370  1    GOTWRD=FALSE;
371  1    DO WHILE IPOS LE LU1;
372  2      IF GOTWRD;
373  3        THEN EXIT;
374  3        ELSE
375  3          IF INBUF(IPOS) EQ BLANK;
376  4            THEN IPOS=IPOS+1;
377  4            ELSE GOTWRD=TRUE;
378  4            ENDIF
379  3        ENDIF
380  2      ENDDO
381  1    JU1=IPOS;
382  1    IF GOTWRD;
383  2      THEN
384  2        FINWRD=FALSE;
385  2        DO WHILE NOT FINWRD;
386  3          IF JU1 GT LU1;
387  4            THEN FINWRD=TRUE;
388  4            ELSE
389  4              IF INBUF(JU1) EQ BLANK;
390  5                THEN FINWRD=TRUE;
391  5                ENDIF
392  4            ENDIF
393  3          JU1=JU1+1;
394  3          ENDDO
395  2        IU1=1;
396  2        LU1=JU1-IPOS-1;
397  2        DO WHILE IU1 LE LU1;
398  3          WRDBUF(IU1)=INBUF(IPOS+IU1-1);
399  3          IU1=IU1+1;
400  3          ENDDO
401  2        WRDLEN=LU1;
402  2        IPOS=JU1;
403  2      ENDIF
404  1    RETURN
405  1    ENDPROC
406  0  *--------------
407  0    DCL IV1,JV1,LV1;
408  0    DCL NE,NB,NHOLES;
409  0  *--------------
410  0  PROC SPREAD;                        * Justify a line
411  1    IF NEXTRA GT 0;
412  2      THEN
413  2        IF OUTWRDS GT 1;
414  3          THEN
415  3            DIRRT=NOT DIRRT;
416  3            NHOLES=OUTWRDS-1;
417  3            NE=NEXTRA;
418  3            IV1=OUTP-1;
419  3            JV1=IV1+NE;
420  3            IF JV1 GT MLINEW;
```

```
421   4                 THEN JV1=MLINEW;
422   4                 ENDIF
423   3               DO WHILE IV1 LT JV1;
424   4                 OUTBUF(JV1)=OUTBUF(IV1);
425   4                 IF OUTBUF(IV1) EQ BLANK;
426   5                   THEN
427   5                     IF DIRRT;
428   6                       THEN NB=NE/NHOLES;
429   6                       ELSE NB=(NE-1)/NHOLES+1;
430   6                       ENDIF
431   5                     NHOLES=NHOLES-1;
432   5                     NE=NE-NB;
433   5                     DO WHILE NB GT 0;
434   6                       NB=NB-1;
435   6                       JV1=JV1-1;
436   6                       OUTBUF(JV1)=BLANK;
437   6                       ENDDO
438   5                   ENDIF
439   4                 IV1=IV1-1;
440   4                 JV1=JV1-1;
441   4                 ENDDO
442   3             ENDIF
443   2         ENDIF
444   1       RETURN
445   1       ENDPROC
446   0     *---------------
447   0     END
 NO ERRORS DETECTED
```