

NASA Contractor Report 178361

TRANSFERABLE OUTPUT ASCII DATA (TOAD)
FILE FORMAT DESCRIPTION

Bradford Bingel and Dana Hammond

COMPUTER SCIENCES CORPORATION
Hampton, Virginia

Contract NAS1-17999
September 1987

(NASA-CR-178361) TRANSFERABLE OUTPUT ASCII
DATA (TOAD) FILE FORMAT DESCRIPTION
(Computer Sciences Corp.) 35 p Avail: NTIS
EC A03/BF A01 CSCL 09B

N87-28323

Unclas
G3/61 0097588



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

Preface

This document describes the Transferable Output ASCII Data (TOAD) file format and its use. The introductory section presents the concept and advantages of TOAD to the nontechnical reader. The remaining bulk of the document presents the TOAD format in detail, and is intended for the applications programmer needing a technical description of the TOAD format.

Table of Contents

	page
Introduction	1
The TOAD Format	3
File Characteristics	
Universal Record Format	
Warts	
Types of Warts	
TOAD Standards	7
Guidelines	8
Appendix A - TOAD Utility Software	

PRECEDING PAGE BLANK NOT FILMED

Introduction

TOAD is an acronym for Transferable Output ASCII Data. The TOAD format is a set of standards used when writing a data file, called a TOAD file. The purpose of the TOAD format is to facilitate the transfer of data and software from one computer installation to another.

The concept of TOAD is very simple yet not immediately obvious. Perhaps the best description of the TOAD format is that it is not concerned with what is stored in a TOAD file, only how it is stored. For example, a very simple TOAD file might contain:

```
BEGIN
COUNT      3
LABEL       YEAR      SALES      PROFIT
DATA        1960      30        5
DATA        1970      100       20
DATA        1980      350       50
DATA        1990      1500      300
END
```

Note: Both this example and the one which follows use an oversimplified TOAD format for demonstration purposes only. The actual TOAD format is described in detail beginning on page 3.

Perhaps this represents summarized financial data for a company over a period of 31 years. The "BEGIN" at the top of the file and the "END" at the bottom of the file simply mark the beginning and end of the data. The "COUNT" value of 3 indicates that there are three variables in the data set. The "LABEL" record assigns a name to each of these three variables: "YEAR" to variable #1, "SALES" to variable #2, and "PROFIT" to variable #3. The remaining "DATA" records contain the data corresponding to these variables. For example, in the "YEAR" 1970, "SALES" were 100 units with a "PROFIT" of 20 units.

As another example, another very simple TOAD file might contain:

```
BEGIN
COUNT      4
LABEL       X          Y          TEMP      PRESS
DATA        1          1          10        2
DATA        1          2          12        4
DATA        1          3          30        8
DATA        2          1          20        2
DATA        2          2          30        4
DATA        2          3          50        9
END
```

Perhaps this represents physical data collected over a 2 x 3 grid. Again, "BEGIN" and "END" mark the beginning and end of the file. The "COUNT" of 4 indicates that there are four variables in the data set, with the labels "X", "Y", "TEMP", and "PRESS", respectively. The remaining "DATA" records make up the data set. For example, at "X"=2 and "Y"=3 (presumably a physical location) the "TEMP" is 50 units and the "PRESS" is 9 units (presumably temperature and pressure).

There are two advantages in using the TOAD format. First, TOAD files are of the preferred type and record length to make them easy to edit, read and write on magnetic tape, or transfer across communications networks. Second, applications programs can write TOAD files directly (making their results easy to postprocess) yet still conform to all ANSI FORTRAN 77 standards.

The TOAD Format

File Characteristics

TOAD files are formatted (as opposed to unformatted or binary), sequential-access (as opposed to direct-access, indexed, or keyed) with fixed-length records of 80 characters. There is no limit on the number of records within a single TOAD file. Each and every record conforms to the Universal Record Format.

Universal Record Format

All records within a TOAD file have the same basic format, called the Universal Record Format. Each record is divided into six fields:

field	1	2	3	4	5	6
columns	1-5	6-20	21-35	36-50	51-65	66-80

The first field, also called the "type" field, contains text and has the format A5. The remaining five fields of 15 characters each may contain text or numeric information. For text information, the format is A15. For integer numeric information, the format is I15. For real numeric information, the format is E15.8. Notice that the text information is left-justified within a field, but numeric (integer or real) is right-justified.

Warts

The Universal Record Format allows event information for up to five variables to be stored on a single record. If data for more than five variables is to be stored, groups of multiple records are used. For example, a seven-variable TOAD file might contain:

```
BEGIN
COUNT      7
LABEL      MACH      ALPHA      2Y/B      LOAD      CL
           CD        CM
DATA       .6        10.        .75        .31        1.2
           .43       -0.04
DATA       .6        20.        .75        .53        1.8
           .73       .02
END
```

As another example, a 13-variable TOAD file might contain:

```
BEGIN
COUNT      13
LABEL       TUNNEL   MODEL   JULIAN   ALPHA   RUN
            ROW     2Y/B   PORT     X/C     PRESS
            TEMP    VEL     STRAIN
DATA        21.     118.   83034.   10.     12.
            3.      .3     5.       .45     .67
            62.     .69    .0002
DATA        21.     118.   83034.   20.     13.
            3.      .3     5.       .45     .59
            61.     .63    .0009
END
```

This "wraparound" method is used for as many records as needed to store all of each event's data, and each group of records corresponding to a single event is called a TOAD "wart". For clarity, only the first record in each wart contains the type of information ("LABEL", "DATA", etc.) stored.

Types of Warts

There are two groups of wart types: official and unofficial. The official wart types are:

BEGIN begins a TOAD file. A BEGIN wart is always a single record. The word "BEGIN" appears in field 1 of the universal record format.

END ends a TOAD file. An END wart is always a single record. The word "END" appears in field 1 of the universal record format.

COUNT indicates the number of variables in the TOAD file. A COUNT wart is always a single record. The word "COUNT" appears in field 1. The type integer variable count is right-justified in field 2. Values less than or equal to zero are not allowed.

LABEL contains the labels (names) of the variables in the TOAD file. A LABEL wart may contain one or more records. The number of records is given by:

$$nrecs = \text{int}((nvars+4)/5)$$

where 'nrecs' is the number of records in the LABEL wart, and 'nvars' is the number of variables given in the COUNT wart. The word "LABEL" appears in field 1 of the first record, but may or may not

appear in field 1 of any subsequent records within the same wart. All labels are type character, 1-15 characters long (A15 format preferred), and left-justified in fields 2-6.

UNITS contains the units of the variables in the TOAD file. A UNITS wart may have one or more records. The number of records is given by:

$$\text{nrecs} = \text{int}((\text{nvars}+4)/5)$$

where 'nrecs' is the number of records in the UNITS wart, and 'nvars' is the number of variables given in the COUNT wart. The word "UNITS" appears in field 1 of the first record, but may or may not appear in field 1 of any subsequent records within the same wart. All units are type character, 1-15 characters long (A15 format preferred), and left-justified in fields 2-6.

DATA contains data for a single event. A DATA wart may have one or more records. The number of records is given by:

$$\text{nrecs} = \text{int}((\text{nvars}+4)/5)$$

where 'nrecs' is the number of records in the DATA wart, and 'nvars' is the number of variables given in the COUNT wart. The word "DATA" appears in field 1 of the first record, left-justified, but may or may not appear in field 1 of any subsequent records within the same wart. All data are type real, 1-15 characters long (E15.8 format preferred), and right-justified in fields 2-6.

SKIP allows comments within a TOAD file. A SKIP wart is always a single record. The word "SKIP" appears in field 1, left-justified. Any text is allowed in columns 6-80. SKIP warts may appear anywhere in a TOAD file, with three exceptions:

- never before a BEGIN wart.
- never after an END wart.
- never between two records within another multi-record wart.

There is no list of unofficial wart types. Users with special needs may create new wart types for their particular application, with the understanding that if that TOAD file is transferred to another user, the information contained on unofficial warts will be

ignored. Examples of unofficial warts might be:

ZERO for data acquisition system calibration.

TOL tolerances for the variables.

FLUKE an unusual data event.

TOAD Standards

1. Exactly one BEGIN wart must appear in the TOAD file, and it must be the very first record.
2. Exactly one END wart must appear in the TOAD file, and it must be the very last record.
3. A COUNT wart must appear before any LABEL, UNITS, or DATA warts.
4. No wart may come between two records within another multi-record wart.
5. SKIP warts may appear anywhere in the TOAD file, subject to condition 4.
6. Multiple COUNT warts are allowed. A secondary COUNT wart may change the number of variables (and the number of records) contained in subsequent LABEL, UNITS, and DATA warts.
7. Multiple LABEL warts are allowed. A secondary LABEL wart may change the variable labels, but unless a secondary COUNT wart is used, the new LABEL wart must contain the same number of labels and the same number of records.
8. Multiple UNITS warts are allowed. A secondary UNITS wart may change the variable units, but unless a secondary COUNT wart is used, the new UNITS wart must contain the same number of units and the same number of records.

(TOAD files containing multiple COUNT, LABEL, or UNITS warts are called "complex" TOAD files)
9. Multiple DATA warts are allowed (and expected). Unless secondary COUNT records are used, all DATA warts must contain the same amount of data and use the same number of records.
10. There is no limit on the number of warts or records in a TOAD file.

Guidelines

1. Many simple TOAD files are preferred to a single complex TOAD file (a "complex" TOAD file is one which contains multiple COUNT, LABEL, or UNITS warts).
2. When writing a TOAD file having multi-record warts, writing the wart type on only the first record of each wart makes the file much more readable.
3. If possible, try to avoid writing TOAD files with an excessive number of warts, or with an excessive number of variables. Breaking up an extremely large TOAD file into many smaller TOAD files, even at the expense of duplicating some of the data, will usually quicken access time and save money.
4. Although the "A" in "TOAD" is for ASCII, EBCDIC (or any other character set) works equally well.

Appendix A

TOAD utility software listed in this appendix:

TOADA	selectively accesses data from a TOAD file.
TOADM	generates a menu of the parameters within a TOAD file.
TOADV	verifies that a file is in TOAD format.
VARMAP	sets up a pointer array for subroutine "TOADA".
FORMAT	creates label and data wart formats.
DBLANK	removes all blanks from a text string.

All of the utility software listed in this appendix is available on the NASA / Langley Research Center CDC mainframes. It is also available from COSMIC as item LAR-13755. For further information, contact:

COSMIC
Software Information Services
Computer Services Annex
The University of Georgia
Athens, GA 30602

(404) 542-3265

```
      SUBROUTINE TOADA ( IN, MAXVAR, CWORK, RWORK,  
+                      NUMSEL, NAMESEL, VALSEL, TOLSEL, IWORK,  
+                      XNAME, XMIN, XMAX, YNAME, YMIN, YMAX,  
+                      NCAPAC, XDATA, YDATA, NUSED, IERROR )
```

```
C  
C-----  
C  
C UTILITY TO SELECTIVELY READ DATA FROM A TOAD FILE.  
C  
C THE "TRANSFERABLE OUTPUT ASCII DATA (TOAD) FILE FORMAT DESCRIPTION"  
C IS AVAILABLE FROM BRADFORD BINGEL (CSC) OR DR. JOHN E. LAMAR (NASA).  
C  
C AN ATTEMPT IS MADE TO RETRIEVE QUALIFIED DATA FROM A TOAD FILE AND  
C STORE IT INTO X- AND Y-AXIS DATA ARRAYS. SEARCHING BEGINS AT THE  
C TOP OF THE TOAD FILE AND CONTINUES UNTIL THE DATA ARRAY CAPACITY IS  
C REACHED OR UNTIL THE LAST RECORD IS READ.  
C  
C NOT ALL OF THE AVAILABLE TOAD FILE DATA IS NECESSARILY RETURNED IN  
C THE X- AND Y-AXIS DATA ARRAYS. IF A RECORD CONTAINS AN X-AXIS DATA  
C LESS THAN THE GIVEN X-AXIS MINIMUM OR GREATER THAN THE GIVEN X-AXIS  
C MAXIMUM, IT IS IGNORED. SIMILARLY, A RECORD CONTAINING A Y-AXIS  
C DATA LESS THAN THE Y-AXIS MINIMUM OR GREATER THAN THE Y-AXIS MAXIMUM  
C IS IGNORED. IF SELECTION VARIABLES ARE USED, ALL OF THE SELECTION  
C VARIABLE NAMES MUST BE PRESENT IN THE FILE, AND ALL CORRESPONDING  
C VALUES MUST EQUAL, PLUS OR MINUS A GIVEN TOLERANCE, A GIVEN TARGET  
C VALUE. DATA ON RECORDS NOT MEETING THESE REQUIREMENTS ARE IGNORED.  
C  
C ERRORS ARE INDICATED WITH THE LAST ARGUMENT, 'IERROR'. A LIST OF  
C THE POSSIBLE VALUES AND THEIR MEANINGS IS GIVEN BELOW. EVEN IF  
C 'IERROR' IS RETURNED WITH A VALUE OF ZERO (INDICATING NO ERRORS),  
C IT'S POSSIBLE THAT NO QUALIFYING DATA WAS FOUND. TO BE SAFE,  
C ALWAYS CHECK THE ACTUAL NUMBER OF WORDS STORED IN THE RETURNED  
C DATA ARRAYS ('NUSED').
```

```
C-----  
C  
C ARGUMENTS:  
C  
C INCOMING  
C  
C IN INTEGER I/O UNIT OF THE ALREADY OPEN TOAD FILE.  
C  
C MAXVAR INTEGER NUMBER OF VARIABLES IN THE LONGEST EXPECTED  
C TOAD FILE WART.  
C  
C CWORK CHARACTER*15 WORKSPACE.  
C (MUST BE AT LEAST 'MAXVAR' ELEMENTS LONG)  
C  
C RWORK REAL WORKSPACE.  
C (MUST BE AT LEAST 'MAXVAR' WORDS LONG)  
C  
C NUMSEL INTEGER NUMBER OF VARIABLES TO USE IN SELECTING THE  
C DATA SET. A VALUE OF ZERO INDICATES THAT SELECTION  
C VARIABLES ARE NOT TO BE USED.
```


- C 14 - INPUT FILE IS NOT IN TOAD FILE FORMAT.
- C
- C 15 - MORE DATA MAY BE AVAILABLE THAN WAS STORED.
- C
- C 16 - ONLY THE FIRST SECTION OF THIS COMPLEX TOAD
- C FILE WAS SEARCHED.
- C
- C 21 - IMPROPER VALUE FOR 'MAXVAR'.
- C (MUST BE AT LEAST 2)
- C
- C 22 - 'MAXVAR' IS TOO SMALL.
- C (TOAD FILE WARTS ARE LONGER THAN EXPECTED)
- C
- C 51 - IMPROPER NUMBER OF SELECTION VARIABLES.
- C (CANNOT BE NEGATIVE)
- C
- C 52 - TOO MANY SELECTION VARIABLES.
- C ('NUMSEL' EXCEEDS THE TOAD FILE'S COUNT)
- C
- C 61 - UNABLE TO MATCH A SELECTION VARIABLE'S NAME.
- C
- C 81 - IMPROPER SELECTION VARIABLE TOLERANCE.
- C (ALL MUST BE POSITIVE NONZERO VALUES)
- C
- C 101 - UNABLE TO MATCH THE X-AXIS VARIABLE'S NAME.
- C
- C 111 - IMPROPER X-AXIS MINIMUM AND MAXIMUM.
- C ('XMIN' MUST BE LESS THAT 'XMAX')
- C
- C 131 - UNABLE TO MATCH THE Y-AXIS VARIABLE'S NAME.
- C
- C 141 - IMPROPER Y-AXIS MINIMUM AND MAXIMUM.
- C ('YMIN' MUST BE LESS THAT 'YMAX')
- C
- C 161 - IMPROPER VALUE FOR 'NCAPAC'
- C (MUST BE AT LEAST 1)

C

C WRITTEN APRIL 1986 BRADFORD BINGEL

C COMPUTER SCIENCES CORPORATION

C NASA / LANGLEY RESEARCH CENTER

C

CHARACTER*15	CWORK (MAXVAR)
CHARACTER*15	NAMSEL (NUMSEL)
CHARACTER*15	XNAME, YNAME

C

CHARACTER*80	TEXT80
CHARACTER*15	FIELD (5)
CHARACTER*5	TYPE
EQUIVALENCE	(TEXT80(1:1) , TYPE)
EQUIVALENCE	(TEXT80(6:6) , FIELD(1))

```

C      CHARACTER*40          LFORM,          LFORM2
C      CHARACTER*40          DFORM,          DFORM2
C
C      LOGICAL              OPEN
C
C      INTEGER              IWORK ( NUMSEL )
C
C      REAL                 RWORK ( MAXVAR )
C      REAL                 VALSEL ( NUMSEL )
C      REAL                 TOLSEL ( NUMSEL )
C      REAL                 XDATA ( NCAPAC )
C      REAL                 YDATA ( NCAPAC )
C
C      IERROR = 0
C      NUSED = 0
C
C-----
C
C      DO SOME QUICKS CHECKS ON THE INCOMING ARGUMENTS.
C-----
C
C      IF ( IN.LT.1.OR.IN.GT.99) THEN
C          IERROR = 11
C          RETURN
C      END IF
C
C      IF ( MAXVAR.LT.2) THEN
C          IERROR = 21
C          RETURN
C      END IF
C
C      IF ( NUMSEL.LT.0) THEN
C          IERROR = 51
C          RETURN
C      END IF
C
C      IF ( XMIN.GT.XMAX) THEN
C          IERROR = 111
C          RETURN
C      END IF
C
C      IF ( YMIN.GT.YMAX) THEN
C          IERROR = 141
C          RETURN
C      END IF
C
C      IF ( NCAPAC.LT.1) THEN
C          IERROR = 161
C          RETURN
C      END IF
C-----

```

```

C
C MAKE SURE THAT THE GIVEN I/O UNIT IS OPEN, THEN REWIND THE FILE.
C
C-----
C
C      INQUIRE (IN,EXIST=OPEN)
C
C      IF (.NOT.OPEN) THEN
C          IERROR = 12
C          RETURN
C          END IF
C
C      REWIND IN
C
C-----
C
C      SKIP OVER THE "BEGIN" WART, FIND THE "COUNT" WART, AND GENERATE
C      THE "LABEL" AND "DATA" WART FORMATS.
C
C-----
C
C      READ (IN,1000,END=913,ERR=914) TEXT80
C      1000 FORMAT(A80)
C
C      IF (TYPE.NE.'BEGIN') THEN
C          IERROR = 14
C          RETURN
C          END IF
C
C      100 READ (IN,1000,END=914,ERR=914) TEXT80
C          IF (TYPE.NE.'COUNT') GO TO 100
C
C      READ (FIELD(1),'(I15)',ERR=914) NUMBER
C
C      IF (MAXVAR.LT.NUMBER) THEN
C          IERROR = 22
C          RETURN
C          END IF
C
C      NLEFT = NUMBER - 2
C      IF (NUMSEL.GT.NLEFT) THEN
C          IERROR = 52
C          RETURN
C          END IF
C
C      DO 200 ISEL = 1, NUMSEL
C          IF (TOLSEL(ISEL).LE.0.) THEN
C              IERROR = 81
C              RETURN
C              END IF
C      200 CONTINUE
C
C      CALL FORMAT(NUMBER,LFORM,LFORM2,DFORM,DFORM2)
C

```

```

      NREAD = MIN(5,NUMBER)
C-----
C
C FIND THE "LABEL" WART.
C-----
C
300 READ (IN,1000,END=914,ERR=914) TEXT80
    IF (TYPE.EQ.'SKIP'.OR.TYPE.EQ.'TEXT') GO TO 300
C
    IF (TYPE.NE.'LABEL') THEN
        IERROR = 14
        RETURN
    END IF
C
    READ (TEXT80,LFORM) TYPE,(CWORK(I),I=1,NREAD)
C
    IF (NUMBER.GT.5)
+READ (IN,LFORM2,END=914,ERR=914) (CWORK(I),I=6,NUMBER)
C-----
C
C MAP OUT THE X-AXIS, Y-AXIS, AND ANY SELECTION VARIABLES.
C-----
C
    CALL VARMAP(NUMBER,CWORK,NUMSEL,NAMSEL,IWORK,XNAME,YNAME,
+              IXNAME,IYNAME,IERROR)
C
    IF (IERROR.NE.0) RETURN
C-----
C
C SEARCH THE DATA ON THE REMAINDER OF THE TOAD FILE.
C-----
C
400 READ (IN,1000,END=914,ERR=914) TEXT80
    IF (TYPE.EQ.'SKIP'.OR.TYPE.EQ.'TEXT') GO TO 400
C
    IF (TYPE.EQ.'COUNT'.OR.TYPE.EQ.'LABEL') THEN
        IERROR = 16
        RETURN
    END IF
C
    IF (TYPE.EQ.'END') GO TO 800
C
    IF (TYPE.EQ.'DATA') THEN
C
        READ (TEXT80,DFORM,END=914,ERR=914) TYPE,(RWORK(I),I=1,NREAD)
        IF (NUMBER.GT.5)
+    READ (IN,DFORM2,END=914,ERR=914) (RWORK(I),I=6,NUMBER)
C

```

```

ELSE IF (NUMBER.GT.5) THEN
C
    READ (IN,LFORM2,END=914,ERR=914) (CWORK(I),I=6,NUMBER)
    GO TO 400
C
    END IF
C
-----
C
C SEE IF THIS RECORD'S DATA SHOULD BE KEPT. IF NOT, SKIP TO THE
C NEXT RECORD.
C
-----
C
    DO 500 ISEL = 1, NUMSEL
    DIFF = RWORK(IWORK(ISEL)) - VALSEL(ISEL)
    IF (ABS(DIFF).GT.TOLSEL(ISEL)) GO TO 400
500 CONTINUE
C
    X = RWORK(IXNAME)
    IF (X.LT.XMIN.OR.X.GT.XMAX) GO TO 400
C
    Y = RWORK(IYNAME)
    IF (Y.LT.YMIN.OR.Y.GT.YMAX) GO TO 400
C
-----
C
C STORE THE DATA, AND CONTINUE UNTIL THE DATA ARRAYS ARE FILLED.
C
-----
C
    NUSED = NUSED + 1
    XDATA(NUSED) = X
    YDATA(NUSED) = Y
C
    IF (NUSED.LT.NCAPAC) GO TO 400
    IERROR = 15
C
-----
C
C CONTROL PASSES HERE WHEN THE ENTIRE FILE HAS BEEN SUCCESSFULLY
C READ OR WHEN THE DATA ARRAYS ARE FILLED TO CAPACITY.
C
-----
C
    800 RETURN
C
-----
C
C END/ERR PATHS    END/ERR PATHS    END/ERR PATHS    END/ERR PATHS
C
-----
C
    913 IERROR = 13

```

```
      RETURN  
C     914 IERROR = 14  
      RETURN
```

```
C-----  
C  
C  
      END
```

SUBROUTINE TOADM (IN, MENU, NCAPAC, NFOUND, IERROR)

C
C-----
C
C UTILITY TO MAKE A LIST OF THE PARAMETERS WITHIN THIS TOAD FILE.
C
C THE "TRANSFERABLE OUTPUT ASCII DATA (TOAD) FILE FORMAT DESCRIPTION"
C IS AVAILABLE FROM BRADFORD BINGEL (CSC) OR DR. JOHN E. LAMAR (NASA).
C-----

C
C ARGUMENTS:
C

C IN INTEGER I/O UNIT OF THE ALREADY OPEN TOAD FILE.
C

C MENU CHARACTER*15 ARRAY TO RECEIVE THE PARAMETER LIST.
C

C NCAPAC INTEGER SIZE OF "MENU" IN THE CALLING ROUTINE.
C

C NFOUND RETURNED INTEGER NUMBER OF PARAMETERS FOUND.
C

C IERROR RETURNED INTEGER ERROR INDICATOR:
C

C 0 - NO ERRORS, BUT CHECK "NFOUND" FOR THE ACTUAL
C NUMBER OF PARAMETERS STORED IN "MENU".
C

C 11 - IMPROPER I/O UNIT.
C (MUST BE BETWEEN 1 AND 99)
C

C 12 - I/O UNIT IS NOT CURRENTLY OPEN FOR INPUT.
C

C 13 - INPUT FILE EXISTS BUT IS EMPTY.
C

C 14 - INPUT FILE IS NOT IN TOAD FORMAT.
C

C 31 - IMPROPER MENU ARRAY CAPACITY.
C

C 32 - ISUFFICIENT MENU ARRAY CAPACITY.
C-----

C
C WRITTEN APRIL 1986 BRADFORD BINGEL
C COMPUTER SCIENCES CORPORATION
C NASA / LANGLEY RESEARCH CENTER
C-----

C
C CHARACTER*80 TEXT80
C CHARACTER*15 FIELD (5)
C CHARACTER*5 TYPE
C EQUIVALENCE (TEXT80(1:1) , TYPE)
C EQUIVALENCE (TEXT80(6:6) , FIELD(1))
C
C CHARACTER*15 MENU (NCAPAC)

```

C          CHARACTER*40          LFORM,          LFORM2
          CHARACTER*40          DFORM,          DFORM2
C
C          LOGICAL              OPEN
C
          NFOUND = 0
          IERROR = 0
C
C-----
C
C          DO SOME QUICK CHECKS ON THE INCOMING ARGUMENTS.
C-----
C
          IF (IN.LT.1.OR.IN.GT.99) THEN
              IERROR = 11
              RETURN
          END IF
C
          IF (NCAPAC.LT.2) THEN
              IERROR = 31
              RETURN
          END IF
C
C-----
C
C          MAKE SURE THAT THE GIVEN I/O UNIT IS OPEN, THEN REWIND THE FILE.
C-----
C
          INQUIRE (IN,EXIST=OPEN)
C
          IF (.NOT.OPEN) THEN
              IERROR = 12
              RETURN
          END IF
C
          REWIND IN
C
C-----
C
C          SKIP OVER THE "BEGIN" WART, FIND THE "COUNT" WART, AND GENERATE
C          THE "LABEL" WART FORMATS.
C-----
C
          READ (IN,1000,END=913,ERR=914) TEXT80
          1000 FORMAT(A80)
C
          IF (TYPE.NE.'BEGIN') THEN
              IERROR = 14
              RETURN
          END IF

```

```

C
100 READ (IN,1000,END=914,ERR=914) TEXT80
    IF (TYPE.NE.'COUNT') GO TO 100
C
    READ (FIELD(1),'(I15)',ERR=914) NFOUND
C
    IF (NCAPAC.LT.NFOUND) THEN
        IERROR = 32
        RETURN
    END IF
C
    CALL FORMAT(NFOUND,LFORM,LFORM2,DFORM,DFORM2)
C
    NREAD = MIN(5,NFOUND)
C
C-----
C
C FIND THE "LABEL" WART.
C
C-----
C
200 READ (IN,1000,END=914,ERR=914) TEXT80
    IF (TYPE.EQ.'SKIP'.OR.TYPE.EQ.'TEXT') GO TO 200
C
    IF (TYPE.NE.'LABEL') THEN
        IERROR = 14
        RETURN
    END IF
C
    READ (TEXT80,LFORM) TYPE,(MENU(I),I=1,NREAD)
C
    IF (NFOUND.GT.5)
+READ (IN,LFORM2,END=914,ERR=914) (MENU(I),I=6,NFOUND)
C
    RETURN
C
C-----
C
C END/ERR PATHS    END/ERR PATHS    END/ERR PATHS    END/ERR PATHS
C
C-----
C
913 IERROR = 13
    RETURN
C
914 IERROR = 14
    RETURN
C
C-----
C
    END

```

```

SUBROUTINE TOADV ( IN, MAXVAR, CWORK, RWORK, IERROR )
C
C-----
C
C UTILITY TO VERIFY THAT A FILE IS IN TOAD FORMAT.
C
C THE "TRANSFERABLE OUTPUT ASCII DATA (TOAD) FILE FORMAT DESCRIPTION"
C IS AVAILABLE FROM BRADFORD BINGEL (CSC) OR DR. JOHN E. LAMAR (NASA).
C-----
C
C ARGUMENTS:
C
C   INCOMING
C
C     IN      INTEGER I/O UNIT OF THE ALREADY OPEN TOAD FILE.
C
C     MAXVAR  INTEGER NUMBER OF VARIABLES IN THE LONGEST EXPECTED
C             TOAD FILE WART.
C
C     CWORK   CHARACTER*15 WORKSPACE.
C             (MUST BE AT LEAST 'MAXVAR' ELEMENTS LONG)
C
C     RWORK   REAL WORKSPACE.
C             (MUST BE AT LEAST 'MAXVAR' WORDS LONG)
C
C     IERROR  RETURNED INTEGER ERROR INDICATOR:
C
C             0 - NO ERRORS - THIS IS A VALID TOAD FILE.
C
C             11 - IMPROPER I/O UNIT.
C                  (MUST BE BETWEEN 1 AND 99)
C
C             12 - I/O UNIT IS NOT OPEN FOR INPUT.
C
C             13 - INPUT FILE EXISTS BUT IS EMPTY.
C
C             14 - INPUT FILE IS NOT IN TOAD FILE FORMAT.
C
C             16 - ONLY THE FIRST SECTION OF THIS COMPLEX TOAD
C                  FILE WAS VERIFIED.
C
C             21 - IMPROPER VALUE FOR 'MAXVAR'.
C                  (MUST BE AT LEAST 2)
C
C             22 - 'MAXVAR' IS TOO SMALL.
C                  (TOAD FILE WARTS ARE LONGER THAN EXPECTED)
C-----
C
C WRITTEN APRIL 1986   BRADFORD BINGEL
C COMPUTER SCIENCES CORPORATION

```

```

C  NASA / LANGLEY RESEARCH CENTER
C
C-----
C
C      CHARACTER*15          CWORK ( MAXVAR )
C
C      CHARACTER*80          TEXT80
C      CHARACTER*15          FIELD ( 5 )
C      CHARACTER*5           TYPE
C      EQUIVALENCE           ( TEXT80(1:1) , TYPE )
C      EQUIVALENCE           ( TEXT80(6:6) , FIELD(1) )
C
C      CHARACTER*40          LFORM,          LFORM2
C      CHARACTER*40          DFORM,          DFORM2
C
C      LOGICAL               OPEN
C
C      REAL                  RWORK ( MAXVAR )
C
C      IERROR = 0
C
C-----
C
C      DO SOME QUICKS CHECKS ON THE INCOMING ARGUMENTS.
C
C-----
C
C      IF (IN.LT.1.OR.IN.GT.99) THEN
C          IERROR = 11
C          RETURN
C      END IF
C
C      IF (MAXVAR.LT.2) THEN
C          IERROR = 21
C          RETURN
C      END IF
C
C-----
C
C      MAKE SURE THAT THE GIVEN I/O UNIT IS OPEN, THEN REWIND THE FILE.
C
C-----
C
C      INQUIRE (IN,EXIST=OPEN)
C
C      IF (.NOT.OPEN) THEN
C          IERROR = 12
C          RETURN
C      END IF
C
C      REWIND IN
C
C-----
C

```

```

C SKIP OVER THE "BEGIN" WART, FIND THE "COUNT" WART, AND GENERATE
C THE "LABEL" AND "DATA" WART FORMATS.
C -----
C
C READ (IN,1000,END=913,ERR=914) TEXT80
1000 FORMAT(A80)
C
C IF (TYPE.NE.'BEGIN') THEN
C IERROR = 14
C RETURN
C END IF
C
C 100 READ (IN,1000,END=914,ERR=914) TEXT80
C IF (TYPE.NE.'COUNT') GO TO 100
C
C READ (FIELD(1),'(I15)',ERR=914) NUMBER
C
C IF (MAXVAR.LT.NUMBER) THEN
C IERROR = 22
C RETURN
C END IF
C
C CALL FORMAT(NUMBER,LFORM,LFORM2,DFORM,DFORM2)
C
C NREAD = MIN(5,NUMBER)
C -----
C
C FIND THE "LABEL" WART.
C -----
C
C 300 READ (IN,1000,END=914,ERR=914) TEXT80
C IF (TYPE.EQ.'SKIP'.OR.TYPE.EQ.'TEXT') GO TO 300
C
C IF (TYPE.NE.'LABEL') THEN
C IERROR = 14
C RETURN
C END IF
C
C READ (TEXT80,LFORM) TYPE,(CWORK(I),I=1,NREAD)
C
C IF (NUMBER.GT.5)
C +READ (IN,LFORM2,END=914,ERR=914) (CWORK(I),I=6,NUMBER)
C -----
C
C VERIFY THE DATA ON THE REMAINDER OF THE TOAD FILE.
C -----
C
C 400 READ (IN,1000,END=914,ERR=914) TEXT80
C IF (TYPE.EQ.'SKIP'.OR.TYPE.EQ.'TEXT') GO TO 400

```

```

C
  IF (TYPE.EQ.'COUNT'.OR.TYPE.EQ.'LABEL') THEN
    IERROR = 16
    RETURN
  END IF
C
  IF (TYPE.EQ.'END') GO TO 800
C
  IF (TYPE.EQ.'DATA') THEN
C
    READ (TEXT80,DFORM,END=914,ERR=914) TYPE,(RWORK(I),I=1,NREAD)
    IF (NUMBER.GT.5)
+   READ (IN,DFORM2,END=914,ERR=914) (RWORK(I),I=6,NUMBER)
C
    ELSE IF (NUMBER.GT.5) THEN
C
      READ (IN,LFORM2,END=914,ERR=914) (CWORK(I),I=6,NUMBER)
C
    END IF
C
    GO TO 400
C
-----
C
C CONTROL PASSES HERE WHEN THE ENTIRE FILE HAS BEEN SUCCESSFULLY
C READ.
C
-----
C
  800 RETURN
C
-----
C
C END/ERR PATHS   END/ERR PATHS   END/ERR PATHS   END/ERR PATHS
C
-----
C
  913 IERROR = 13
    RETURN
C
  914 IERROR = 14
    RETURN
C
-----
C
  END

```

```

SUBROUTINE VARMAP ( NUMBER, CWORK, NUMSEL, NAMSEL, IWORK,
+                 XNAME, YNAME, IXNAME, IYNAME, IERROR )
C
C-----
C
C GENERATE A VARIABLE MAP.
C-----
C
C WRITTEN APRIL 1986   BRADFORD BINGEL
C COMPUTER SCIENCES CORPORATION
C NASA / LANGLEY RESEARCH CENTER
C-----
C
C CHARACTER*(*)          CWORK ( NUMBER )
C CHARACTER*(*)          NAMSEL ( NUMSEL )
C CHARACTER*(*)          XNAME,          YNAME
C
C INTEGER                IWORK ( NUMSEL )
C-----
C
C MAP THE X-AXIS VARIABLE.
C-----
C
C DO 100 INDEX = 1, NUMBER
C
C IF (CWORK(INDEX).EQ.XNAME) THEN
C   IXNAME = INDEX
C   GO TO 150
C END IF
C
C 100 CONTINUE
C
C IERROR = 101
C RETURN
C
C 150 CONTINUE
C-----
C
C MAP THE Y-AXIS VARIABLE.
C-----
C
C DO 200 INDEX = 1, NUMBER
C
C IF (CWORK(INDEX).EQ.YNAME) THEN
C   IYNAME = INDEX
C   GO TO 250
C END IF
C

```

```
200 CONTINUE
C
  IERROR = 131
  RETURN
C
250 CONTINUE
C
C-----
C
C MAP THE SELECTION VARIABLES.
C
C-----
C
  DO 400 ISEL = 1, NUMSEL
C
  DO 300 INDEX = 1, NUMBER
C
  IF (CWORK(INDEX).EQ.NAMSEL(ISEL)) THEN
    IWORK(ISEL) = INDEX
    GO TO 400
  END IF
C
300 CONTINUE
C
  IERROR = 61
  RETURN
C
400 CONTINUE
C
C-----
C
  RETURN
  END
```

```

SUBROUTINE FORMAT ( NUMBER, LFORM, LFORM2, DFORM, DFORM2 )
C
C-----
C
C UTILITY ROUTINE TO GENERATE TOAD "LABEL" AND "DATA" WART FORMATS.
C-----
C
C ARGUMENTS:
C
C     NUMBER    INTEGER NUMBER OF ITEMS PER "WART".
C
C     LFORM     RETURNED CHARACTER*40 PRIMARY "LABEL" FORMAT.
C
C     LFORM2    RETURNED CHARACTER*40 SECONDARY "LABEL" FORMAT.
C
C     DFORM     RETURNED CHARACTER*40 PRIMARY "DATA" FORMAT.
C
C     DFORM2    RETURNED CHARACTER*40 SECONDARY "DATA" FORMAT.
C-----
C
C WRITTEN OCTOBER 1984  BRADFORD BINGEL
C COMPUTER SCIENCES CORPORATION
C NASA / LANGLEY RESEARCH CENTER
C-----
C
C     CHARACTER*40          LFORM,          LFORM2
C     CHARACTER*40          DFORM,          DFORM2
C-----
C
C PRIMARY FORMATS.
C-----
C
C     NREP = INT ( ( NUMBER + 4 ) / 5 ) - 1
C
C     WRITE (LFORM,1100) NREP
1100 FORMAT(' (A5,5A15,',I5,' (:,/,5X,5A15)')
C
C     LFORM(40:40) = ' )'
C
C     CALL DBLANK(LFORM,40,JUNK)
C
C     WRITE (DFORM,1200) NREP
1200 FORMAT(' (A5,5E15.8,',I5,' (:,/,5X,5E15.8)')
C
C     DFORM(40:40) = ' )'
C
C     CALL DBLANK(DFORM,40,JUNK)
C-----

```

```
C -  
C SECONDARY FORMATS. -  
C -  
C -----  
C -  
C WRITE (LFORM2,1300) NREP  
1300 FORMAT('(',I5,'(5X,5A15,;,/)')  
C LFORM2(40:40) = ')'  
C CALL DBLANK(LFORM2,40,JUNK)  
C  
C WRITE (DFORM2,1400) NREP  
1400 FORMAT('(',I5,'(5X,5E15.8,;,/)')  
C DFORM2(40:40) = ')'  
C CALL DBLANK(DFORM2,40,JUNK)  
C -  
C -----  
C  
C RETURN  
C END
```

SUBROUTINE DBLANK (TEXT, OLDLEN, NEWLEN)

```
C
C-----
C
C REMOVE ALL OF THE BLANKS FROM A TEXT STRING.
C
C FOR EXAMPLE:
C
C   ' A BC  DEF G  H    I J '
C   ** *  ***   * ***  ***** ** *
C
C   * MARKS EVERY BLANK WHICH WILL BE REMOVED.
C
C THE NEW TEXT STRING WILL BE:
C
C   'ABCDEFGHIJ'
C-----
C
C ARGUMENTS:
C
C   TEXT      CHARACTER VARIABLE CONTAINING THE TEXT TO BE
C             DEBLANKED.
C
C   OLDLEN    INTEGER LENGTH OF THE ORIGINAL TEXT STRING.
C
C   NEWLEN    RETURNED INTEGER LENGTH OF THE NEW TEXT STRING.
C-----
C
C WRITTEN JULY 1983  BRADFORD BINGEL
C COMPUTER SCIENCES CORPORATION
C NASA / LANGLEY RESEARCH CENTER
C-----
C
C CHARACTER*(*)      TEXT
C LOGICAL            OFFSET
C INTEGER            OLDLEN,
C INTEGER            COLUMN,      NEWLEN
C                                     INDEX,      ZAPPED
C-----
C
C INDEX = 0
C ZAPPED = 0
C OFFSET = .FALSE.
C-----
C
C LOOK AT EACH CHARACTER IN THE TEXT STRING.  WHEN A BLANK IS FOUND,
C ELIMINATE IT.
C-----
C
```

```

DO 100 COLUMN = 1, OLDLEN
C
IF (TEXT(COLUMN: COLUMN).EQ.' ') THEN
    ZAPPED = ZAPPED + 1
    OFFSET = .TRUE.
ELSE
    INDEX = INDEX + 1
    IF (OFFSET) TEXT(INDEX: INDEX) = TEXT(COLUMN: COLUMN)
END IF
C
100 CONTINUE
C
C-----
C
C DETERMINE THE LENGTH OF THE NEW TEXT STRING, AND BLANK FILL ANY
C RIGHT-HAND CHARACTER POSITIONS LOST FROM SHIFTING.
C
C-----
C
NEWLEN = OLDLEN - ZAPPED
C
IF (OFFSET) TEXT(NEWLEN+1: OLDLEN) = ' '
C
C-----
C
RETURN
END

```

1. Report No. NASA CR-178361	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Transferable Output ASCII Data (TOAD) File Format Description		5. Report Date September 1987	
		6. Performing Organization Code	
7. Author(s) Bradford Bingel and Dana Hammond		8. Performing Organization Report No.	
		10. Work Unit No. 505-60-21-02	
9. Performing Organization Name and Address Computer Sciences Corporation Applied Technology Division 3217 N. Armistead Avenue Hampton, VA 23666		11. Contract or Grant No. NAS1-17999	
		13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code	
		15. Supplementary Notes Langley Technical Monitor: Neal T. Frink	
16. Abstract This report describes a format for writing ASCII data on a file to facilitate its transfer from one computer system to another. The TOAD format conforms to all ANSI FORTRAN 77 standards. There are two advantages in using the TOAD format. First, TOAD files are of the preferred type and record length to make them easy to edit, read from and write on magnetic tape, or transfer across communications networks. Secondly, application programs, using the TOAD format to write computational results, are more portable and the answer files easier to postprocess. TOAD utility software is listed in an appendix.			
17. Key Words (Suggested by Author(s)) ASCII data Data format Data transfer		18. Distribution Statement Unclassified - Unlimited Subject Category 61	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 36	22. Price A03