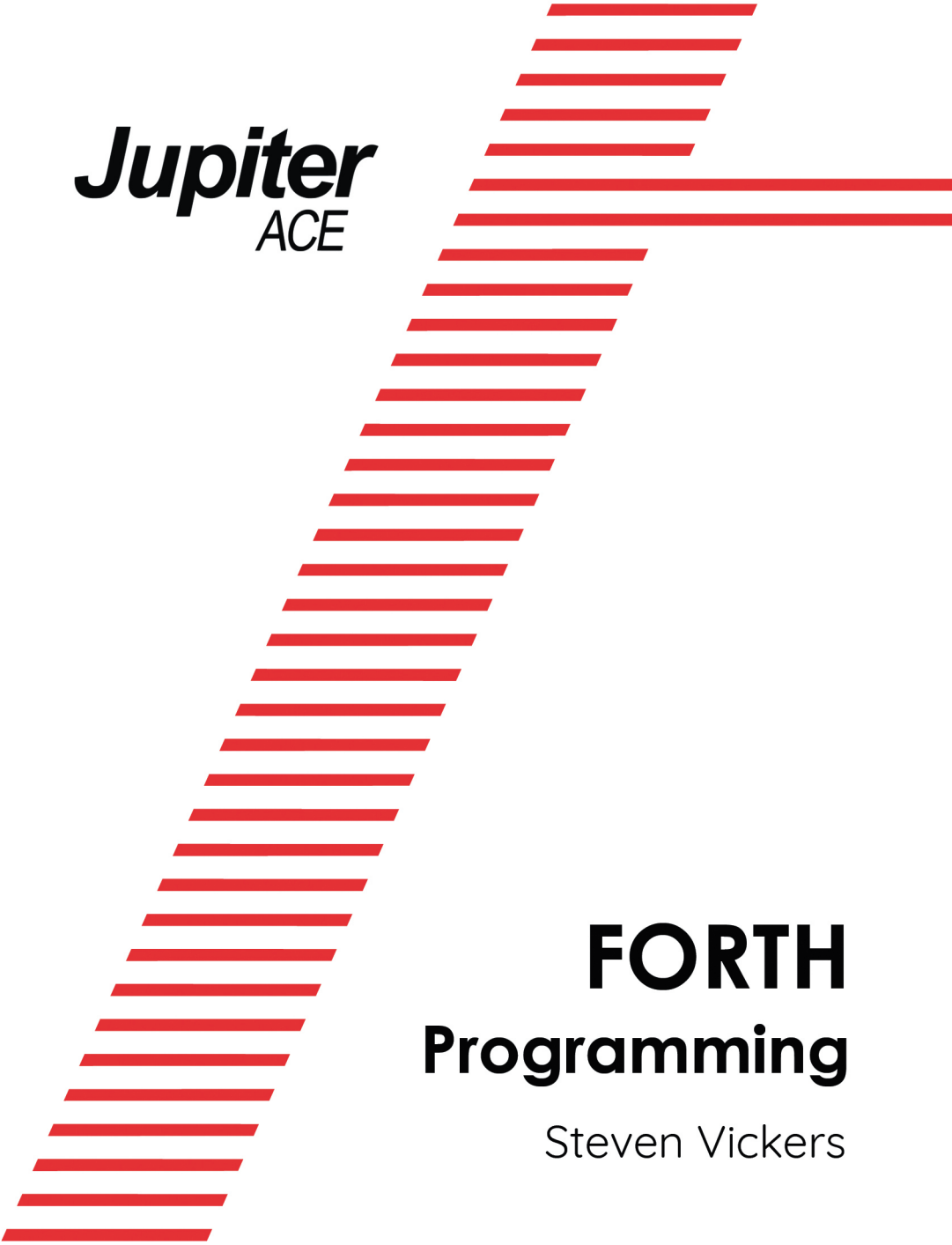


FORTH
Programming

Steven Vickers

35TH ANNIVERSARY EDITION

Contents

First published in 1982 by
Jupiter Cantab Ltd.
This 35th Anniversary Edition
published in 2017 by
Andrews UK Limited
www.andrewsuk.com

35 Years On, The Jupiter Ace Lives Again!

Launched at the peak of the legendary 1980's microcomputer rivalry, the Jupiter ACE was created by the vexed designers of the Sinclair Spectrum and became the most intriguing machine of them all.

> "Clive Sinclair was making all this money out of us, why shouldn't we just make our own computer and make the money for ourselves"
>
> *-Steve Vickers*

And with its unconventional FORTH system...

> "It's ten times faster than BASIC"
>
> *-Richard Altwasser*

The ACE was a significant machine running inspired software, but it struggled to compete in a marketplace increasingly dominated by home users wanting to play games with colour and sound. The monochrome display, tiny beeper and unfamiliar language all combined to limit its popular appeal, and very soon the ACE was in financial trouble. Jupiter Cantab Ltd, the short-lived company set up by Vickers and Altwasser to produce the machine, ceased trading during the autumn of 1983 and was in liquidation by November.

But the ACE had a second chance…

At a meeting on 6th February 1984 with Dennis Cross, the appointed Liquidator of Jupiter Cantab, it was agreed that Boldfield Computing would make a staged purchase of all the assets and stock, starting two weeks later on 21st February. Mail order sales commenced straightaway, with Boldfield Computing working quickly to add the missing software and accessories that would make the ACE viable as a machine for enthusiasts. A full brochure of products was launched during the summer of '84, and a dedicated Jupiter ACE showroom opened in Cambridge later the same year.

With the ACE now successfully selling to a niche market, and having bought up all the known stockpiles in Britain, Boldfield scoured the European outlets too, bringing back more ACEs and converting them to the UK standard. Boldfield Computing's commitment and support for the product was maintained well into 1986 – allowing the Jupiter ACE to outlast its rivals with dignity, and be assured its place in history.

The original Boldfield directors retained the Jupiter ACE brand, along with a treasure trove of its historic artefacts', but in 2015 they decided the time had come to pass on this iconic brand to another custodian. The new owners, Andrews UK Limited, are working with others not only to preserve the past, but also secure the future of the much-loved Jupiter ACE.

www.jupiter-ace.com

www.jupiterace.uk

Introduction

In 1950 the National Physical Laboratory made the Pilot ACE (Automatic Computing Engine), one of the earliest British computers. Internally it could store an amount of information measured as 1½ Kilobytes, it took 32 microseconds to perform its simplest operation and, with its large number of wires, valves and tubes filled with mercury, occupied a space the size of a small kitchen. Most of its remains can now be seen in the Science Museum at South Kensington.

Based on the Pilot ACE, English Electric developed their DEUCE (Digital Electronic Universal Computing Engine). Over six years they sold about forty of these, costing between £30,000 and £40,000 each.

Now, in 1982, Jupiter Cantab Ltd have produced their own Ace. It can store 3 Kilobytes of information (which can easily be extended) and has an extra 8 Kilobytes of program built into it permanently; the Z80A microprocessor at its heart executes its simplest instruction in just over 1 microsecond, and it is small enough to rest in your lap. Thousands of them will be made, costing less than £100 each.

How do we at Jupiter Cantab manage it? Not by being extraordinarily clever (although, of course, we are). We are simply the beneficiaries of thirty-two years of development that invented the printed circuit board, the transistor, and then methods of packing thousands of transistors onto one small silicon chip; and in the process transformed computers into machines for everyone.

Chapter 1

SETTING UP THE ACE

This manual is delivered with a few accessories, which you should check:

1. A Jupiter Ace computer.

2. A mains adaptor, which converts mains electricity into a low voltage suitable for the Ace. It will work properly only in certain countries (normally including the one to which the Ace was delivered), so if you take your Ace abroad you may need a different mains adaptor.

The mains adaptor is a heavy plastic box, three or four inches in size, and extending from it is a lead with a jack plug at the end.

3. A video lead. This is a single coaxial lead with a phono plug at one end and an aerial plug at the other. It is used to connect the Ace to a television.

4. A pair of leads with jack plugs at both ends, used to connect the Ace to a cassette tape recorder. The plugs are colour-coded, so that you can tell the two leads apart.

You will need to provide for yourself a mains electricity supply and a television, which must work on a 625 line 50Hz UHF system. (This is how most televisions in Britain work, but there are some older ones that don't. If your television can receive BBC2 then it should work with the Ace.)

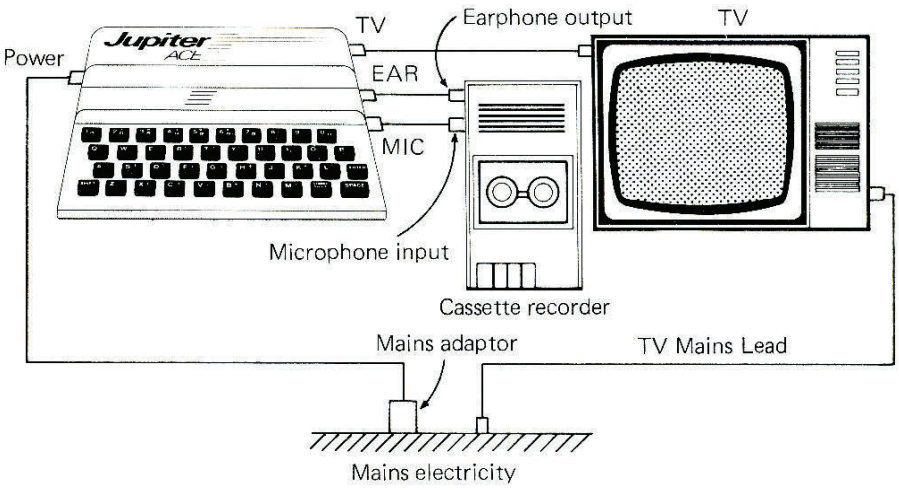
Later you will need a cassette tape recorder and tape, but these aren't immediately necessary.

Having collected all these, plug the mains adaptor into the mains and switch on there, and plug its jack plug into the socket on the left-hand side of the Ace marked (underneath) 'POWER'. There is no switch on the Ace, so as soon as you do this it starts working. However, you won't know what it's doing until you connect it to the television, so that's the next step.

Somewhere at the back of the television there should be a socket where the aerial plugs in; but instead of the usual aerial, you must plug in the video lead from the computer. Only one end will fit properly; the other end plugs in to the socket on the right-hand side of the Ace marked 'TV'

Now plug the television into the mains (unless it uses batteries, of course), switch it on, turn its volume right down, and tune it to channel 36 UHF. (If it uses buttons to select the different channels, you'll have to pick one of these and find a way of tuning it in to the computer.)

When you've tuned it just right the screen will be a uniform dark grey, except for a small white square near the bottom left-hand corner.



(Now you've set it up, you can start pressing a few keys at random on the Ace keyboard, just to see what happens. You can always get back to the starting position by momentarily disconnecting the power supply from the Ace.)

The Ace understands a powerful computing language called FORTH. FORTH was invented around 1970 by Charles Moore, and was chosen for the Ace because of its speed, its economical use of computer memory, and the way a few simple concepts give an elegant power to the whole language.

If you already know about FORTH then you will use this manual largely for reference. Chapter 2 describes the input buffer, and Appendix D describes the principle features unique to Ace FORTH.

If you know nothing about FORTH but you want to learn how to use it, then this manual is for you. Start at the beginning and work right the way through. The exercises at the end of each chapter often make interesting points that the main part of the chapter doesn't cover, so don't overlook them even if you don't feel like doing them.

There remains a third group of Ace owners who aren't interested at all in programming it, but who have bought other people's programs on cassette tape and want to be able to run them. If you're in this third group, Chapters 2 and 3 should be enough to get you going.

Chapter 2

TYPING AT THE KEYBOARD

If you've never used a computer before, you're probably feeling a bit overawed, wondering what it's going to do. The answer is nothing, until you tell it by typing in your instructions at the keyboard. Try some random typing just to see what happens. If you get in a mess, remember that you can always clear the computer out by momentarily disconnecting it from its power supply.

The first thing you'll notice is that the characters (i.e. letters, digits, punctuation marks, symbols or anything else) you type appear at the bottom of the television screen. This area is called the *input buffer* and is where the computer will look for your instructions. If you type in enough to fill up a whole line (this is easily done by holding a key down for a few seconds, because it starts repeating itself), the line will move up to make some extra space beneath it: thus the input buffer has the power of expanding upwards if necessary.

Letters usually come up as lower case (small) letters, but, as on an ordinary typewriter, you can get capitals by using the SHIFT key (bottom left-hand corner). If you have this held down when you press a letter key, the letter will come out as a capital (try it).

There is another shift key called SYMBOL SHIFT (near the bottom right-hand corner, next to SPACE) that is used for typing in the symbols — full stop, comma, +, —, brackets and so on — that you can see in the corners of many of the keys. This works in the same way as the other, capitals, shift; you keep it held down while you press another key. For instance, to get '+' you hold down SYMBOL SHIFT, press the K key, and then let up both the keys.

> Beware! Computers are very fussy that you should distinguish between the digit nought and the letter O. To make it absolutely clear, nought appears on the keyboard and television as `0`, with a slash through it. It will be printed like that in the manual too.
>
> You also need to distinguish between the digit one (1), the capital letter I, and the small letter L (l). On an ordinary typewriter you'd quite probably type a capital letter O for a nought and a small L for a one, but you mustn't do this with a computer. All ten digits are on the top row of the keyboard.

You may well be wondering by this stage why the computer isn't taking any notice of all this rubbish you've typed in. The reason is not that it's already noticed it's rubbish, but simply that it hasn't looked yet. It won't take any notice until you press what is just about the most important key on the keyboard, the one marked ENTER (on the right-hand side, one row up). Just pressing this means, 'OK computer, I've

typed in your orders. Now go and obey them.'

If you press ENTER now, the most likely effect is that a . ? will appear at the beginning. ? means, 'Do you want to change any of this?', which in your case is a tactful way of telling you it doesn't understand a word you're saying. Clear the computer out by momentarily disconnecting the power, to give yourself a chance to type in orders that it does understand.

If you now press ENTER, the computer will print 'OK' on the television screen near the top — it has happily obeyed everything you typed in (i.e. nothing) and come back for more.

The first thing to remember is that, like us, the computer understands words-- not English words, however, but FORTH words. To make the distinction, we shall print FORTH words in **BOLD** type — not because you need somehow to type them into the computer in **BOLD**, but just so that you know whether we're using a word in a FORTH sense or an English sense.

Here's a FORTH word:

VLIST

It stands for 'vocabulary list'. If, with the computer clear, you type in **VLIST** (it doesn't matter whether you use lower case letters or capitals or a mixture) and then press ENTER, you will see this (written in white on black):

VLIST

FORTH UFLOAT INT FNEGATE F/ F* F + F— LOAD BVERIFY VERIFY BLOAD B SAVE SAVE LIST EDIT FORGET REDEF INE EXIT . ' ([+LOOP LOOP DO UN TIL REPEAT BEGIN THEN ELSE WHILE IF] LEAVE J I' I DEFINITIONS V OCABULARY IMMEDIATE RUNS> DOES> COMPILER CALL DEFINER ASCII LITE RAL CONSTANT VARIABLE ALLOT C, , CREATE : DECIMAL MIN MAX XOR AN D OR 2— 1— 2+ 1+ D+ — + DNEGATE NEGATE U/MOD */ * MOD / */MOD /M OD U* D< U< < > = 0> 0< 0= ABS O UT IN INKEY BEEP PLOT AT F. EMIT CR SPACES SPACE HOLD CLS # #S U . . SIGN #> <# TYPE ROLL PICK OV ER ROT ?DUP R> >R ! @ C! C@ SWAP DROP DUP SLOW FAST INVIS VIS CO NVERT NUMBER EXECUTE FIND VLIST WORD RETYPE QUERY LINE PAD BAS E CURRENT CONTEXT HERE ABORT QUI T OK

■

This is a complete list of all the words that the Ace understands when you first turn it on (its *dictionary). You* can see that some of them are the same as English words, some are abbreviations, some are mathematical, and some are strange combinations of symbols. Near the bottom you can see **VLIST** itself. (The **VLIST** at the top is just what you typed in, copied up as a record of your typing.) The 'OK' right at the end is not a FORTH word, but just what the computer says when it's finished your orders.

You can type in more than one word at once, like

VLIST VLIST

(The computer copies up the first **VLIST,** executes by it listing the dictionary, does the same with the second **VLIST,** and then prints OK.)

It is important to put spaces in between the words. If I suddenly flip and start running allmywordstogether or spli ttingt he mup then you still know what I'm trying to say, but the computer isn't so clever. It relies very much on having spaces in between words, and no spaces in the middle of a single word. On the other hand, a word can spill over from one line to the next, like

VLI

ST

with twenty eight spaces before the V, because the computer is hardly even aware of the separate lines within the input buffer.

To summarise,

- Typing from the keyboard goes to the *input buffer* at the bottom of the screen.

- Letters are usually in lower case, but you can get capitals by keeping the key marked SHIFT held down while you press the letter key.

- In the same way, you get punctuation marks and other symbols by using the SYMBOL SHIFT key.

- The computer has a built-in dictionary of 142 FORTH words that it understands, and you can type them in using lower case or capitals, as you wish.

- If you type more than one word into the input buffer, they must be separated by spaces.

- The computer doesn't start looking at what you've typed until you press ENTER. Then it takes the words from the input buffer one by one, copying each one up to the top for the record and then executing it.

- **VLIST** is a FORTH word. It tells the computer to write a list on the television of all the FORTH words in the dictionary.

● If the computer finds a word that it doesn't understand in the input buffer, it puts in a .? at the beginning.? means, 'Do you want to change any of this?'

What if you make a typing mistake?

So far the only cure you know is to disconnect the power supply, but there are much cleverer ways which rely on the cursor - the little white square that moves along as you type. This shows where the next character that you type will appear, so if you could somehow move it back to the middle of the line you could get characters to appear in the middle.

You do this using the cursor control keys, the ones marked ⇦,⇧, ⇩ and ⇨ Although these are normally just the keys for 5, 6, 7 and 8, if you shift one - just as you would for capital letters, by holding SHIFT down - it will move the cursor in the direction of the arrow. Thus ⇦ is shifted 5, is shifted 6 and so on. (There is another up arrow, the ↑ that is symbols shifted H. This is not the same as ⇧, and just gives a character looking like ↑ .)

Afterwards, when you type in more visible characters, they will be inserted just to the left of the cursor.

Another key you will find useful is shifted 0 (DELETE) which deletes the character immediately to the left of the cursor.

As an example, suppose that you type

vlost■

by mistake. If you press ⇦ (shifted 5) twice the cursor moves back two characters:

vlo■ st

Next, DELETE (shifted 0) rubs out the 'o'

vl■ st

and finally you type 'i' to get

vli■ st

which is what you wanted. When you press ENTER, the computer doesn't mind the fact that the cursor is still in the middle.

The 'cursor up' key (⇧, shifted 6) can work in two different ways. Bearing in mind that the input buffer may have spread over several lines, ⇧ will normally just move the cursor vertically up one line. But if it is already on the top line of the input buffer (or if you'd only typed in one line anyway), ⇧ sends it to the beginning of that line. Similarly, ⇩ (shifted 7) moves the cursor either down one line or to the end of the line. Type in several lines of characters and try these two out.

Most of the other digit keys also have special meanings when shifted:

DELETE LINE (shifted 1) deletes the entire input buffer.

CAPS LOCK (shifted 2) makes subsequent letters automatically come out as capitals (like the shift lock on an ordinary typewriter). It changes the cursor to C to show that it's doing this. It doesn't automatically shift the digits to give cursor movements and so on; you still need SHIFT for these.

To get back to the usual system, press shifted 2 a second time.

INVERSE VIDEO (shifted 4) makes whatever you type come out in reverse colours — i.e. black on white instead of white on black. Again, to get back to the usual way round you press INVERSE VIDEO again.

GRAPHICS (shifted 9) changes the cursor to a G and allows you to type in the graphics characters (the black and white patterns on the digit keys). Press GRAPHICS again for normal characters.

CAPS LOCK, INVERSE VIDEO and GRAPHICS can all be turned on and off independently of each other. For instance,

Press CAPS LOCK — now letters will be capitals.
Press INVERSE VIDEO — letters will be inverse capitals.
Press CAPS LOCK again to turn it off — letters will still be inverse, but lower case. Press GRAPHICS — digits will give the graphics characters, but inverted.
Press INVERSE VIDEO again to turn it off — digits will give graphics characters exactly as on the keyboard.
Press GRAPHICS again — now everything is back to normal.

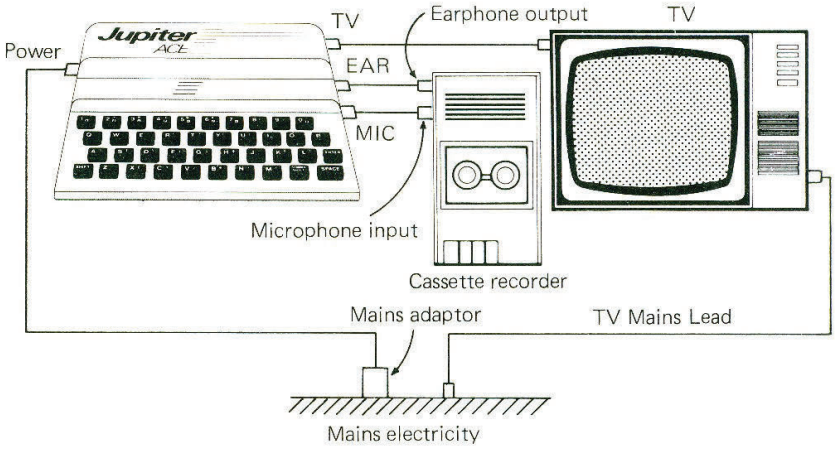
Chapter 3

LOADING PROGRAMS FROM TAPE

If you already have some cassette tapes with Ace programs recorded on them then this chapter tells you how to load those programs into the computer; otherwise skip the chapter for the time being. You can only use programs that have been recorded specifically for the Ace, and not for some other computer.

You will need an ordinary cassette tape recorder - preferably a cheap one, because expensive hi-fi stereo machines often do things to the signal that the computer won't understand. It needs to have a socket for a microphone and a socket to run an earphone, and these two sockets should fit the plugs on the pair of leads supplied with the computer.

Now connect the computer to the tape recorder with this pair of leads. One of them connects the earphone socket on the tape recorder to the socket marked EAR on the computer (make sure it's the same lead at both ends - you can tell by the colours of the plugs). The other, although you won't actually need it yet, connects the microphone socket on the tape recorder to the socket marked MIC on the computer:



A tape can have several programs, coded by the computer into a signal suitable for recording on tape. Each program has a name of up to ten characters, again coded electronically onto the tape. Let us suppose that your tape has an interesting program

]

called DVLC — it runs a game in which you are menaced by hundreds of vehicle licence application forms falling out of the sky, and you have to catch them and destroy the enclosed vehicle registration documents.

Put your tape in the tape recorder, and wind it to somewhere before the program DVLC — or right back to the beginning if you're not sure where it is. Turn the tone control, if there is one, to minimum (i.e. most bass, least treble), and turn the volume control to three quarters maximum. Type in

LOAD DVLC

press ENTER, and start the tape playing. (Note — normally on the Ace it doesn't matter whether you use capital letters or lower case; but for the name of a program on tape you must get it exactly right.)

As the computer finds various programs on the tape, it will write their names on the television screen. Eventually it will write

Dict: DVLC

and, after a few quiet clicks, OK. The program is now successfully loaded, and you can stop the tape. What the program consists of is the definitions of some more FORTH words, additional to those built into the computer. The instructions for the program should tell you how to use these words.

If the loading failed for any reason Of it just goes on and on, you can stop it by pressing SPACE — it will say 'ERROR 3'), then

- Check that the computer is correctly connected to the tape recorder.
- Check that you typed the name of the program correctly, distinguishing between capitals and lower case.
- Check that the plugs fit properly in the sockets on the tape recorder. On some tape recorders the plugs may need to be pulled out just a fraction of an inch from being fully in.
- It is possible that the volume setting matters a lot with your tape recorder. Try two or three different settings, including maximum.
- It may help to clean the tape heads on the tape recorder.

If you're not sure what programs are on the tape, rewind it to the beginning, type

LOAD

press ENTER and start the tape. The computer will eventually write up the names of all the programs.

You can have more than one program in the computer at a time (if there's room). Just load them one after another.

Some parts of programs may need to be loaded differently, with a word **BLOAD**. The instructions for the tape should tell you about this. The most usual form is

`0 0` **BLOAD** name

where 'name' means whatever name is used on the tape (like DVLC).

Chapter 4

DEFINING NEW WORDS

When you do **VLIST**, you see a list of all the words that the computer already knows about — its dictionary. When you first switch on these are the words that are built into the Ace, but the dictionary isn't final because you can define your own words. This is the process of writing a computer program, or *telling the computer how to do something new*.

As a (not very practical) example, suppose you want to teach the computer a new word **BILL**, which is to mean 'Do **VLIST** twice'. You do this using two special words, **:** (colon) and **;** (semicolon), like this:

```
: BILL
  VLIST VLIST
;
```

: is a word telling the computer that you're going to define a new word. First will come its name (**BILL**) and then the definition saying how to execute **BILL**.

So, type i n **:** (and ENTER) ... oops! Sorry, I forgot to tell you that : needs the name of the new word straight away, there and then. Otherwise it says ERROR 6 — you can look up the various error numbers in Appendix B at the back of the manual, where you can find out what went wrong. If you ever get ERROR when you're half way through defining a new word, then you have to start all over again from :.

All right, this time type in

```
: BILL
  ↑
remember the space
```

When you press ENTER the computer doesn't say OK, but that's just to remind you that you're in the middle of a definition. At least it doesn't say ERROR.

Next comes the central part of the definition, saying what the computer is to do when you use **BILL**: it is to do **VLIST** twice. Type in

```
VLIST VLIST
```

and ENTER. Again, there's no OK. Also, thankfully, there's no long list of words printed up — the computer knows it's in the middle of a definition, so **VLIST** doesn't need to be executed.

Finally, **;** means, 'The definition is finished. Now you know what **BILL** means', so

type in ; and ENTER. This time the computer will print OK.

Now the computer knows the new word **BILL**, and you can prove this in two ways.

First, if you use **VLIST**, you'll see that **BILL** has appeared at the beginning of the dictionary.

Second, if you type in **BILL**, the computer will execute it will do **VLIST** twice. If you type in BILL once too often for your patience, and get depressed at seeing the dictionary yet again, press BREAK (shifted SPACE). The computer will stop, saying ERROR 3. If you want to interrupt the computer when its in the middle of something, BREAK nearly always works. What's more - unlike pulling the plug out - it doesn't destroy the words you've defined. In some circumstances, for instance when the computer is using the tape recorder, unshifted SPACE also acts as BREAK.

BILL is a moronically useless word and nobody would normally bother to define it. But you will soon see that the same partnership of **:** and **;** can be used to define tremendously powerful words, so make sure you understand them. Remember, to define a new word, you need

first, **:**

second, and on the same line as **:** followed by a space, the name of the new word

third, the definition of the new word (which shows how the new word is made up from old ones)

and fourth, **;**

This is called a *colon* definition, because it uses **:** (there are other sorts of definition as well).

I had you defining **BILL** on three separate lines, so that I could explain it all as we went along. In practice, you'd type it all in at once, as

```
: BILL VLIST VLIST ;
```

Spaces

This is quite permissible. Also, in practice you'd use a more suggestive name --something like **2VLISTS**.

Here's a construction that can liven up word definitions; in fact it can only be used in word definitions. It enables the word to print out a message when it is executed, and consists of the word **."** (followed by a space), then the message, then the character ". (SYMBOL SHIFT P.) **."** is pronounced *dot-quote*. It's often used in conjunction with a word **CR** (Carriage Return), which makes the next message start on a new line. Here's an example:

```
: BEN
  CR ." Aah bobbop tipop weed. "
;
```