



TRS-80TM MODEL I



Profile

CUSTOM MANUFACTURED IN USA BY RADIO SHACK  A DIVISION OF TANDY CORPORATION

Cat. No.
26-1562

Radio Shack[®]

TRS-80

MODEL III
MICRO
COMPUTER

© (P) TANDY CORP., 1980.
All Rights Reserved

Cat. No. 26-1562

Profile

DISK 3

TRSDOS OPERATING SYSTEM

© TANDY CORP. 1980

Radio Shack

TRS-80

**MICRO
COMPUTER
SYSTEM**

© © TANDY CORP. 1980

Cat. No. 26-1562

Profile

Program Diskette

TRSDOS OPERATING SYSTEM
© TANDY CORP., 1979

MINIDISK

Radio Shack

TRS-80

**MICRO
COMPUTER
SYSTEM**

PROFILE

Catalog Number 26-1562

The logo is contained within a white rectangular border. Inside this border is a smaller, rounded rectangular area with a light gray background. The text "Radio Shack®" is at the top in a bold, sans-serif font. Below it, "TRS-80" is written in a smaller, bold, sans-serif font. Underneath that, "MICRO" is written in a bold, sans-serif font. At the bottom, "COMPUTER SYSTEM" is written in a bold, sans-serif font, with "COMPUTER" on one line and "SYSTEM" on the line below it.

Radio Shack®
TRS-80
MICRO
COMPUTER
SYSTEM

Profile is a program designed to store information of any kind. Since it is a computerized filing system, you may use it for anything at all. Profile can keep track of names and addresses, accounts, records, or even personal references. Specialized businesses could find many different uses:

1. List of Customer Accounts
2. Personnel Records
3. Prospective Employees or Accounts
4. General Information Referral

Hardware Requirements:

16K Level II Computer

Expansion Interface

One Disk Drive (more may be used for greater storage capacity)

Printer (optional)

This Page Is Intentionally Blank

PROFILE

Radio Shack

 A DIVISION OF TANDY CORPORATION
FORT WORTH, TEXAS 76102

Second Edition — 1979

All rights reserved. Reproduction or use, without express permission, of editorial or pictorial contents, in any manner, is prohibited. While every precaution has been taken in the preparation of this manual, Tandy Corp. assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

© Copyright 1979, Radio Shack
A Division of Tandy Corporation,
Fort Worth, Texas 76102, U.S.A.

Table of Contents

Introduction	1
Features	1
Some Things You Must Know Before Using Profile	3
Setting Up Your Form	5
The Computer's Table of Contents	9
Adding Data	11
Sorting Records	13
Editing Records	15
Printing Records	17
Ending the Program	21
Starting All Over	21
Appendices	
1 – How to Format Your Data Diskettes	23
2 – How to Back Up Your Diskettes	24
3 – Using Disk Basic with Profile	25
4 – The Profile Control File	26
5 – Random File Example	28
6 – Example Using Sequential Access	29

This Page Is Intentionally Blank

Introduction

Profile is designed to store information of any kind. Since it is a computerized filing system, you may use it for anything at all. Profile can keep track of names and addresses, accounts, records, or even personal references. Specialized businesses could find many different uses:

1. List of customers or accounts.
2. Personnel records.
3. Prospective employees or accounts.
4. General information referral.

Profile can also serve many personal uses. It can be used as an address book, recipe file, or grocery inventory. You can even create your own personalized encyclopedia. Because of the program's versatility, the possible uses are endless.

This manual has been written to guide you step-by-step through Profile's capabilities. We will give you examples that will give you a feel for the system. Once you understand the simple methods of the program's operation, there are no limits to Profile's applications.

Features

1. User-created form for records.
2. Up to four Drives may be used to store data.
3. Easy editing features.
4. Versatile printing operations.
5. Repeating keys for easy typing.
6. Sorting by any item in ascending or descending order.
7. Easily retrieves specified groups of records.
8. Data files can be read by DISK BASIC.

This Page Is Intentionally Blank

Some Things You Must Know Before Using Profile

Before you begin running the program, there are a few terms that should be defined:

- Record —** A record is like a piece of paper with information on it. It is, literally, a unit of data, placed in a specified form.
- File —** A file is a place where data is kept. If a record is like a piece of paper with information, then a file is like a folder that holds many such pieces of paper.
- Field —** A field is a movable line that contains information. Think of a field as a line on the paper that will be filled with information, a blank to be filled in.
- Field Name —** This is the name of a line. It is simply an identifier.
- Field Data —** This is the information that will go into a specific line or “field”.

Note: You cannot use the Lower Case Driver to enter data or create forms using Profile.

This Page Is Intentionally Blank

Setting Up Your Form

Diskettes can be damaged through contact with magnetic fields, mishandling, etc. To insure against loss of your program, before you begin running Profile, you should make a copy (Backup) of your Program Diskette, and file it in a safe place. If you are using more than one Drive, you must also prepare (Format) a blank diskette(s) which will be used to store your data. If you are not familiar with Format and Backup functions, refer to Appendices 1 and 2 in the back of this manual for detailed instructions.

Follow these steps in exact order:

1. Turn on the Disk Drives, Printer, Video Display, and the Expansion Interface.
2. Insert the Backup copy of Profile in Drive #0. (This diskette should not have a tab over the write protect notch.)
3. Insert your Data diskette(s) in Drive(s) #1-3 (if used).
4. Turn on the TRS-80 keyboard.

The screen will show:

TRSDOS DISK OPERATING SYSTEM VER 2.3

DOS READY

You type:

PROFILE and press **ENTER** .

The screen will show:

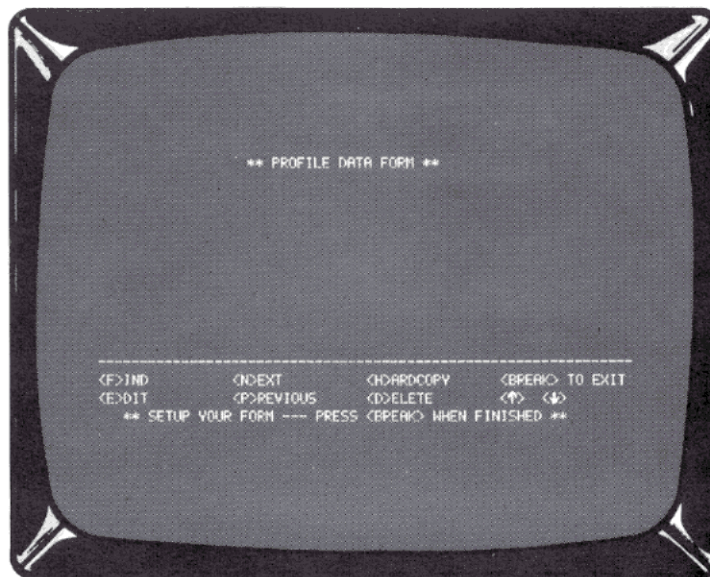
* *INITIALIZATION * *

HOW MANY DRIVES (1-4)?_

Press the number of Drives you will be using.

Setting Up Your Form (continued)

The screen will show :



Notice the blinking cursor, near the upper left-hand corner; this is what you will use to organize what you write on the screen. Type your name on the keyboard. Notice how the cursor moves along the screen as your name is printed. Press the **↵** key and you will see that the cursor moves to the right. If you keep the **↵** key held down, the cursor will keep moving. Now use the **←** arrow to move the cursor to the left. Any of the arrows, if held down, will keep the cursor moving in that direction until they are released.

~~Use the arrows to move the cursor to the first letter of your name. Depress the **SHIFT** key and press **␣**. Your name moved to the right on the screen because you just Inserted a space. Depress the **SHIFT** key again and hold down **␣**, so that your name moves to the center of the screen. Now hold down the **SHIFT** key and press **␣**. This will Delete spaces between the cursor and your name. Keep the **SHIFT****␣** held down until your name is deleted.~~

All of the keys are repeating keys. Press any key and keep it held down. See how the key repeats on the screen? Now go back and delete the string of characters you have just typed, using the arrows, and **SHIFT****␣**.

Before we go on, you might want to practice using these tools — the **↵** and **←** keys, **SHIFT****␣** and **␣**, and the repeating keys, just so you get used to using them. After you have finished, delete all of the characters you have printed, by pressing **CLEAR**. The **CLEAR** key is always used to restore the original contents of the screen.

Now you are ready to set up the form — the way you want your records to appear. We will give you an example. You can do your own records in any manner you wish. For practice, though, follow along with us.

For our sample, we will do a list of club members, their addresses, etc. We will use seven fields (remember — a place for storing information): Member, Address, City, State, Birthdate, Dues Per Year, and Dues Paid.

Note: Remember you can use whatever fields you want when you create your own data file.

Bring the cursor to the first line after the title line. Type the first field name **MEMBER**. Immediately after the word, place a colon **:** and a space. Then hold down the period key **.** to make a string of periods. The number of periods determine how many characters the program will accept for a member's name. This is the field space itself — it is here that you will eventually place your field data. Be sure to make the line long enough to fit everything you will need. (Later on, you will enter the actual field data — the person's name.)

Note: You can have several fields on the same line. You must have continuous periods with at least one space between each field. The maximum number of fields allowed for one record is 32.

The whole thing should look like this:

MEMBER:

Once you have finished, press **ENTER**. This will take you to the next line on the screen.

Now type in the other fields in the same manner. For state you will need only two periods. For Birthdate, it is usually better to put **MM/DD/YY** (month, day, year) instead of the periods. This makes it easier to read. Remember the colon and the space after each field name. For Dues Per Year and Dues Paid, you might want to use **XXX.XX**, for the amount. (We used periods before, because they are easy to see, but any character can be used.)

You can now arrange the lines to make them look neater. Use the **SHIFT** **I** and **D** to line up the colons.

If you wish, you may change the title in the same manner. Move the cursor to the beginning of the title **** PROFILE DATA FORM **** and type in the new title. Now use the **SHIFT** **I** and **D** to adjust the new title to the middle of the screen. The top line is used for title only. Do not try to enter fields on the first line or they will be ignored.

Setting Up Your Form (continued)

The screen should look something like this:

```

      ** CLUB MEMBERS **
MEMBER: .....
ADDRESS: .....
CITY: .....
STATE: .....
BIRTHDATE: MM/DD/YY
DUES PER YEAR: XXX XX
DUES PAID: XXX XX

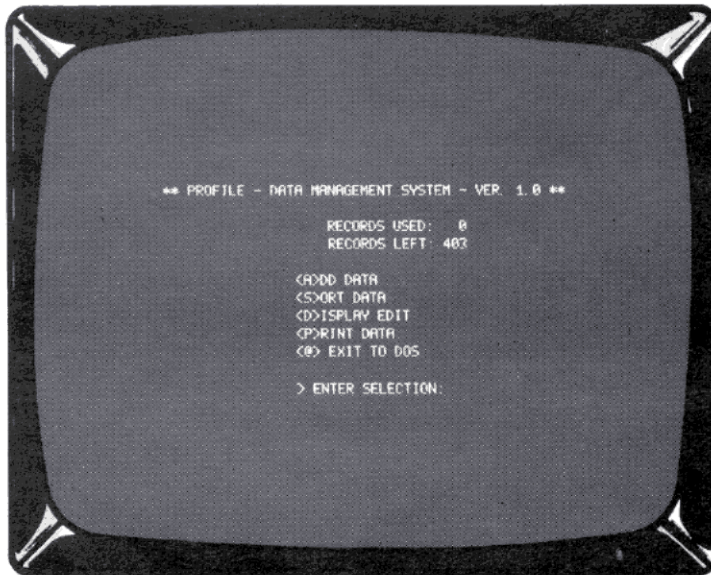
-----
<F>IND      <D>EXT      <D>ARCOPY      <BREAK> TO EXIT
<E>DIT      <P>REVIEW    <D>ELETE      <P>  <D>
** SETUP YOUR FORM --- PRESS <BREAK> WHEN FINISHED **

```

Press the **BREAK** key. The computer will now store the form you have just typed. When you add records (the actual names, etc.) this is the form that will be used. You only have to do this once, until you want to change the form of your entries.

The Computer's Table of Contents

The computer's Table of Contents is called a "Main Menu". This is how it appears on your screen:



This is what each function does:

- | | |
|-----------------|---|
| (A)DD DATA | This is where you will enter the field data (names, etc.) for your "records". |
| (S)ORT DATA | This option allows you to put your records in alphabetical or numerical order, ascending or descending, by specified field. |
| (D)ISPLAY/EDIT | This will allow you to find any of your records, as well as change, or delete them. |
| (P)RINT DATA | This will print your records, completely or partially, depending on which you choose. |
| (@) EXIT TO DOS | This lets you stop the program at any time. |

This Page Is Intentionally Blank

Adding Data

Press **A** at the Main Menu for Add Data.

This is where you fill in all the information for your records. Notice that the blinking cursor is now located at the beginning of your first field. Go ahead and type in your name. The name you just typed in is called the field data. After you have typed in the name, press **ENTER**. The cursor will move to the next field.

There are a few other editing features we should mention here. Pressing the **CLEAR** key erases all field data and allows you to start over again. If you would like to erase the information in only one field, hold down the **SHIFT** key and press **←** key. The **↑** and **↓** keys can be used to move between fields, allowing you to make corrections.

Fill in the rest of the fields in the record, in the same manner. When you finish, you will see the question, ARE ALL ENTRIES CORRECT (Y/N)?_. If they are correct, press **Y**. If you press **N**, the cursor will return to the first field. Once you press **Y**, the form (record) is stored, and you may begin filling in a new record.

Now go ahead and do several more. Use any names, addresses, cities, etc. that you like. This is only for practice. Fill in about ten records. When you have finished, press the **BREAK** key. You will return to the Main Menu.

The screen will show:

RECORDS USED:

RECORDS LEFT:

When you add names, the RECORDS USED will increase, while the RECORDS LEFT will decrease. This lets you keep track of how much information you have entered and how many more records you may add before you fill up your disk.

This Page Is Intentionally Blank

Sorting Records

Press **[S]** at the Main Menu for Sort Data.

The screen will show:

```
SORT ACCORDING TO WHICH FIELD?
NAME OF FIELD: _ _ _ _ _
```

The computer will automatically place your records in alphabetical or numerical order, according to whichever field you specify. Pick a field to Sort by. You can sort by Member, Address, City, State, etc. Type in one of these items and press **[ENTER]**.

The question SORT IN (A)SCENDING OR (D)ESCENDING ORDER (A/D)?_ will appear. This simply means, "do you want to start with A and go to Z, or start with Z and go to A?". Press **[A]** or **[D]**.

The computer will do a number of passes (depending on how many records you have) through the records, refining the order on each pass. The screen will show the number of passes that remain before the computer has completed. If you have a large number of records, this may take some time. When the sorting has finished, you will automatically return to the Main Menu. Your records are now in the order specified.

A word or two should be said about sorting dates and number values or prices.

Enter your dates with the information most critical to you in front. If you wish to sort by the months, a date format of **MM/DD/YY** is best. But if you wish to sort item into chronological order you would use a date format of **YY/MM/DD** or possibly **YYYY/MM/DD** if the century is important. If you need to sort by both the month and the year, then you must have them be individual fields:

```
MONTH: . .      DAY: . .      YEAR: . .
YEAR: . .      MONTH: . .      DAY: . .
```

In order to sort and find numbers and prices correctly you will need to be sure that the decimal point (either actual or implied) is always in the same place. If you had the field **\$\$\$\$.cc** and entered the values **234.34** and **734.57**, the sort would place 734.57 first, because in one of the fields, the blank comes before the 7. To keep this from happening, you must enter **234.34** and **734.57**. This always puts the decimal point as the fifth character.

A 23__ is greater than a __99 so enter them as __23 and __99. This way you "imply" that a decimal point follows the field as the fifth character.

Press the **[BREAK]** key to return to the Main Menu.

This Page Is Intentionally Blank

Editing Records

Press **[D]** at the Main Menu for Display/Edit.

At the bottom of the screen you will see that there are a number of options that you may use to display, edit, or delete any of the information on file.

Options

(F)IND	Lets you find a specific record or group of records. Press [F] , and then you will be asked which record(s) you wish to see.
(N)EXT	After you Find a group of records, this option lets you step through the records, one at a time.
(P)REVIOUS	The same function as Next, but in reverse. It lets you step backwards, from the current position, through the records.
(E)DIT	Allows you to change the field data in a particular record.
(D)ELETE	Allows you to erase a particular record in the file.
(H)ARDCOPY	This will let you have a printout of any particular record in the file.
(↑) and (↓)	These arrows move you from the first record in the file to the last, and last to first, respectively.
[BREAK]	Pressing the [BREAK] key will return you to the Main Menu.

The Find option uses a simple formula to find records quickly:

FIND

(everything that is stored)

(that has a relationship to)

(something).

In other words, the computer looks for certain records that have a field that contains field data equal to, less than, greater than, etc., a particular field data.

Editing Records (continued)

If you wanted to find a member's name beginning with a B, the method would be: NAME OF FIELD: **M E M B E R**

RELATION: **E Q** (for Equal To)
SEARCH FOR: **B**

All Relations use two-letter symbols:

EQ — equal to; NE — not equal to; GT — greater than; LT — less than;
GE — greater than, or equal to; LE — less than, or equal to.

To find names beginning with C, and everything greater than C, you would say, **G E** (greater than, or equal to) **C**. If you wanted to find all of the records from A-Z, you type something like **N E** (not equal to) **@**. This would give you every record on the file, from A to Z.

Press **F** for Find. Type in the field name, then the relation. Do several of these, so that you understand them. Press **N** for Next and go through them. Use **P** for Previous to go back.

Now we are going to edit a record. You edit records, one at a time. Press **E** for Edit. The cursor will be at the beginning of the first field. Change the data by typing in new information, and press **ENTER**. The cursor has moved to the second field. Press **ENTER** and the information will remain the same. Press **ENTER** for every other field in the record. You can use all of the editing features we have talked about earlier.

Now we are going to delete a record. As in editing, you only delete one record at a time, the one on the screen. Press **D** for Delete.

The screen will show:

ARE YOU SURE (Y/N)?_

Press **Y**. The record will now be deleted.

If you would like a hardcopy printout of the records, first Find the record you want to print. When the record you want is on the screen, simply press **H** for Hardcopy.

Press **BREAK**. You will return to the Main Menu.

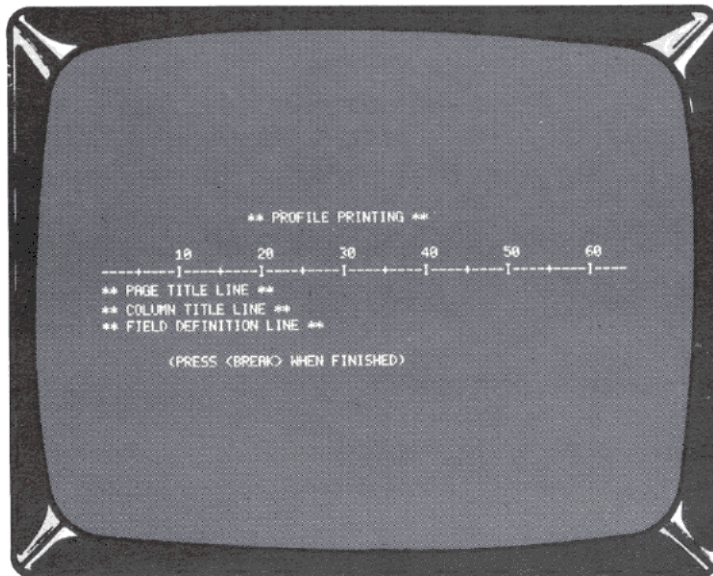
Note: Records counter will not be corrected for the deletion until you sort.

Printing Records

Just as you set up the form for your records, you may design your own form for printing.

Press **[P]** for Print.

The screen will show:



This is a graph that represents the paper on the line printer. There are 132 characters spaces on which you can print. You can see that there are three lines that you may use. The first line is the title line. The second line is the column name line. The third line is the field data line.

You may be thinking that the screen is not showing 132 characters. Press the **[→]** key and follow it across the screen. See how it shifts. Once you reach the end, press the **[←]** key and go back to the beginning of the line.

The first line is the title line. You can call the printout anything you wish. You simply write with the cursor, as you did with setting up the form. Type **[C][L][U][B][M][E][M][B][E][R][S]** in the Page Title Line.

The second line is the label you are using for the fields. You may use as many of the fields as you wish. You must be careful to space them properly, as the field data should fit underneath them. Put all seven fields on the screen.

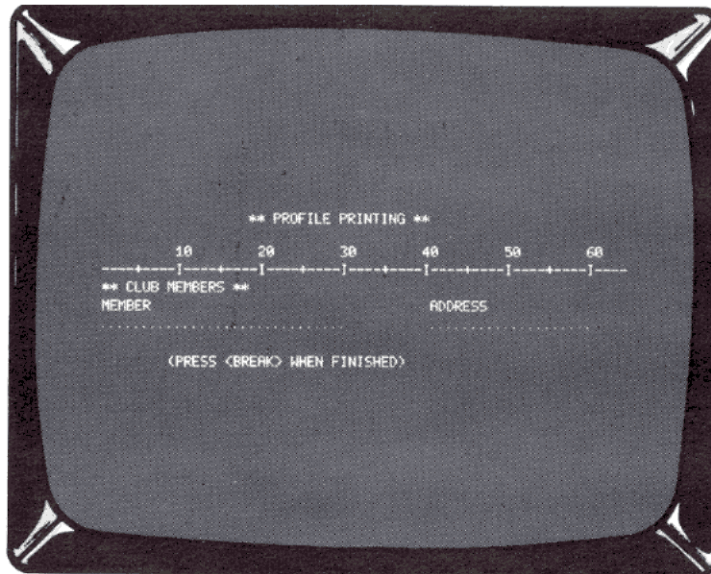
The third line simply needs periods as in the field space itself.

Put as many periods as you will need for each field.

Printing Records (continued)

If there is not enough room for the fields, you may adjust the names. Use the **SHIFT** **I** and **D**.

Your printer form should look something like this:



```

      ** PROFILE PRINTING **
      10      20      30      40      50      60
      -----
      ** CLUB MEMBERS **
      MEMBER                                ADDRESS
      .....
      (PRESS <BREAK> WHEN FINISHED)

```

Press **BREAK**.

The screen will show:

ARE ALL CORRECT (Y/N)?_ Press **Y**.

The computer will ask for the Field Names. Give each name as it appears on the records. The computer places the field it wishes you to name at the left margin and asks you to give the field name of the data you wish to be placed in each field. The column titles need not correspond to the actual field names.

You will now be asked how you want to print. You will be given the same kind of formula as for Find:

NAME OF FIELD:

RELATION:

SEARCH FOR:

Printing will begin immediately. In order to stop, hold down the @key. The printer will move up to the next page and stop. You will now return to the formula again and may print in a different manner. If you allow it to complete the printing, it will move up to the next page and return to the printing criteria. Press **BREAK** and you will return to the Main Menu.

This Page Is Intentionally Blank

Ending the Program

Once you have finished, press **Alt+F4** to exit. You will now be in DOS.

Starting All Over

Now you will want to clear it all out and start over with your own records and information. Once you are in DOS, look at the Directory by typing **DIR** and pressing **ENTER**. You will see listed four data files:

PRODAT	The large file containing all the data.
INFOFILE	The file containing the fields.
FORMFILE	The file containing the form for the records.
LPFORM	The file containing the form for your printing.

In order to start over, you must kill each of these files. Type:

KILL PRODAT and press **ENTER**.

KILL INFOFILE and press **ENTER**.

KILL FORMFILE and press **ENTER**.

KILL LPFORM and press **ENTER**.

Now type **DIR** and press **ENTER** — the files are gone. Once again, you have a clean slate and may start over with your own records.

This Page Is Intentionally Blank

Appendix 1 – How to Format Your Data Diskettes

This process prepares blank diskettes for use on the disk system. All data diskettes must be formatted before being used. Here's how you do it:

1 — Insert a blank diskette in Drive #1

2 — Do the following exactly as shown:

Screen Shows:

DOS READY
WHICH DRIVE etc.
DISKETTE NAME?
CREATION DATE?
MASTER PASSWORD
LOCK OUT ANY TRACKS?
HIT "ENTER" TO CONTINUE

You Type:

FORMAT

1

DATA80

01/01/80

PASSWORD

NO

press **ENTER**

press **ENTER**

press **ENTER**

press **ENTER**

press **ENTER**

press **ENTER**

press **ENTER**

Appendix 2 — How to Back Up Your Diskettes

Use this procedure **EXACTLY**:

- 1 — Turn on everything except the TRS-80 keyboard. If this is the first time you've ever used the Radio Shack Disk System, refer to the Disk Operating System Manual for detailed instructions.
- 2 — Insert a new, blank diskette in Drive #1 (farthest from the Interface) with the diskette notch up, label to the right.
- 3 — Close the Drive #1 Door.
- 4 — Insert the diskette you wish to copy in Drive #0 with the notch up, label facing right, and close the door. (Drive #0 is the first drive connected to the interface.)
- 5 — Turn on the TRS-80 Keyboard. (The switch on the right rear apron.)

This is what the screen shows:

DOS READY
SOURCE DRIVE NUMBER?
DESTINATION DRIVE NUMBER?
BACKUP DATE (MM/DD/YY)?

HIT 'ENTER' TO CONTINUE

This is what you type:

B A C K U P , and press **ENTER**
0 , and press
1 , and press
0 1 0 1 0 8 0 ENTER
(Example for January 1, 1980
STOP — Don't press anything!!!)

Now we can check to see if the "BACUP" procedure was successful:

- 1 — Remove the original diskette from Drive #0.
- 2 — Take the backup diskette out of Drive #1. Place the backup copy in Drive #0 and close the door.
- 3 — Now you can press **ENTER**. If the screen shows: DOS READY, your backup was successful.

If something went wrong, go back to step 1.

Using Disk Basic with Profile

If you are experienced in using DISK BASIC and disk files, you will be happy to know that DISK BASIC programs can be written to read the data files created by Profile. The Profile program was not designed to perform any arithmetic on numeric data you enter. DISK BASIC, however, can be used to calculate billing, compare dates, and perform other processing on the data which Profile creates.

Profile was designed so that its data files may easily be read by DISK BASIC. Profile always stores data as fixed records with a single carriage return following the record. Thus the data created by Profile, can be read as a sequential or a random file. Deleted records are filled with X'CO' characters (decimal 192), and should be ignored if found.

When using random files, you should create your data form so that the amount of data is one less than a number which is evenly divisible into 256. (The numbers 3, 7, 15, 31, 63, 127 and 255 are the only choices.) Since Profile always adds a carriage return to the end of each record, the actual logical record length will be 4, 8, 16, 32, 64, 128 and 256. This makes the deblocking of records very simple with DISK BASIC. If we wish to read the 6th record in the file, and the record length is 64, we would use the following DISK BASIC statements:

```
10 OPEN"R",1,"PRODAT:0": INPUT"ENTER RECORD NUMBER";N
20 PR = 1+INT((N-1)/4): LR = (N-1) -(PR-1)*4
30 GET 1,PR: FIELD 1, (64*LR) AS DUMMY$, 64 AS R$
```

In the program above, PR is the Physical Record (sector) in the PRODAT file which contains the desired logical record. LR gives the number of the logical record within the physical sector. The logical records are numbered 0, 1, 2, and 3 in each sector — four logical records per sector. DUMMY\$ is a dummy string used for record alignment, and R\$ is the actual 64 characters of the record we want. Since we wanted the 6th record, PR would be equal to two (2) and the second sector of PRODAT is read. LR is equal to one (1) and the second logical record in sector two is placed into R\$.

Reading the PRODAT file as a sequential file is much easier to do. Since each record is followed by a carriage return, the file can be read using the LINEINPUT# statement. The following program reads in the first record of the PRODAT file:

```
10 OPEN"I",1,"PRODAT:0"
20 LINEINPUT#1, R$
```

The data for the first record which Profile created is read into the variable R\$.

The Profile Control File

The Profile program creates an important file called INFOFILE on Drive 0 of your Profile System. This file contains all information you will need to know about the PRODAT file(s). The following DISK BASIC program reads all the important information out of INFOFILE.

```
10 CLEAR2000: DEFINT A-Z: DIM D(4),NM$(32),CU(32),LN(32)
20 OPEN"R",1,"INFOFILE:0": GET 1,1: D(0)=0
30 FIELD 1, 2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
    1 AS NF$,2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)
40 NR=CVI(NR$): MX=CVI(MX$): MD=ASC(MD$): DR=CVI(DR$)
50 RL=ASC(RL$): NF=ASC(NF$): FOR I=1 TO 4: D(I)=CVI(D$(I)): NEXT I
60 GET 1,2: J=0: FOR I=0 TO NF-1: IF I=8 THEN GET 1,3: J=0
70 FIELD 1, (J*16) AS DUMMY$, 13 AS N$, 2 AS CU$, 1 AS L$
80 NM$(I+1)=N$: CU(I+1)=CVI(CU$): LN(I+1) = ASC(L$)
90 J=J+1: NEXT I: CLOSE
```

This code should be executed at the beginning of any DISK BASIC program which will access the PRODAT files. The meanings of the variables and arrays are as follows:

Variable	Meaning
NR	The Number of Records in the PRODAT file(s), including any deleted records.
MX	The Maximum Number of records allowed.
MD	The Maximum Drive number used (0 to 3)
DR	The number of Deleted Records on disk.
RL	The total Record Length of the Fixed-length records which Profile creates. (The final carriage return is included in this length.)
NF	The Number of Fields defined at initialization.
D(i)	The highest record number stored on Drive i-1 (Record numbers start at zero.)
NM\$(i)	The Name of the ith data field.
CU(i)	The PRINT@ position plus 15360, where the ith data field starts on the video display.
LN(i)	The Length of the ith data field.

Examples of Using Disk Basic

Let's use the Club Member list which was used previously in this manual. We will set up the data form with 63 characters, for a total record length of 64. The data fields have the following lengths:

Field Name	Length of Data Field
NAME OF MEMBER	16
ADDRESS	15
CITY	10
STATE	2
BIRTHDATE	8
DUES/YEAR	6
PAID DUES	6

The form would look something like this:

** PROFILE DATA FORM **

NAME OF MEMBER:

ADDRESS:

CITY: STATE: ..

BIRTHDATE: MM/DD/YY

DUES/YEAR: \$\$\$.~~cc~~ PAID DUES: \$\$\$.~~cc~~

Random File Example

The following program will read any record from disk and display it on the video display:

```
10 CLEAR2000: DEFINT A-Z: DIM D(4),NMS(32),CU(32),LN(32)
20 OPEN"R",1,"INFOFILE:0": GET 1,1: D(0)=0
30 FIELD1,2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
    1 AS NF$, 2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)
40 NR=CVI(NR$): MX=CVI(MX$): MD=ASC(MD$): DR=CVI(DR$)
50 RL=ASC(RL$): NF=ASC(NF$): FORI=1TO4:D(I)=CVI(D$(I)):NEXTI
60 GET1,2: FORI=0TONF-1: IF I=8 THEN GET1,3
70 FIELD 1, (I*16) AS DUMMY$, 13 AS N$, 2 AS CU$, 1 AS L$
80 NMS(I+1)=N$: CU(I+1)=CVI(CU$): LN(I+1)=ASC(L$)
90 NEXT I: CLOSE
100 FORI=0TOMD: OPEN"R",I+1,"PRODAT:"+CHR$(48+I): NEXT I
110 CLS: INPUT"ENTER RECORD#";N$:IFN$="END"THEN999 ELSE PRINT
120 N=VAL(N$):IFN<1 OR N>NR THEN PRINT"OUT OF RANGE": GOTO 200
130 DR=1: FOR I=1TOMD+1:IF N>D(I)+1 THEN DR=I+1:NEXT ELSE NEXT
140 NN=N-(D(DR-1)+1): IF DR>1 THEN NN=NN-1
150 PR=1+INT(NN/4): LR=NN - 4*INT(NN/4)-1: GETDR,PR: D=0
160 FORI=1TONF:FIELD DR,(LR*64+D)AS D$,LN(I) AS R$(I)
170 D=D+LN(I): NEXT I
180 IF MID$(R$(1),1,1)=CHR$(192) THEN PRINT"DELETED":GOTO200
190 FORI=1TONF: PRINT NMS(I)": "R$(I): NEXT I
200 PRINT:INPUT"PRESS (ENTER) TO GO ON";A$: GOTO 110
999 CLOSE: END
```

Example Using Sequential Access

The following program prints out a list of all Club Members that still owe any dues for this year. DISK BASIC sequentially reads in each Club Member, performs some arithmetic, and prints their name and the amount owed to the line printer.

```

10 CLEAR2000: DEFINT A-Z: DIM D(4),NM$(32),CU(32),LN(32)

20 OPEN"R",1,"INFOFILE:0": GET 1,1: D(0)=0

30 FIELD1,2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
  1 AS NF$, 2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)

40 NR=CVI(NR$): MX=CVI(MX$): MD=ASC(MD$): DR=CVI(DR$)

50 RL=ASC(RL$): NF=ASC(NF$): FORI=1TO4:D(I)=CVI(D$(I)):NEXTI

60 GET1,2: FORI=0TONF-1: IF I=8 THEN GET1,3

70 FIELD 1, (I*16) AS DUMMY$, 13 AS N$, 2 AS CU$, 1 AS L$

80 NM$(I+1)=N$: CU(I+1)=CVI(CU$): LN(I+1)=ASC(L$)

90 NEXT I: CLOSE: ONERROR GOTO 200

100 FORI=0TOMD: OPEN"R",I+1,"PRODAT:"+CHR$(48+I): NEXT I

110 DR=1: FORJ=1TONR: IF J>D(DR)+1 THEN DR=DR+1

120 LINEINPUT#DR, R$:IFLEN(R$)=0THEN120 ELSE NM$=MID$(R$,1,16)

130 IF MID$(R$(1),1,1)=CHR$(192) THEN 160

140 DY!=0:PD!=0:DY!=VAL(MID$(R$,52,6)): PD!=VAL(MID$(R$,58,6))

150 LPRINT "NAME: "NM$, "AMOUNT OWED:" DY! - PD!

160 NEXTJ: CLOSE: END

200 IF ERR=2 THEN RESUME NEXT

210 ONERROR GOTO0
  
```

This Page Is Intentionally Blank

This Page Is Intentionally Blank

IMPORTANT NOTICE

ALL RADIO SHACK COMPUTER PROGRAMS ARE DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY.

Radio Shack shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or programs sold by Radio Shack, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer or computer programs.

NOTE: Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

RADIO SHACK  **A DIVISION OF TANDY CORPORATION**

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA

**280-316 VICTORIA ROAD
RYDALMERE, N.S.W. 2116**

BELGIUM

**PARC INDUSTRIEL DE NANINNE
5140 NANINNE**

U. K.

**BILSTON ROAD WEDNESBURY
WEST MIDLANDS WS10 7JN**

PRINTED IN U.S.A.

ADDENDUM - MODEL I / MODEL III PROFILE

Version 3.0

- A. On a Model III or upper/lowercase Model I, you can now use lowercase, in field names and data. The SHIFT key functions as on a regular typewriter, and pressing SHIFT 0 locks in or releases uppercase. Profile starts in uppercase lock. Press SHIFT 0 once to undo this. (On an uppercase only Model I, Profile will "lock in" uppercase only.) On one character menu options, Y/N responses, etc., your response is always caps locked, as are the responses to any question that needs a relation (EQ, LT, etc.) as an answer. Pressing SHIFT 0 will have no effect on these responses.
- B. Since the SHIFT key is now used to distinguish between upper and lowercase, a different means of specifying "insert/delete" is needed. The Control (SHIFT ↓) key is used. CTL-I is the same as a right arrow. CTL-S is now used for "insert" and CTL-D is used for "delete". (Note that on the Model III, the control key is only the left SHIFT plus ↓).

If you convert your Model I data to Model III, the up arrow will appear as a [and the down arrow as a \. If your data is created on either the Model I or Model III, the up and down arrows will appear on the menu as <UP> and <DOWN>. Use the arrow keys (↑ and ↓) when these are indicated.

Transferring Model I 1.0 data files to Model I 3.0 Profile

If you wish to use data created with Profile version 1.0 with version 3.0, you will need to transfer the data files on the Profile program diskette (Drive 0) to the new 3.0 program diskette.

You must have at least two disk drives to transfer your files.

Before you begin, make sure you have BACKUP copies of both the new program disk (version 3.0) and your old program disk (version 1.0).

NOTE: The form that you set for printing on your Model 1.0 program disk will not be retained. Make certain that you have a hardcopy of this format before you begin your conversion.

1. Place the new Model I 3.0 program disk in Drive 0 and the Model I 1.0 program disk in Drive 1.
2. Copy your data to the Model I 3.0 disk using the following commands at TRSDOS READY.

```
COPY FORMFILE:1 TO FORMFILE:0  ENTER
COPY INFOFILE:1 TO INFOFILE:0  ENTER
COPY PRODAT:1 TO PRODAT:0  ENTER
```

DO NOT COPY ANY OTHER FILES!

3. Before you begin using your new converted disk, you must first redesign your form for printing using the PRINTING RECORDS function.

If you are using Drives 1 through 3 for data, these disks can be used with the new version without revision.

Transferring Model I 1.0 data files to Model III 3.0 Profile

NOTE: The form that you set for printing on your Model I 1.0 program disk will not be retained. Make certain that you have a hardcopy of this format before you begin your conversion. .

1. Place the new Model III 3.0 program disk in Drive 0 and the old Model I 1.0 program disk in Drive 1.
2. At TRSDOS READY, type PROFDCV and press ENTER .
3. When prompted "Source Drive", answer with 1 and press ENTER .
4. When prompted "Destination Drive", answer with 0 and press ENTER .
5. When the message "Existing File. Use it?" is displayed, answer with N for no and press ENTER . Do this for all such messages.
6. Remove the old Model I 1.0 diskette from drive 1.
7. A file named "LPFORM" will be transferred to the Model III diskette if you have defined a printer format. This file must be removed from the diskette. To delete the LPFORM file, enter the following command at TRSDOS Ready :

KILL LPFORM and press ENTER

Before you begin using your new version, re-enter your printing form using the PRINTING RECORDS function in Profile.

Using Disk BASIC With Profile

If you are experienced in using Disk BASIC and data files, you can write Disk BASIC programs to access the data files created with Profile. The Profile program was not designed to perform any mathematical calculations on numeric data. However, you may write your own programs to do calculations, comparisons and processing on the data that Profile creates.

Profile was designed so that its data files could be read by BASIC using either Random or Sequential Access methods. All the data you enter is stored in the file(s) named PRODAT. If you are using more than one disk, each disk will contain a PRODAT file. Profile always stores data in "Fixed Length Records" which are terminated with a single carriage return (0D Hex). Deleted records are filled with 'C0' Hexidecimal (192 decimal) characters. Deleted records should be ignored when found.

Note on using BASIC with Profile: The Profile disks must always be in the proper drives when accessing Profile data. Load BASIC and your programs from a TRSDOS diskette. Place the Profile disks in the proper drives before running your program. Do not swap diskettes when files are open.

Random Access

The length of the Profile data records is determined during Setup when the Fields are defined. In order for BASIC to accomodate record lengths other than 256, you must use the "Variable file mode" in Disk BASIC. This mode allows you to assign the record length when the file is opened. When reading records in the file, only the number of bytes specified will be read into the buffer.

The assignment of record lengths using the "Variable" mode is a two step process. When you enter Disk BASIC, the question HOW MANY FILES? will appear. Answer this question with the maximum number of files open at any one time followed by a "V" (example: 3V). The next step is to specify the record length of the file in the OPEN statement. This is done by adding a comma and the length to the end of the normal OPEN statement (This applies to Random Access files only). For example, if the Profile data file PRODAT had a record length of 108, the OPEN statement might look like this:

```
OPEN "R", 1, "PRODAT", 108
```

The record length for the PRODAT file is stored in the INFOFILE. Reading INFOFILE is covered in the next section.

Sequential Access

When using Sequential file access, the files may be read as any other data file with one exception. The records should be read using a LINEINPUT. This will eliminate the possibility of reading only parts of a record if there are commas or other delimiters in the record.

The Profile Control File

INFOFILE

The Profile program creates a file called INFOFILE on drive 0 of your Profile system. This file contains all the information you will need to know about the system. To read the information in INFOFILE, the following program can be used.

```

10 REM ** PROGRAM TO ACCESS PROFILE DATA **
100 CLEAR 2000
110 DEFINT A-Z : DIM D(4), NM$(32), CU(32), LN(32), R$(32,4)
120 REM ** READ "INFOFILE" **
130 OPEN "R", 1, "INFOFILE", 128
140 FIELD 1, 2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
    1 AS NF$, 2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)
150 GET 1, 1
160 NR=CVI(NR$) : MX=CVI(MX$) : MD=ASC(MD$)
170 DR=CVI(DR$) : RL=ASC(RL$) : NF=ASC(NF$)
180 FOR I=1 TO 4 : D(I)=CVI(D$(I))+1 : NEXT I
190 FOR I=0 TO 3 : GET 1, I+3
200 FOR J=0 TO 7 : K=I*8+J+1
210 FIELD 1, (J*16) AS D$, 13 AS N$, 2 AS CU$, 1 AS L$
220 NM$(K)=N$ : CU(K)=CVI(CU$) : LN(K)=ASC(L$)
230 NEXT J, I : CLOSE

```

This routine will read the INFOFILE information and should be included in your programs. The descriptions of the variables and arrays are as follows:

Variable	Meaning
NR	The total number of records in the PRODAT file(s), including deleted records.
MX	Maximum number of records allowed.
MD	Maximum drive number used (0 - 3).
DR	The total number of deleted records.
RL	The total record length for PRODAT records including all delimiters.
NF	The number of Fields defined.
D(i)	The highest record number stored on Drive i-1. (Record numbers start at 0).
NM\$(i)	The name of data field number i.
CU(i)	The PRINT @ position (plus 15360) where data field number i starts on the screen.
LN(i)	The length of data field number i.
DN	The drive number.

Appendix 4 – The Profile Control File (continued)

Examples of Using Disk BASIC

Let's use the Club Member list which was used previously in this manual. We will set up the data form with 63 characters, for a total record length of 64. The data fields have the following lengths:

Field Name	Length of Data Field
NAME OF MEMBER	16
ADDRESS	15
CITY	10
STATE	2
BIRTHDATE	8
DUES/YEAR	6
PAID DUES	6

The form would look something like this:

**** PROFILE DATA FORM ****

NAME OF MEMBER:

ADDRESS:

CITY:

STATE: ..

BIRTHDATE: MM/DD/YY

DUES/YEAR: \$\$\$.*cc*

PAID DUES: \$\$\$.*cc*

Random File Example

The following program will read any record from disk and display it on the video display:

```

10 REM ** PROGRAM TO ACCESS PROFILE DATA **
100 CLEAR 2000
110 DEFINT A-Z : DIM D (4), NM$ (32), CU (32), LN (32), R$ (32,4)

120 REM ** READ "INFOFILE" **
130 OPEN "R", 1, "INFOFILE", 128
140 FIELD 1, 2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
    1 AS NF$, 2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)
150 GET 1, 1
160 NR=CVI(NR$) : MX=CVI(MX$) : MD=ASC(MD$)
170 DR=CVI(DR$) : RL=ASC(RL$) : NF=ASC(NF$)
180 FOR I=1 TO 4 : D(I)=CVI(D$(I))+1 : NEXT I
190 FOR I=0 TO 3 : GET 1,I+3
200 FOR J=0 TO 7 : K=I*8+J+1
210 FIELD 1, (J*16) AS D$, 13 AS N$, 2 AS CU$, 1 AS L$
220 NM$(K)=N$ : CU(K)=CVI(CU$) : LN(K)=ASC(L$)
230 NEXT J, I : CLOSE

300 REM ** READ "PRODAT" DATA RECORDS — RANDOM ACCESS **
310 FOR I=1 TO MD+1 : D=0
320 OPEN "R", 1, "PRODAT:" + CHR$(47 + I), RL
330 FOR J=1 TO NF : FIELD I, (D) AS D$, LN(J) AS R$ (J, I)
340 D=D$LN(J)
350 NEXT J, I
360 CLS : INPUT " > ENTER RECORD NUMBER (OR 'END') : " ; N$
370 N=VAL(N$) : IF N$="END" THEN 999 ELSE PRINT
380 IF N < 1 OR N > NR THEN PRINT "** OUT OF RANGE **"
    : GOTO 460
390 DN=1 : FOR I=1 TO MD+1 : IF N > D(I) THEN DN=I+1
400 NEXT : NN=N-D(DN-1) : GET DN, NN
410 IF ASC(R$(1, DN))=192 THEN PRINT "** DELETED **"
    : GOTO 460
420 CLS : PRINT @ 896, "RECORD # : " ; N$
430 FOR I=1 TO NF
440 PRINT @ CU(I)-15374, NM$(I) ; " : " ; R$(I, DN) ;
450 NEXT I
460 PRINT @ 960, " > PRESS ( ENTER ) TO CONTINUE : " ;
470 INPUT A$ : GOTO 360
999 CLOSE : END

```

Example Using Sequential Access

The following program prints out a list of all Club Members that still owe any dues for this year. Disk BASIC sequentially reads in each Club Member, performs some arithmetic, and prints their name and the amount owed to the line printer.

```

10 REM ** PROGRAM TO ACCESS PROFILE DATA **
100 CLEAR 2000
110 DEFINT A-Z : DIM D(4), NM$(32), CU(32), LN(32), R$(32,4)

120 REM ** READ "INFOFILE" **
130 OPEN "R", 1, "INFOFILE", 128
140 FIELD 1, 2 AS NR$, 2 AS MX$, 1 AS MD$, 2 AS DR$, 1 AS RL$,
    1 AS NF$, 2 AS D$(1), 2 AS D$(2), 2 AS D$(3), 2 AS D$(4)
150 GET 1, 1
160 NR=CVI(NR$) : MX=CVI(MX$) : MD=ASC(MD$)
170 DR=CVI(DR$) : RL=ASC(RL$) : NF=ASC(NF$)
180 FOR I=1 TO 4 : D(I)=CVI(D$(I)) + 1 : NEXT I
190 FOR I=0 TO 3 : GET 1, I + 3
200 FOR J=0 TO 7 : K=I*8+J+1
210 FIELD 1, (J*16) AS D$, 13 AS N$, 2 AS CU$, 1 AS L$
220 NM$(K)=N$ : CU(K)=CVI(CU$) : LN(K)=ASC(L$)
230 NEXT J, I : CLOSE

300 REM ** PRINT CONTENTS OF DATA FILE — SEQUENTIAL ACCESS **
310 ON ERROR GOTO 400
320 FOR I=1 TO MD+1 : OPEN "I", I, "PRODAT:" + CHR$(47+I) : NEXT I
330 DN=1 : FOR J=1 TO NR : IF J > D(DN) THEN DN=DN+1
340 LINEINPUT # DN, R$ : IF LEN(R$)=0 THEN 340
350 NM$=MID$(R$, 1, 16) : IF ASC(NM$)=192 THEN 380
360 DY!=VAL(MID$(R$, 52, 6)) : PD!=VAL(MID$(R$, 58, 6))
370 LPRINT "NAME:" ; NM$, "AMOUNT OWED:" ; DY!-PD!
380 NEXT J : CLOSE : END
390 REM ERROR ROUTINE
400 IF ERR=2 THEN RESUME NEXT
410 ON ERROR GOTO 0

```