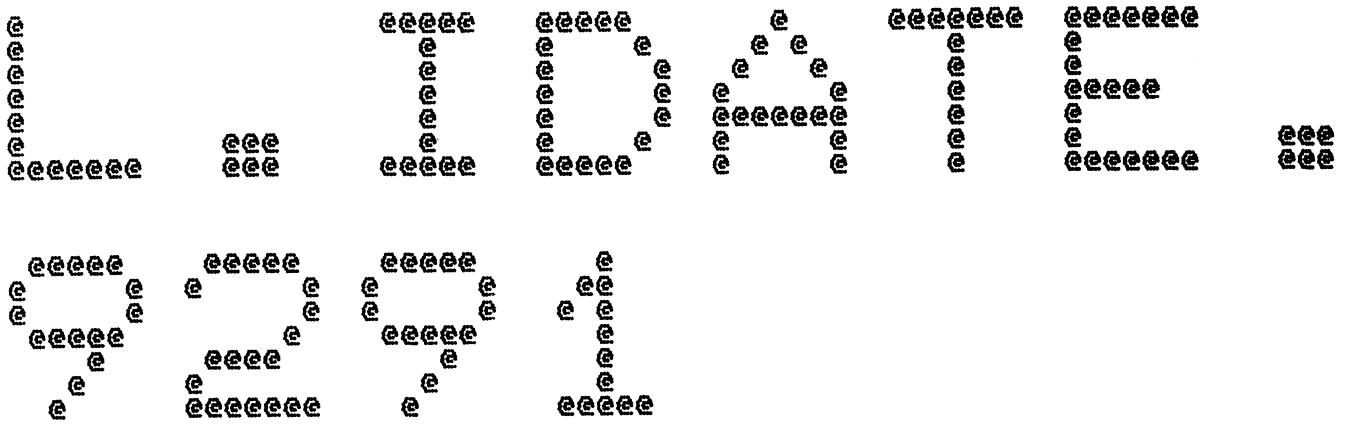
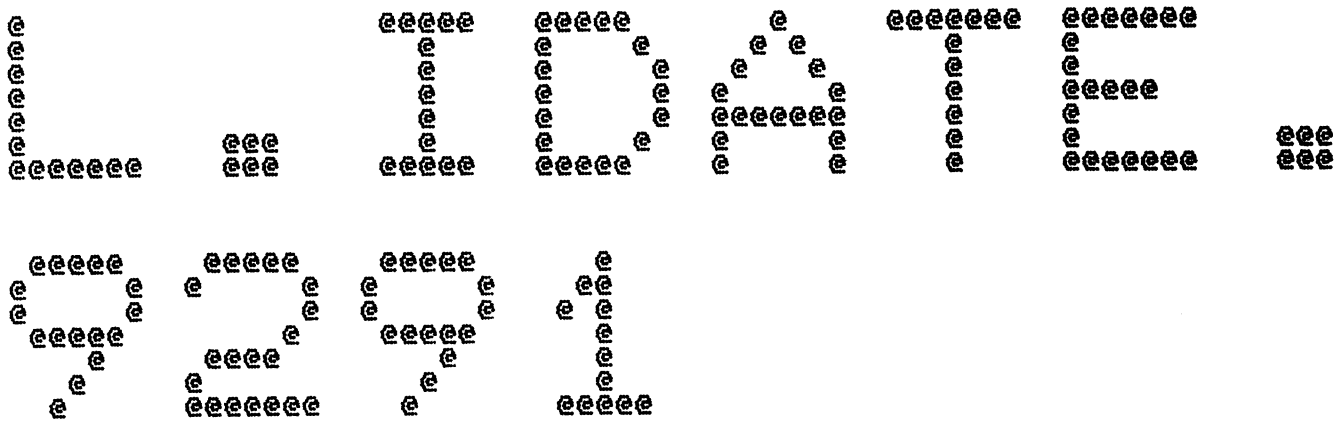


Spool Queue Line #: 41
IRIS LU/Filename : 18/L.IDATE.9291

Printed on/at : FEB 6, 1990 12:52:51
For Group/User: 0 , 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For R9.5 RELSE CN



Spool Queue Line #: 41
IRIS LU/Filename : 18/L.IDATE.9291

Printed on/at : FEB 6, 1990 12:52:58
For Group/User: 0 , 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For R9.5 RELSE CN

```
.EOT   ;"IDATE" (DISCSUBS GROUP 5) FOR R9.xx
.EOT
.EOT   ;"DSUBDEFS" FOR IRIS
.END
```

```
ASM 18/A.IDATE.9291!,@18/L.IDATE.9291!,B050,-B051,B052
FEB  6, 1990  10:05:01


;       Batchfile:   R95JCL.IDATE

;
;       ; D = 9291
;
;       -R95DEFSPZ
;       -R95DSUBDEFSD
;       R92IDATESA
;       *.END
;
                    .EOT   ;"IDATE" (DISCSUBS GROUP 5) FOR R9.xx
```

```
;                    << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>
;LAST EDITED BY RDC FOR R9.0
;       ORIGINAL ASSEMBLY DATE = 3319

;       DATE.S - Universal date conversion routine
;
;       This routine permits a numeric specification of the date and time
;       to be converted to a string of the required format.  The converse
;       function is also provided.
;
;       The following formats are supported:-
;            30-Dec-83, 30/12/83 and 12/30/83 for date and
;            3:55p and 15:55 for time.
;          On input of the date as a string the year may be entered as the
;          full four digits or only the last two.  Eg 2054, 1954 and 54
;          will all return the year number as 54.  Years before 1900 or after
;          199 will return values outside the range 0..99.
;
;       The routine has been designed to permit the required format to
;       be specified with a minimum of change.  Similarly the delimiters
;       within date, names of the months and 'am'/'pm' characters are
;       stored at fixed easily changed locations.
;
;       Calling sequence:-
;            CALL $DATE,mode,year,month,day,hour,minute,string$,error
;
;       Mode=0 - causes the revision of the routine to be returned in year.
;                No other parameters are required, altered or checked.
;       Mode=1 - converts the numeric version of the date and time to a
;                string in string$ according to the required format.
;       Mode=2 - converts the string$ parameter to the appropriate numeric
;                parameters.  In this mode the order of month and date is
;                presumed to be identical to that for the output conversion.
;                This ordering is over-ridden if the ssecond parameter is
;                alphameric.  The delimiters used within the date are not
;                checked and may be any non numeric character.  At least one
;                space is required between the date and time fields.
;                In mode 2 the error status is set to 0 if the string is
;                read and interpretted, -1 otherwise.  If status=-1 then the
;                first unread parameter is also set to -1.
;
;       The string parameter must be dimensioned to at least 16 characters.
;
;       The following locations may be set to specify the required format:-
;            4 - the address of the routine for the first field
;                   to output.  This is assembled as the day number
;            5 - the address of the second field output sequence,
;                   this is assembled as month name, assume numeric
;                   input in the form dd/mm/yy
```

```
;                    << SI = R92IDATESA; BO = 18/A. IDATE. 9291! >>
;
;          If it is required to output the month as a number, the address
;          of the required routine is assembled at 7.
;
;                6 - the characters for 'am' and 'pm' indication.  These
;                    are asembled as 'AP'
;               10 - in numeric input of date, accept mm/dd/yy = 1 or
;                                                   dd/mm/yy = 0
;               11 - the names of the months.  Three characters each
;                    separated by a separator.  This is assembled as
;                    'Jan-Feb...Dec-'.  The sequence is terminated
;                    by a zero word at 41.
;               42 - the hours in the clock.  14 or 30.
;               43 - the separator to place between months, days and years.
;
;
;       The following locations are included for completeness:-
;
;                7 - the routine to output the month as a number
;
;Revision History:
;
;     5   23-Jul-84
;     6   30-Apr-86  set clock to display 24 or 12 hour mode
;     7   29-Sep-86  conver 12:00A to zero in mode 2

            7          REVIS   =       7        ;NEW REVISION

            1          .TXTM   1                ;byte sex for strings

       103000          .LOC  LIDAT              ;assembly address

103000  40161 DATE:    IDATE                    ;the number of this routine and that it
                                                 ; is extended
103001      3          START-DATE               ;required starting address at location 1
103002 177053          DATE-DSBEN               ;size at location 2 by definition

103003  34445 START:   LDA     3,C30
103004  54440         STA     3,CLOCK            ;initialize for 24 hours
103005   4447         JSR     GO                 ;skip the parameter tables but place address
                                                 ; in A3

103006    366 TABLE:   DAYOU-TABLE              ;routine to output first field DAYOU or MONOU
103007    373          MONOU-TABLE              ;routine for second date field DAYOU, MONOU or
                                                 ; MONNM (for month names)
                                                 ; NOTE: these are addresses relative to TABLE

103010 140720          "A*400+100200+"P         ;Am and Pm

103011    415          MONNU-TABLE              ;address of routine to output month as a
                                                 ; number - included for completeness and
                                                 ; not actually accessed, use of this routine
                                                 ; causes the order of all numeric dates on
                                                 ; input to be assumed as dd/mm/yy
103012      0          0                        ;accept dd/mm/yy if 0 or mm/dd/yy if 1
```

```
;                    << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>

103013 145341           .TXTF    "Ja
103014 167255 n-
103015 143345 Fe
103016 161255 b-
103017 146741 Ma
103020 171255 r-
103021 140760 Ap
103022 171255 r-
103023 146741 Ma
103024 174655 y-
103025 145365 Ju
103026 167255 n-
103027 145365 Ju
103030 166255 l-
103031 140765 Au
103032 163655 g-
103033 151745 Se
103034 170255 p-
103035 147743 Oc
103036 172255 t-
103037 147357 No
103040 173255 v-
103041 142345 De
103042 161400 c"

103043      0           0                          ;terminate the month string
103044     30 CLOCK:    30                         ;presume the 24 hour clock
103045    255           "-+200                     ;delimiter in date

             1 FLD2.    =        1                 ;table address of the second date field
             2 MERI.    =        FLD2. +1          ;displacement to meridians
             4 MD.      =        MERI. +2          ;day/month : month/day indicator
             5 MONT.    =        MD. +1            ;where the month list starts
            36 F122.    =        CLOCK-TABLE       ;displacement to the clock entry in table
            37 DELI.    =        F122. +1          ;delimiter within date field

103046 177770 CM8:      177770                     ;used in counting to 8
103047 120240 SPACE:    " *400+" +100200           ;a pair of spaces
103050     30 C30:      30
103051      0 MODE:     0                          ;mode of CALL
103052      4 VSIZE:    EVTBL+1-VTABL              ;number of routines supported, size
                                                   ;  of the vector table


103053      7 REV:      REVIS                      ;core resident revision number

        103054 GO:      ;we actually start the code here.  Since we JSRed we have the      3
                        ; address of the parameters tables in A3
103054 165000           MOV      3,1               ;note for later                          1
103055   4404           JSR      EOSTK             ;skip the stack and load its address     3

103056      0           0                          ;subroutine return stack
103057      0           0
103060      0           0
```

```
;                    << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>

        103061 EOSTK:   ;end of stack, A3 contains address of stack
103061    451           JMP     EOST2           ;save the stack pointer
103062  50510 EOSTO:    STA     2,APT           ;A2 enters with the pointer to the args
103063   4505           JSR     STABL           ;pointer to table, set by GO, above
103064   4434           JSR     PICK            ;go and get the mode parameter          ?1??
103065  44764           STA     1,MODE          ;save the CALL mode
103066  30100           LDA     2,.INFO         ;the clock flag in INFO
103067   6102           FLAGCHECK               ;test the clock bit
103070  20006           SPCF.!SKIPO             ;is clock bit set?
103071  10000           IDCLK
103072    403           JMP     EOST1           ;   NO, 24 hour clock already set
103073  30034           LDA     2,C14           ;   YES, 12 hour clock requested
103074  50750           STA     2,CLOCK         ;set the new clock
103075   4405 EOST1:    JSR     VCEND           ;skip vector table and load its address    3

        103076 VTABL:   ;displacements to the various routines supported here
103076     36           RETRV-VTABL             ;return revision required
103077     75           RETDA-VTABL             ;return date as a string
103100    342           GETDA-VTABL             ;read string into numbers
103101    173 EVTBL:    ERRS-VTABL              ;not currently supported

        103102 VCEND:   ;vector table address exists in A3, where should we go          0  3
                        ; the mod eparameter is stored in A0
103102  20750           LDA     0,VSIZE         ;the number of vectors supported         1
103103  24746           LDA     1,MODE          ;get the CALL mode
103104 122433           SLE     1,0             ;is the mode switch valid
103105    564           JMP     ERRS            ;we have an illegal value
103106 171000           MOV     3,2             ;note the address of the VTABL           2
103107 137000           ADD     1,3             ;add the mode to the base of the table   3
103110  35400           LDA     3,0,3           ;get the displacement to the required    3
                                                ; routine from the table
103111 157000           ADD     2,3             ;add address of the table to make absol  3
103112  24561           LDA     1,.TABL         ;pass the table address to next phase    1
103113  30457           LDA     2,APT           ;it may also need APT                    2
103114   1400           JMP     0,3             ;go to the required routine
```

```
;                    << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>

        103115 PICKN:   ;pick up the value of the next parameter from the program
103115   10455          ISZ     APT             ;step the argument pointer, 2 word entries
103116   10454          ISZ     APT             ; there is no skip risk
103117   30453          LDA     2,APT           ;get the pointer
                                                ;drop into the PICK routine
;NO CODE HERE PLEASE

        103120 PICK:    ;pick the value of the next parameter, generate an error if      2
                        ; it is not numeric positive.  Place the value in A1
                        ; A2 contains APT
103120   56552          STA     3,@RETS         ;store return address in return stack
103121   25001          LDA     1,1,2           ;get the number type                     1
103122   31000          LDA     2,0,2           ;the address                              2
103123  102520          SUBZL   0,0             ;generate 1, decimal load              0
103124    6120          DECIMAL                 ;get the value into DA                 ????
103125    6121          FIX                     ;fix DA into A1, sign in A0            01??
103126     543          JMP     ERRS            ;error return if not integer
103127  101014          MOV#    0,0,SZR         ;check sign is positive
103130     541          JMP     ERRS            ;it was not
103131     424          JMP     RETN            ;return from subroutine

103132   54540 EOST2:   STA     3,RETS          ;save the stack pointer
103133     727          JMP     EOSTO           ;return

        103134 RETRV:   ;return revsion number of routine in YEAR
103134   24717          LDA     1,REV           ;get the core resident revision          1
103135    4402          JSR     SAVA1           ;save A1 as the next numeric paramet
103136    2113          JMP     @.SRET          ;perform a good return to basic


           131 SAVA.    =       .-TABLE         ;displacement to the routine
           132 SAVX.    =       SAVA.+1         ;the next entry point

        103137 SAVA1:   ;save A1 into the next parameter and step APT                    1 3
103137  102400          SUB     0,0             ;sign of returned value is +ve         0
        103140 SAVXX:   ;we are gosing to return a signed value, sign in A0            0
103140   56532          STA     3,@RETS         ;the return address
103141    6122          FLOAT                   ;float the number into DA              ????
103142   10430          ISZ     APT             ;step to next parameter
103143   10427          ISZ     APT             ; no risk of the skip
103144   30426          LDA     2,APT           ;pointer to input param table            2
103145   25001          LDA     1,1,2           ;number type of parameter                1
103146   31000          LDA     2,0,2           ;the address                              2
103147  125112          MOVL#   1,1,SZC         ;check parameter is not a string
103150     521          JMP     ERRS            ;it was
103151  102400          SUB     0,0             ;A0=0 for decimal store                0
103152    6120          DECIMAL                 ;store the number from DA to param     ????
103153     402          JMP     RETN            ;return from subroutine

        103154 PRETN:   ;pop stack and return
103154   14516          DSZ     RETS            ;pop the stack
        103155 RETN:    ;return from subroutine
103155   36515          LDA     3,@RETS         ;return address                           3
103156    1400          JMP     0,3             ;do the return
```

```
;                    << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>

          103157 SETST:  ;get address of the string parameter, check that dimension is
                         ; large enough and return A2 as string pointer
103157     30413         LDA     2,APT           ;get pointer to the argument list        2
103160     25015         LDA     1,15,2          ;the nmumber type of the string          1
103161    125123         MOVZL   1,1,SNC         ;is it a string and double it            1
103162       507         JMP     ERRS            ;NO
103163     20044         LDA     0,C40           ;double the minimum dimension 32         0
103164    122032         SGE     1,0             ;check dimension is large enough
103165       504         JMP     ERRS            ;it is not
103166     31014         LDA     2,14,2          ;get address of the string               2
103167      1400         JMP     0,3             ;return

103170     44503 STABL:  STA  1,.TABL          ;save pointer to TABLE
103171      1400         JMP  0,3            ;return

           164 APT.      =       .-TABLE         ;used for generation of the address
                                                ; of APT relative to the table
103172       0 APT:     0                       ;pointer to arguments from basic
```

```
;                    << SI = R92IDATESA; BO = 18/A. IDATE. 9291! >>

        103173 RETDA:   ;return in the string, the date that is specified by the
                        ; numeric parameters

103173   4764           JSR     SETST           ;find the string, set its address and     2
                                                ; check that the dimension is at least
                                                ; sixteen characters
103174  50575           STA     2,STPTR         ;note the string address for later
103175 102400           SUB     0,0             ;we have read no bytes yet                0
103176  40574           STA     0,BYTAD         ;so set the pointer
103177   4716           JSR     PICKN           ;go and pick the year                  ?1??
103200  44474           STA     1,YEAR          ;note it
103201   4714           JSR     PICKN           ;get the month
103202  44473           STA     1,MONTH
103203   4712           JSR     PICKN
103204  44472           STA     1,DAY

103205  30564           LDA     2,STPTR         ;address of the string                    2
103206  20641           LDA     0,SPACE         ;a pair of spaces for clearing            0
103207  24637           LDA     1,CM8           ;we are going to clear 8 words            1
        103210 PADLP:   ;pad the string with spaces
103210  41000           STA     0,0,2           ;clear next two bytes of string
103211 151400           INC     2,2             ;ready for the next bytes                 2
103212 125404           INC     1,1,SZR         ;reduce count                             1
103213    775           JMP     PADLP           ;we are not at zero yet

103214  34457           LDA     3,.TABL         ;address of the function table            3
103215  31400           LDA     2,0,3           ;the first vector, day or month output    2
103216 157000           ADD     2,3             ;vector is relative - make absolute       3
103217   5400           JSR     0,3             ;go and do it, eg output 01 for 1st
103220   4460           JSR     DELOU           ;output the delimter
103221  34452           LDA     3,.TABL         ;address of the function table            3
103222  31401           LDA     2,FLD2.,3       ;address of the second field output       2
                                                ; routine, eg Jan
103223 157000           ADD     2,3             ;convert from relative to table to abs.   3
103224   5400           JSR     0,3             ;go and output it
103225   4453           JSR     DELOU           ;delimiter between month and year      ????
103226  24446           LDA     1,YEAR          ;we have output the month and day             1
103227   4511           JSR     OUTNZ           ;output it with a zero
103230  24032           LDA     1,C12           ;byte at which to place time              1
103231  44541           STA     1,BYTAD         ;set time position, spaces are already
                                                ; in position from PADLP
103232   4663           JSR     PICKN           ;get the hour                             1
103233  44540           STA     1,TEMP          ;we will need it for meridian
103234  34437           LDA     3,.TABL         ;address of function table
103235  35436           LDA     3,F122.,3       ;the 12/24 hour flag                      3
103236 136433           SLE     1,3             ;is hour less than or equal to max
103237 166400           SUB     3,1             ;NO, so reduce by required hours          1
                                                ; eg 15 with a 12 hour clock => 3
103240  30034           LDA     2,C14           ;in case we have 12 hour and gen'd 0      2
103241 156415           SNE     2,3             ;skip if not a twelve hour clock
103242 125014           MOV#    1,1,SZR         ;skip if we did genberate zero hrs
103243    402           SKIP                    ;we were not 12 hour clock or did not
                                                ; generate zero so don't fix up
```

```
;                    << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>
;                  NO CODE HERE

103244 145000          MOV     2,1             ;the 0 => 12                          1
103245  20032          LDA     0,C12           ;see if there are any tens of hours   0
103246 122032          SGE     1,0             ;if so don't output a space
103247  10523          ISZ     BYTAD           ;since no tens of hours, skip one
                                               ; character to allign colon etc.
103250   4452          JSR     OUTN            ;output the hours from A1             ????
103251  20555          LDA     0,COLON         ;placed between hour and minute       0
103252   4430          JSR     OUTCH           ;get it out                           ????
103253  20520          LDA     0,TEMP          ;the hour before fix-up               0
103254  34417          LDA     3,.TABL         ;function table                         3
103255  31436          LDA     2,F122.,3       ;12 or 24 hour clock                  2
103256  35402          LDA     3,MERI.,3       ;the a and p                            3
103257  24034          LDA     1,C14           ;see if before noon                   1
103260 146414          SEQ     2,1             ;if it is 24 hour clock we are done
103261    405          JMP     MINUT           ;go and output the minutes as 24hr
103262 122433          SLE     1,0             ;if so we need the A
103263 175300          MOVS    3,3             ;after noon so we need the P            3
103264  30505          LDA     2,STPTR         ;where the string is                  2
103265  55007          STA     3,7,2           ;save meridian in string, both bytes will
                                               ; be stored but the wrong one will be over
                                               ; written by the minutes units
       103266 MINUT:   ;get and output the minutes
103266   4627          JSR     PICKN           ;get the minutes                      ?1??
103267   4451          JSR     OUTNZ           ;output them with a leading zero      ????
103270   2113 EXIT:    JMP     @.SRET          ;perform a good return to BASIC

       103271 ERRS:    ;something went wrong in the parameter set up
103271   2112          JMP     @.NRET          ;bad return to BASIC is error 38
                                               ; something went wrong

103272 103054 RETS:    EOSTK-5                 ;absolute stack pointer
103273 103006 .TABL:   TABLE                   ;the absolute address of the table

103274      0 YEAR:    0                       ;Year
103275      0 MONTH:   0                       ;month to process
103276      0 DAY:     0                       ;day

                                               ; set by intialization routine

103277    655 JPRET:   JMP     PRETN           ;stepping stone

       103300 DELOU:   ;output a delimiter
103300  30773          LDA     2,.TABL         ;functional table                       2
103301  21037          LDA     0,DELI.,2       ;get the delimiter                    0
                      ;JMP     OUTCH           ;output it
```

```
;                        << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>
;NO CODE HERE PLEASE

          103302 OUTCH:   ;output the character contained in A0 to the string.
  103302   56770          STA     3,@RETS          ;save address on return stack
  103303   34064          LDA     3,C377           ;mask for one byte                    3
  103304  163400          AND     3,0              ;do it                              0
  103305   30464          LDA     2,STPTR          ;where is the string                 2
  103306   24464          LDA     1,BYTAD          ;which byte                         1
  103307  125220          MOVZR   1,1              ;set carry if an even byte and convert  1  C
                                                   ; to a word displacement                C
  103310  133000          ADD     1,2              ;make the absolute address           2 C
  103311   25000          LDA     1,0,2            ;the word to set into                1 C
  103312  175303          MOVS    3,3,SNC          ;place mask in high end of word and     3C
                                                   ; skip the next instruction if an
                                                   ; even we are to overlay an even byte
  103313  125300          MOVS    1,1              ;its an odd byte so swap             1  C
  103314  167400          AND     3,1              ;remove the byte that we will replace  1  C
  103315  123003          ADD     1,0,SNC          ;add in the new byte and see if we      0
                                                   ; need to check sex
  103316  101300          MOVS    0,0              ;swap if sex is wrong               0
  103317   41000          STA     0,0,2            ;place back in string
  103320   10452          ISZ     BYTAD            ;we have output another byte
  103321     634  JRETN:  JMP     RETN             ;return from subroutine


          103322 OUTN:    ;output the number contained in A1
  103322  152400          SUB     2,2              ;zero the flag as no leading 0 allowed    2
          103323 OUTNX:   ;output digit(s)
  103323   50414          STA     2,OUTNF          ;save the leading zero flag
  103324   56746          STA     3,@RETS          ;return address
  103325   10745          ISZ     RETS             ;increase stack pointer to allow
                                                   ; return address within SPLIT
  103326    4414          JSR     SPLIT            ;split A1 into tens in A0 and UNITS     0???
  103327   30060          LDA     2,C260           ;ascii zero                            2
  103330   24407          LDA     1,OUTNF          ;get the leading zero supression flag    1
  103331  125005          MOV     1,1,SNR          ;if the leading zero is to be output
                                                   ; then always output
  103332  142414          SEQ     2,0              ;skip if we have a zero
                                                   ; the above statements are equivalent to
                                                   ;   IF A1=0 IF A0="0" then skip output
          103333 OUTN1:   ;output the tens
  103333    4747          JSR     OUTCH            ;output the character
  103334   20471          LDA     0,UNITS          ;the units                          0
  103335    4745          JSR     OUTCH            ;get them out
  103336     616          JMP     PRETN            ;pop stack and return

  103337       0 OUTNF:   0                        ;zero for leading zero supression
                                                   ; 1 otherwise

          103340 OUTNZ:   ;output digits with leading zeros
  103340  152520          SUBZL   2,2              ;allow leading zeros                   2
  103341     762          JMP     OUTNX            ;actually output the digits
```

```
;                    << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>

        103342 SPLIT:  ;divide the number in A1 into two ascii printable digits          0
                       ;  the tens are placed in A0,  the units in UNITS

                       ;the algorithm used is:-
                       ;       D2=H2:  D1=H1*6+H2
                       ;    L:  IF D1>9 D1=D1-10:  D2=D2+1:  GOTO L
                       ;where:-
                       ;       H1 and H2 are hex digits of the value and
                       ;       D1 and D2 are the generated decimal oned

103342  56730          STA     3,@RETS          ;save the return address

                       ;split the value into its hex digits
103343  34037          LDA     3,C17            ;four bit mask                          3
103344 137400          AND     1,3              ;A3<=H1                                 3
103345 127120          ADDZL   1,1              ;A1=A1*4                              1
103346 127120          ADDZL   1,1              ;  bits 4..7 -> 8..11                 1
103347  30460          LDA     2,C7400          ;111 100 000 000 (bits 8..11)           2
103350 133700          ANDS    1,2              ;mask out all other bits and place      2
                                                ;  8..11 into 0..3
                                                ;A2<=H2
103351 141000          MOV     2,0              ;D2=H2                                0
103352 103000          ADD     0,0              ;A0=D2*2                              0
103353 143120          ADDZL   2,0              ;A0=(D2*2+D2)*2                       0
                                                ;  =D2*3*2 =D2*6
103354 117000          ADD     0,3              ;D1=H1+D2*6                             3
103355  20032          LDA     0,C12            ;10                                   0
        103356 SPLI1:  ;L: of above algorithm
103356 162032          SGE     3,0              ;D1>9
103357    404          JMP     SPLI2            ;NO
103360 116400          SUB     0,3              ;YES, D1=D1-10                          3
103361 151400          INC     2,2              ;D2=D2+1                              2
103362    774          JMP     SPLI1            ;GOTO L

        103363 SPLI2:  ;D1 and D2 have been adjusted
103363  24060          LDA     1,C260           ;the ascii "0"                        1
103364 137000          ADD     1,3              ;convert units to ascii                 3
103365  54440          STA     3,UNITS          ;save
103366 147000          ADD     2,1              ;covert tens
103367 121000          MOV     1,0              ;tens are returned in A0              0
103370    731          JMP     JRETN            ;return via stack

103371      0 STPTR:   0                        ;pointer to start of string parameter
103372      0 BYTAD:   0                        ;byte address within string
103373      0 TEMP:    0                        ;temporary storage


        103374 DAYOU:  ;output a day number
103374  24702          LDA     1,DAY            ;the day has already been found       1
        103375 DATOX:  ;used for output of other numeric fields
103375  56675          STA     3,@RETS          ;save return address
103376  10674          ISZ     RETS             ;make stack space for others
103377   4741          JSR     OUTNZ             ;output day preserve zero
103400    677          JMP     JPRET            ;pop stack and return
```

```
;                     << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>

         103401 MONOU:   ;output the month as a name
 103401   56671          STA     3,@RETS             ;return address
 103402   10670          ISZ     RETS                ;make stack space
 103403   24672          LDA     1,MONTH             ;we already got the month number          1
 103404  125120          MOVZL   1,1                 ;Jan=2, Feb=4, etc                        1
 103405   30666          LDA     2,.TABL             ;address of the table functions            2
 103406  133000          ADD     1,2                 ;add displacement for this month +2        2
 103407   21003          LDA     0,MONT.-2,2         ;first two chars for the month           0
                                                     ; the displacement -2 is required
                                                     ; to convert from base 0 to base 1
                                                     ; with two words per month name
 103410   50415          STA     2,UNITS             ;note where we are
 103411   40762          STA     0,TEMP              ;note them since we will need the 2nd
 103412  101300          MOVS    0,0                 ;output the first char first             0
 103413    4667          JSR     OUTCH               ;output the first                        ????
 103414   20757          LDA     0,TEMP              ;get the chars again                     0
 103415    4665          JSR     OUTCH               ;put out the second                      ????
 103416   30407          LDA     2,UNITS             ;where this month is in the table         2
 103417   21004          LDA     0,MONT.-1,2         ;get the third character                 0
 103420  101300          MOVS    0,0                 ;place third character not 4th for out   0
 103421    4661          JSR     OUTCH               ;put it out                              ????
 103422     655          JMP     JPRET               ;pop stack and return

         103423 MONNU:   ;output month as a number
 103423   24652          LDA     1,MONTH             ;we saved it before                       1
 103424     751          JMP     DATOX               ;output a number

 103425       0 UNITS:   0                           ;units of the number to display, used
                                                     ; for remembering the address of the
                                                     ; month name for a very short period
                                                     ; in the routine MONOU
 103426     272 COLON:   ":+200                      ;time delimiter
 103427    7400 C7400:   7400                        ; 111 100 000 000 bit 8..11 mask

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 103430     340 C340:    340                         ;diffenece between 1 and "a used to
                                                     ; convert from letter number to LCS

 103431       0 TEMPY:   0                           ;working storage

         103432 SAVM1:   ;save a minus one as an error indicator in the next param
 103432  102520          SUBZL   0,0                 ;generate a magnitude of one             0
 103433  105000          MOV     0,1                 ;and sign negative
         103434 JSAVX:   ;jump off to the routine to save a signed A1
 103434   30637          LDA     2,.TABL             ;base of the code                         2
 103435    1132          JMP     SAVX.,2             ;jump to the routine and save the A1

         103436 JSAVA:   ;jump to the routine that saves A1 as the next param
 103436   30635          LDA     2,.TABL             ;convert relative to absolute
 103437    1131          JMP     SAVA.,2             ;go an perform the routine
```

```
;                    << SI = R92IDATESA;  BO = 18/A.IDATE.9291! >>

        103440 GETDA:   ;get the date from the string and return it as a list of          12
                        ; numbers, this is the mode 2 call routine
                        ; A1 and A2 contain the address of the table and the APT

103440    401           JMP     .+1             ;debug code only
103441   4455           JSR     STTAB           ;go and save the A1 pointer
103442  50767           STA     2,TEMPY         ;note as we will damage the value
103443   4767           JSR     SAVM1           ;save minus one as all the values,        ????
103444   4766           JSR     SAVM1           ; so that we assume an error, YR, MON
103445   4765           JSR     SAVM1           ; day
103446   4764           JSR     SAVM1           ; hour
103447   4763           JSR     SAVM1           ; minute
103450  34623           LDA     3,.TABL         ;address of the function table             3
103451  11564           ISZ     APT.,3          ;step arg pointer past the string, we
103452  11564           ISZ     APT.,3          ; need not worry about the zero
                                                ; condition

103453   4757           JSR     SAVM1           ;set error flag to -1
103454  30755           LDA     2,TEMPY         ;restore the argument pointer               2
103455  34577           LDA     3,.TABX         ;address of absolute tables                 3
103456  51564           STA     2,APT.,3        ;reset the argument pointer
103457   4576           JSR     JMAKR           ;make the byte address for the string     ????
                                                ; relative to SBA
103460   4435           JSR     JRDNM           ;read the day field, or month if USA      ?1??
103461  44467           STA     1,FIRST         ;it is the first that we have read
103462   4576           JSR     GETCH           ;skip the delimiter
103463   4575           JSR     GETCH           ;get the next character from strin        ???2
103464   6130           ISA2LETTER              ;see if it was a month name               ????
103465    464           JMP     NOTAL           ;NO, so its a numeric second field
                        ;we have a named month
103466 102400           SUB     0,0             ;assume month = Jan                       0
103467  40460           STA     0,MONTX         ;and note the fact
103470  20062           LDA     0,C300          ;convert the 'a or 'A to 'A, ISA2L        0
                                                ; returns the letter number
103471 113300           ADDS    0,2             ;do conversion and make it the 1st byt      2
                                                ; eg "J *400
103472  50564           STA     2,TEMPX         ;note what we have made
103473   4565           JSR     GETCH           ;get the next character                   ???2
103474   6130           ISA2LETTER              ;is this an alpha
103475    447           JMP     BADMO           ;NO so we have an illegal month name
103476  20732           LDA     0,C340          ;convert the character to lower case
103477 113000           ADD     0,2             ; eg 'a or 'A becomes 'a and place it       2
                                                ; in the second byte
103500  20556           LDA     0,TEMPX         ;get back the first character             0
103501 143000           ADD     2,0             ;place them together                      0
                                                ; eg "J*400 +"a
103502  40554           STA     0,TEMPX         ;remember the pair of characters
                                                ; eg 'Ja
103503  34551           LDA     3,.TABX         ;the address of the table of all the        3
                                                ; parameters
```

```
;                    << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>

        103504 MONLP:   ;loop here seeing if we have found the required month in
                        ; the parameter table
103504   25405          LDA     1,MONT.,3       ;get the first two characters of the      1
                                                ; next month to check
103505   10442          ISZ     MONTX           ;we are checking the next, Ja =1 etc
103506  106415          SNE     0,1             ;skip if we have not found the month
103507     411          JMP     DAYI3           ;the first two chars match
103510  125015          MOV#    1,1,SNR         ;are we at the end of month list
103511     433          JMP     BADMO           ;YES - its a bad month name
        103512 NXTMO:   ;we must check the next month as some character is wrong
103512  175400          INC     3,3             ;ready to try next month                    3
103513  175400          INC     3,3             ; two words per month name                  3
103514     770          JMP     MONLP           ;try the next month name

103515     552 JRDNM:   JMP     RDNUM           ;stepping stone

        103516 STTAB:   ;save A1 as the table pointer                                       1
103516   44536          STA     1,.TABX         ;address of function table
103517    1400          JMP     0,3



        103520 DAYI3:   ;the first two characters of the month name match, try the 3rd
103520   54571          STA     3,TEMP1         ;where we are getting characters
                                                ; from in the parameter table
103521   40535          STA     0,TEMPX         ;the first two characters from string
103522    4536          JSR     GETCH           ;get next character                       ??2?
103523    6130          ISA2LETTER              ;chekk that its alpha
103524     420          JMP     BADMO           ;NO, its not so we have an error
103525   20703          LDA     0,C340          ;convert letter number to a lower         0
103526  113000          ADD     0,2             ; case character in ls byte                 2
103527   34562          LDA     3,TEMP1         ;where we are getting the month name        3
                                                ; from
103530   21406          LDA     0,1+MONT.,3     ;get the third and fourth chars           0
103531  101300          MOVS    0,0             ;place character of interest in           0
                                                ; low byte
103532   24064          LDA     1,C377          ;reduce to the third                        1
103533  107400          AND     0,1             ;                                           1
103534   20522          LDA     0,TEMPX         ;restore the Ja in case we return           1
103535  132414          SEQ     1,2             ;see if the tird characters match
103536     404          JMP     NXTM            ;they don't

                        ;we have a three character match
103537   20411          LDA     0,FIRST         ;the day that we stored                   0
103540   40552          STA     0,DAYX          ;save it as DAY
103541     425          JMP     YEARS           ;month was set above so we are ready
                                                ; for the years

        103542 NXTM:    ;we have discovered that the third character does not match
103542   14523          DSZ     BYTAX           ;so that we don't think that we have
                                                ; read and checked the character
                                                ; and so we read it again
103543     747          JMP     NXTMO           ;go and see if any more months have
                                                ; matching first pairs of characters
```

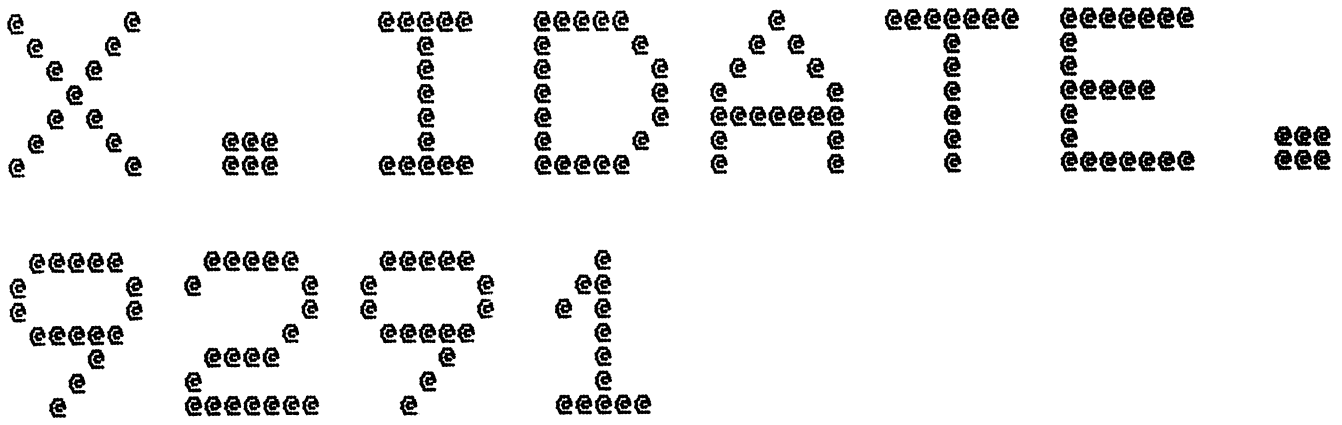
```
;                    << SI = R92IDATESA; BO = 18/A.IDATE.9291! >>

          103544 BADMO:   ;the month has been detected as bad, prepare to return the
                          ; appropriate status
103544 126400             SUB     1,1             ;generate a zero for the year          1
103545   4671             JSR     JSAVA           ;save as the first parameter
103546    542             JMP     XEXIT           ;we are done, as all other params
                                                  ; were already set -1


103547      0 MONTX:    0                         ;month number
103550      0 FIRST:    0                         ;the days or in USA months

          103551 NOTAL:   ;the second field is not alpha, assume it is the month or
                          ; the day if USA
103551  14514             DSZ     BYTAX           ;reduce the byte pointer as we
                                                  ; must re-read the character as
                                                  ; part of the number
103552   4515             JSR     RDNUM           ;get the numeric value                 ?1??
103553  44503             STA     1,TEMPX         ;this is the second numeric field
103554  34500             LDA     3,.TABX         ;the address of the function table       3
103555  21404             LDA     0,MD.,3         ;the input order flag mm/dd or dd/mm   0
103556  24500             LDA     1,TEMPX         ;assume US order, month first          1
103557  30771             LDA     2,FIRST         ;and day second                          2
103560 101014             MOV#    0,0,SZR         ;skip if we have dd/mm/yy format
103561    403             JMP     MFRST           ;we require the mm/dd/yy
103562  24766             LDA     1,FIRST         ;get the first field read              1
103563  30473             LDA     2,TEMPX         ;the second field found                  2
          103564 MFRST:   ;American or UK order of day/month is in A1 an A2
103564  44526             STA     1,DAYX          ;it is the day
103565  50762             STA     2,MONTX         ;it was the month

          103566 YEARS:   ;we have recognized and stored the month and day
103566  10477             ISZ     BYTAX           ;skip the delimiter
103567   4500             JSR     RDNUM           ;read a number, the years              ?1??
103570   4523             JSR     CENTU           ;reduce 20xx or 19xx to xx             ?1??
103571   4645             JSR     JSAVA           ;it is years which is first param
103572  24755             LDA     1,MONTX         ;next comes months                     1
103573   4643             JSR     JSAVA           ;output it
103574  24516             LDA     1,DAYX          ;get the days                          1
103575   4641             JSR     JSAVA           ;they are the third parameter

                          ;DECIMAL has skipped the spaces between fields for us
103576   4471             JSR     RDNUM           ;read the hours                        ?1??
103577  44512             STA     1,TEMP1         ;note the value read
103600   4460             JSR     GETCH           ;get next character                    ??2?
103601  20625             LDA     0,COLON         ;the character to check                0
103602 105400             INC     0,1             ;the semi-colon
103603 142414             SEQ     2,0             ;is it a colon?
103604 146415             SNE     2,1             ;is it a semi-colon?
103605    402             SKIP                    ;it is a colon or a semi-colon
103606    502             JMP     XEXIT           ;NO, so we have an error exit
103607  24502             LDA     1,TEMP1         ;restore the hours value               1
103610   4626             JSR     JSAVA           ;save as hours
103611   4456             JSR     RDNUM           ;read the minutes                      ?1??
103612   4624             JSR     JSAVA           ;save as minutes
```

```
;                    << SI = R92IDATESA;  BO = 18/A. IDATE. 9291! >>

;                    NO CODE HERE

103613  34441           LDA     3,.TABX         ;absolute address of table                  3
103614  11564           ISZ     APT.,3          ;skip past the string
103615  11564           ISZ     APT.,3          ; two word table
103616 126400           SUB     1,1             ;clear the error flag                    1
103617   4617           JSR     JSAVA           ;time is OK
103620   4440           JSR     GETCH           ;get the next character                ???2
103621   6130           ISA2LETTER              ;convert to a letter number            ???2
103622    466           JMP     XEXIT           ;its not a letter, so not PM
103623  50433           STA     2,TEMPX         ;make a note of it
103624  34430           LDA     3,.TABX         ;where the table is                       3
103625  31402           LDA     2,MERI.,3       ;the meridian characters                2
103626  20064           LDA     0,C377          ;the byte mask                        0
103627 113400           AND     0,2             ;get afternoon idicator from word       2
103630   6130           ISA2LETTER              ;convert to the letter number for      ???2
                                                ; case insensitivity
103631    457           JMP     XEXIT           ;no letter, therefore we can't recnize
103632 102400           SUB     0,0             ;initilize the offset
103633  24423           LDA     1,TEMPX         ;the chacater that we read               1
103634 132414           SEQ     1,2             ;is it afternoon
103635  20422           LDA     0,CM12          ;  NO, set offset
103636  24453           LDA     1,TEMP1         ;the hours stored before                 1
103637  30034           LDA     2,C14           ;noon
103640 132433           SLE     1,2             ;is it already converted to affternoon ?
103641    447           JMP     XEXIT           ;  YES, we'er done
103642 132414           SEQ     1,2             ;  NO, is it noon ?
103643 147000           ADD     2,1             ;  NO, convert to afternoon
103644 107000           ADD     0,1             ;adjust with offset (0 if PM  -12 if AM)
103645  34407           LDA     3,.TABX         ;address of the table, absolute           3
103646  31564           LDA     2,APT.,3        ;the address of the status field        2
103647  20030           LDA     0,C10           ;step back past the string, mins      0
                                                ; and hours
103650 112400           SUB     0,2             ;so we may re-set the hours
103651  51564           STA     2,APT.,3        ;where expected by store routine
103652   5531           JSR     SAVA.,3         ;the address of the save A1 routine
103653    435           JMP     XEXIT           ;we are done

103654      0 .TABX:    0
103655    425 JMAKR:    JMP     MAKRL           ;stepping stone to make relative
103656      0 TEMPX:    0                       ;temporary storage
103657 177764 CM12:     177764                  ;-12

        103660 GETCH:   ;read the next character from the string
103660  54406           STA     3,RETX          ;save return address
103661  24404           LDA     1,BYTAX         ;where to get from
103662   6144           XGETBYTE                ;we are to get it                      ???2
103663  10402           ISZ     BYTAX           ;we got another one
103664   2402           JMP     @RETX           ;we are done, return

103665      0 BYTAX:    0                       ;byte pointer within string relative
                                                ; to SBA
103666      0 RETX:     0                       ;subroutine return address
```

```
;                    << SI = R92IDATESA; BO = 18/A. IDATE. 9291! >>

        103667 RDNUM:   ;read a number from the string
103667   54777          STA     3,RETX          ;note the return address
103670   24775          LDA     1,BYTAX         ;get the address                    1
103671   20026          LDA     0,C6            ;tell decimal to do an input        0
103672    6120          DECIMAL                 ;read an ascii sequence
103673     415          JMP     XEXIT           ;we have failed so return with the
                                                ; -1 set
103674   44771          STA     1,BYTAX         ;save address of next byte to get
103675    6121          FIX                     ;convert number to integer          01??
103676     412          JMP     XEXIT           ;return with -1 if not valid
103677  101014          MOV#    0,0,SZR         ;check that sign is positive
103700     410          JMP     XEXIT           ;it was not so return the pre-set err
103701    2765          JMP     @RETX           ;we are done

        103702 MAKRL:   ;make BYTAX to a relative to SBA address             2
103702   25014          LDA     1,14,2          ;address of the string              1
103703   20040          LDA     0,SBA           ;the source byte address            0
103704  106420          SUBZ    0,1             ;find difference                     1
103705  125120          MOVZL   1,1             ;convert to bytes                    1
103706   44757          STA     1,BYTAX         ;save as byte address
103707    1400          JMP     0,3             ;we are done

103710    2113 XEXIT:   JMP     @.SRET          ;return with no error 38

103711       0 TEMP1:   0                       ;temporary storage

103712       0 DAYX:    0                       ;storage of day number

        103713 CENTU:   ;remove any complete centuries from the year as entered
                        ; A1 eneters and leaves with the year

103713   20410          LDA     0,C1900         ;remove a century at time           0
103714  122032          SGE     1,0             ;are we still greater than 100
103715    1400          JMP     0,3             ;NO, return
103716  106400          SUB     0,1             ;reduce year to base 1900           1
103717   20405          LDA     0,HUNDR         ;if we are in the 21st cent         0
103720  122033          SLS     1,0             ; then redmove the 100 years
103721  106400          SUB     0,1             ;21st cent so remove 100             1
103722    1400          JMP     0,3             ;we are done with the century fix

103723    3554 C1900:   3554                    ;1900 as an octal number
103724     144 HUNDR:   144                     ;difference between 2000-1900

        103725 DSBEND   =       .               ;where are we

             0          .ERR    DATE+1000<.;OVERFLOW CHECK

                .END
```

APT	103172	APT.	164	BADMO	103544	BINDI	6115	BINMU	6116		
BPI	16	BSACF	75	BUMPU	6117	BYTAD	103372	BYTAX	103665		
C10	30	C100	51	C1000	67	C11	31	C12	32		
C13	33	C14	34	C15	35	C16	36	C160	174		
C163	175	C166	176	C17	37	C170K	21	C171	177		
C177	52	C1777	70	C1900	103723	C2	2	C20	42		
C200	53	C2000	71	C205	54	C215	55	C240	56		
C244	57	C260	60	C271	61	C3	3	C30	103050		
C300	62	C334	63	C340	103430	C37	43	C377	64		
C4	24	C40	44	C400	65	C4000	72	C5	25		
C6	26	C600	100	C7	27	C7400	103427	C77	50		
C774C	22	C777	66	CALL	6101	CENTU	103713	CHANN	6106		
CLOCK	103044	CM12	103657	CM400	23	CM8	103046	COLON	103426		
DA	160	DAC	164	DAS	165	DATAP	6110	DATE	103000		
DATOX	103375	DAY	103276	DAYI3	103520	DAYOU	103374	DAYX	103712		
DB	166	DBA	41	DBC	172	DBS	173	DECIM	6120		
DELI.	37	DELOU	103300	DFTCA	34106	DMCAL	34110	DQUEU	6105		
DSBEN	103725	EOSTO	103062	EOST1	103075	EOST2	103132	EOSTK	103061		
ERRF	76	ERRS	103271	ESCF	73	ETSF	74	EVTBL	103101		
EXIT	103270	F122.	36	FINDL	6123	FIRST	103550	FIX	6121		
FLAGC	6102	FLD2.	1	FLOAT	6122	FREEN	6107	GETBY	6124		
GETCH	103660	GETDA	103440	GO	103054	HALTS	6153	HUNDR	103724		
INBYT	6125	INSTB	6126	IOCAL	34103	IOP	6	ISA2D	6127		
ISA2L	6130	JFLTO	151	JMAKR	103655	JPRET	103277	JRDNM	103515		
JRETN	103321	JSAVA	103436	JSAVX	103434	LACNT	4000	LAFSE	13000		
LALCO	47400	LALLO	1400	LATOE	36000	LBAKU	106000	LBILD	5000		
LBUIL	4400	LCALL	75000	LCHAN	41000	LCHFL	30000	LCHSU	61000		
LCLEA	7400	LCLOS	7000	LCLPY	76000	LCNVA	11400	LCNVD	12000		
LCOMM	33400	LDALC	2000	LDALL	1000	LDB7A	114000	LDB7B	114400		
LDB7C	115000	LDB7D	115400	LDB7E	116000	LDB7F	116400	LDB7G	117000		
LDB7H	117400	LDB7I	120000	LDEKE	52400	LDELE	3400	LDIRE	50400		
LDLTP	20400	LDREN	37400	LDSB1	400	LDSB2	22400	LDSB3	47000		
LDSB4	65000	LDSB5	77000	LDSB6	106400	LDSB7	113400	LECHO	37000		
LEO87	105400	LERRO	23000	LFAUL	400	LFFIL	2400	LFIXD	57400		
LFNDC	112000	LFNDL	20000	LFOFI	17000	LGETR	10000	LGHOP	107400		
LGHOS	107000	LGMUX	16000	LHCON	17400	LIBCA	44400	LIBEN	45000		
LIBTR	45400	LIDAT	103000	LLINK	35400	LLOAD	34400	LLOGI	32000		
LLUIN	112400	LMAPB	73000	LMDEO	65000	LMDE1	66000	LMDE5	71400		
LMRC3	56400	LMRFH	57000	LMRFI	54000	LMTAP	55400	LMTAS	54400		
LMTFP	56000	LMTFY	60400	LMTNX	55000	LMTPL	60000	LOADD	6131		
LOPEN	6000	LOPNM	13400	LPATQ	110000	LPEXP	23400	LPFAB	72000		
LPFLN	73400	LPFNA	3000	LPFRL	72400	LPFSE	67000	LPFSH	70000		
LPFSX	70400	LPLOG	24400	LPPWR	33000	LPRAN	36400	LPRCO	71000		
LPSIN	25400	LPSQR	22400	LPTAN	25000	LQIBF	63400	LQICL	63000		
LQIOP	62400	LRDFH	26400	LRDIS	31400	LRDSE	110400	LREDC	50000		
LREDI	11000	LREDM	14000	LREDP	74000	LRENA	15000	LREOP	53000		
LRESO	42000	LRWIT	113000	LRWMB	14400	LRWSX	111400	LS105	77000		
LS152	102000	LS153	101000	LS154	100400	LS156	101400	LS157	100000		
LSAVE	43000	LSAVP	43400	LSEAB	64000	LSEAR	51000	LSETF	40000		
LSHUF	52000	LSIGP	12400	LSING	40400	LSMCS	106400	LSPEC	27000		
LSTRI	32400	LSYSC	30400	LTPO1	102400	LTPO3	104000	LTPO4	104400		
LTPO5	105000	LVMUX	42400	LWRIT	47000	LXMIN	62000	MAKRL	103702		
MD.	4	MERI.	2	MFRST	103564	MINUT	103266	MODE	103051		
MONLP	103504	MONNU	103423	MONOU	103401	MONTH	103275	MONTX	103547		
MONT.	5	NOTAL	103551	NXTM	103542	NXTMO	103512	OUTBY	6132		
OUTCH	103302	OUTN	103322	OUTN1	103333	OUTNF	103337	OUTNX	103323		
OUTNZ	103340	OUTTE	6133	PADLP	103210	PIB	4	PICK	103120		
PICKN	103115	PRETN	103154	PUTBY	6134	QCHAR	6103	QUEUE	6104		

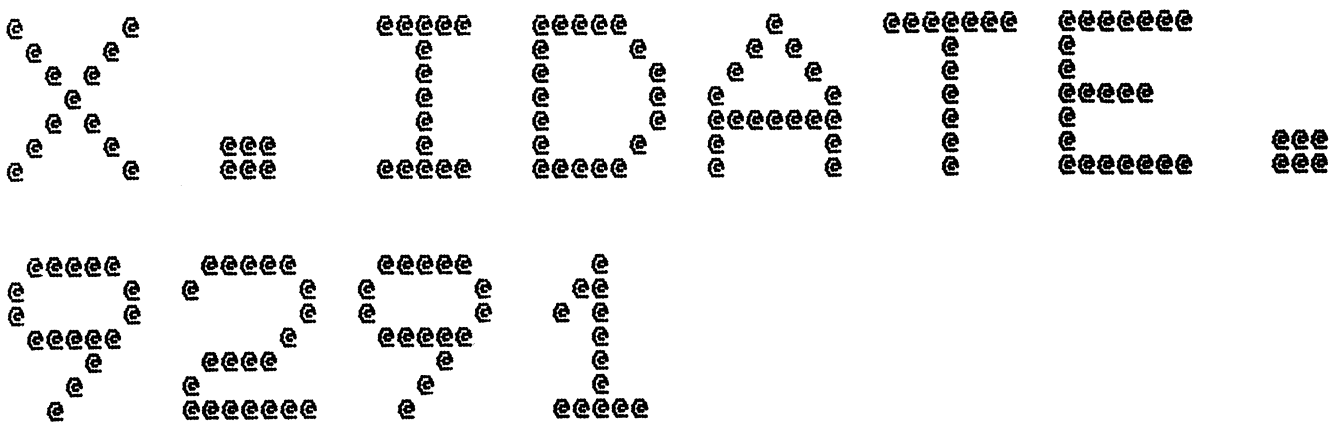
RDNUM	103667	READB	6135	RELJM	6136	RETDA	103173	RETN	103155
RETRV	103134	RETS	103272	RETX	103666	REV	103053	REVIS	7
RJSR	6136	RTP	7	RUP	5	SAVA1	103137	SAVA.	131
SAVM1	103432	SAVXX	103140	SAVX.	132	SBA	40	SCDCA	34147
SETST	103157	SPACE	103047	SPINP	6146	SPLI1	103356	SPLI2	103363
SPLIT	103342	STABL	103170	START	103003	STINP	6140	STINT	6147
STORD	6137	STOUT	6141	STPTR	103371	STTAB	103516	TABLE	103006
TASKQ	15	TEMP	103373	TEMP1	103711	TEMPX	103656	TEMPY	103431
TRAPF	6142	UNITS	103425	VCEND	103102	VSIZE	103052	VTABL	103076
WRITB	6143	XEXIT	103710	XGETB	6144	XPUTB	6145	YEAR	103274
YEARS	103566	.ABA	14	.BPS	77	.BSA	10	.DA	174
.DA3	175	.DB	176	.DB3	177	.FLTO	152	.HBA	11
.HXA	12	.INFO	100	.INTR	111	.LCM	114	.NRET	112
.SRET	113	.SSA	13	.TABL	103273	.TABX	103654		



Spool Queue Line #: 42
IRIS LU/Filename : 18/X.IDATE.9291

Printed on/at : FEB 6, 1990 12:56:20
For Group/User: 0 , 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For R9.5 RELSE CN



Spool Queue Line #: 42
IRIS LU/Filename : 18/X.IDATE.9291

Printed on/at : FEB 6, 1990 12:56:30
For Group/User: 0 , 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For R9.5 RELSE CN

***** J O B S T A T I S T I C S *****

1	TOTAL # DUPLICATE KEYS
0	TOTAL # DIR. RE-ORGS
420	TOTAL # KEYS INSERTED
0	TOTAL # ASSEMBLY ERRS

. ERR 17.055

. INFO 5.013

. NRET 9.037

. SRET 6.034 9.033 17.031

. TABL 5.039 7.021 8.033 8.038 8.051 9.018 9.041:
9.053 12.012 12.054 12.058 13.020

. TABX 13.027 13.056 14.024 15.025 16.010 16.019 16.037
16.046:

. TXTF 4.007

APT 5.009 5.040 6.008 6.009 6.010 6.045 6.046
6.047 7.010 7.026:

APT. 7.024= 13.021 13.022 13.028 16.011 16.012 16.038
16.042

BADMO 13.047 14.015 14.035 15.007:

BYTAD 8.016 8.047 9.012 10.013 10.027 11.051:

BYTAX 14.056 15.020 15.038 16.053 16.055 16.058: 17.010
17.015 17.028

C10 16.039

C12 8.046 9.010 11.034

C14 5.018 8.056 9.021 16.031

C17 11.021

C1900 17.040 17.049:

C260 10.039 11.043

C30 3.043 4.045:

C300 13.040

C340 12.045: 13.048 14.036

C377 10.010 14.043 16.021

C40 7.014

C6 17.011

C7400 11.025 12.040:

CENTU 15.040 17.037:

CLOCK 3.044 4.033: 4.040 5.019

CM12 16.029 16.049:

CM8 4.043: 8.026

COLON 9.015 12.039: 15.051

DATE 3.038: 3.040 3.041 17.055

DATOX 11.057: 12.033

DAY 8.022 9.045: 11.056

DAYI3 14.013 14.029:

DAYOU 3.048 11.055:

DAYX 14.051 15.034 15.044 17.035:

DECIM 6.021 6.053 17.012

DELI. 4.041= 9.054

DELOU 8.037 8.043 9.052:

DSBEN 3.041 17.053=

EOST0 5.009: 6.029

EOST1 5.017 5.020:

EOST2 5.008 6.028:

EOSTK 4.056 5.007: 9.040

ERRS 5.026 5.033 6.023 6.025 6.051 7.013 7.016
9.036:

EVTBL 4.047 5.026:

EXIT 9.033:

F122. 4.040= 4.041 8.052 9.019

FIRST 13.032 14.050 15.016: 15.028 15.031

FIX 6.022 17.016

FLAGC 5.014

FLD2. 4.036= 4.037 8.039

FLOAT 6.044

GETCH 13.033 13.034 13.045 14.033 15.050 16.015 16.051:

GETDA 5.025 13.007:

GO 3.045 4.053:

HUNDR 17.044 17.050:

IDATE 3.038

IDCLK 5.016

ISA2L 13.035 13.046 14.034 16.016 16.023

JMAKR 13.029 16.047:

JPRET 9.049: 11.061 12.028

JRDNM 13.031 14.021:

JRETN 10.028: 11.048

JSAVA 12.057: 15.010 15.041 15.043 15.045 15.058 15.060
16.014

JSAVX 12.053:

LIDAT 3.036

MAKRL 16.047 17.023:

MD. 4.038= 4.039 15.026

MERI. 4.037= 4.038 9.020 16.020

MFRST 15.030 15.033:

MINUT 9.023 9.030:

MODE 4.046: 5.012 5.031

MONLP 14.007: 14.019
MONNU 3.055 12.031:
MONOU 3.049 12.007:
MONT. 4.039= 12.014 12.025 14.009 14.040
MONTH 8.020 9.044: 12.010 12.032
MONTX 13.039 14.011 15.015: 15.035 15.042
NOTAL 13.036 15.018:
NXTM 14.047 14.055:
NXTMO 14.016: 14.059
OUTCH 9.016 10.008: 10.047 10.049 12.021 12.023 12.027
OUTN 9.014 10.031:
OUTN1 10.046:
OUTNF 10.034 10.040 10.052:
OUTNX 10.033: 10.057
OUTNZ 8.045 9.032 10.055: 11.060
PADLP 8.027: 8.031
PICK 5.011 6.014:
PICKN 6.007: 8.017 8.019 8.021 8.049 9.031
PRETN 6.056: 9.049 10.050
RDNUM 14.021 15.023 15.039 15.048 15.059 17.008:
RETDA 5.024 8.008:
RETN 6.026 6.054 6.058: 10.028
RETRV 5.023 6.031:
RETS 6.017 6.028 6.043 6.057 6.059 9.040: 10.009
10.035 10.036 11.018 11.058 11.059 12.008 12.009
RETX 16.052 16.056 16.060: 17.009 17.020

REV 4.051: 6.032

REVIS 3.032= 4.051

SAVA. 6.037= 6.038 12.059 16.043

SAVA1 6.033 6.040:

SAVM1 12.050: 13.015 13.016 13.017 13.018 13.019 13.025

SAVX. 6.038= 12.055

SAVXX 6.042:

SBA 17.025

SETST 7.008: 8.011

SKIPO 5.015

SPACE 4.044: 8.025

SPCF. 5.015

SPLI1 11.035: 11.040

SPLI2 11.037 11.042:

SPLIT 10.038 11.007:

STABL 5.010 7.021:

START 3.040 3.043:

STPTR 8.014 8.024 9.026 10.012 11.050:

STTAB 13.013 14.023:

TABLE 3.048: 3.049 3.055 4.040 6.037 7.024 9.041

TEMP 8.050 9.017 11.052: 12.019 12.022

TEMP1 14.030 14.038 15.049 15.057 16.030 17.033:

TEMPX 13.044 13.051 13.054 14.032 14.045 15.024 15.027
15.032 16.018 16.027 16.048:

TEMPY 12.048: 13.014 13.026

UNITS 10.048 11.045 12.018 12.024 12.035:

VCEND 5.020 5.028:

VSIZE 4.047: 5.030

VTABL 4.047 5.022: 5.023 5.024 5.025 5.026

XEXIT 15.011 15.056 16.017 16.025 16.033 16.044 17.013
17.017 17.019 17.031:

XGETB 16.054

YEAR 8.018 8.044 9.043:

YEARS 14.052 15.037: