



Centre for Information
Technology in Education

**A User's Guide to the
Voice
Screen Reader**

Alistair D N Edwards & Peter Grove

CITE Report No.57



Institute of Educational Technology
Open University
Walton Hall
Milton Keynes
Great Britain
MK7 6AA

A User's Guide to the

Voice

Screen Reader

Alistair D N Edwards & Peter Grove

Institute of Educational Technology
The Open University
Walton Hall
Milton Keynes
MK7 6AA

Notice

Copyright of the Voice software and associated documentation is owned by the Open University. However, permission is granted for it to be copied - for educational purposes only. Persons wishing to make copies for any other purposes must obtain written permission from the University.

associated
iversity.
: to be
only.
ware or
s must

document
copied for
Persons wishing
document
firstly consult the University.

A User's Guide to the
Voice Screen Reader

Alistair D N Edwards & Peter Grove
Institute of Educational Technology
The Open University

Reading this manual

This user manual is divided into two parts. The first section, the *Tutorial Guide*, is intended for users who are eager to try out their screen reader. It leads the user step-by-step into getting Voice up and running and gives them a chance to try it out as quickly as possible. The second section, the *Technical Report* gives more detail about how Voice works and describes its full range of capabilities.

Contents

Tutorial Guide

1. What is Voice?
2. Getting started
3. Reading text
4. Using Voice with an application

Technical Report

1. Introduction
2. Connecting the speech synthesizer to the computer
3. Adjusting the speech
4. Browsing
5. Windows
6. Help
7. Command summary
8. Hal emulation
9. Command-line parameters
10. Making up a Voice system disc

Glossary

Appendices

- A. The Dolphin Mimic Speech Synthesizer
- B. Display Technology and screen readers
- C. Software compatibility
- D. Trouble-shooting
- E. Backing-up a floppy

Tutorial Guide

1. What is Voice?

Voice is a system which enables blind people to use computers, by converting text displayed on the computer screen into synthetic speech. It works on computers running the MS-DOS operating system and operates in conjunction with standard **application programs**¹ (word processors, spreadsheets, databases etc), so that blind people can have access to the same software as their sighted colleagues. Screen readers are not a new idea, but Voice differs from existing systems in that it will work with a greater number of **applications programs**. Also Voice offers an easily accessible on-line help facility.

2. Getting started

The purpose of this tutorial is to give you the opportunity to try out Voice. It leads you step-by-step through setting up the systems and using it. If you have any problems of Voice not working in the way described below, refer to Appendix D, which is a *trouble-shooting* guide.

There are two components of Voice: a **speech synthesizer** and a piece of software which must be loaded into the computer's memory alongside your application program. The software must be sorted out first. It is supplied on **floppy disc** and, as such, is easily damaged. It is always important to keep more than one copy of any important discs, such as your Voice disc. Ideally you should only ever use your Voice master disc once - to make a copy. Thereafter you should put the master disc away somewhere safe. Hopefully you will never need to use it again, but if you should need to - because all your other copies have been damaged - then it will be there, safe and intact. However, here is a 'Catch-22'. The first thing you should do is to make a **backup** copy of your Voice Master Disc, but if you cannot use the computer without Voice because you cannot see the screen, then you will have to use your Master Disc to make the **backup** of itself. There are two options here. Either you take a chance, load Voice and using it take a **backup** as soon as possible, or you get a sighted colleague to take the **backup** for you before you

¹ Words which appear in bold type in the text are explained in the glossary.

use Voice. Appendix E contains instructions on making a backup.

Having sorted out the software, one way or the other, you must prepare the hardware. Firstly you must connect the **speech synthesizer** to your computer. The connection is made with a cable on one end of which is a metal, tube-shaped connector with three pins (known as a *three-pin DIN* plug). That plugs into the socket on the left-hand side of the synthesizer, near the speaker. It will only go into the socket one way round, with the screw head in its side facing *upwards*. The other end of the cable has a box-shaped plastic plug. That goes into the **serial port** of your computer. Again it only fits one way up, so if you cannot get it to fit, try turning it through 180°. For a more detailed description of the synthesizer and its connections see Appendix A.

The synthesizer has two round knobs on top. The left-hand one is the on-off switch and volume control. Turn the synthesizer on by rotating that knob clockwise. Turn it a good way round to set the volume to an audible level. You should feel the knob 'click' on, and the synthesizer will say "*Ninety-six hundred*". (The significance of this little speech is explained in the Technical Report). The second, right-hand, knob controls the pitch of the speech. As you start to use Voice you may want to adjust this knob to suit your own tastes and hearing, but for the moment you can set it to an average level by finding a 'click' in the middle of its range of rotation.

How you run Voice depends to some extent on what hardware you have, whether you have one or two floppy disc drives and/or a hard disc. For the purposes of this tutorial guide we will take the 'lowest common denominator' and assume you are using a single-floppy disc system. Section 10 in the Technical Report gives you more information on running with different configurations.

You must now **boot** your system. To do that, switch on the power and place a floppy disc containing the MS-DOS system file in the floppy drive. Once the system has booted the disc drive will stop whirring and you can remove the floppy disc. Now insert your Voice disc (your Master Disc if you have not got a backup, but ideally a copy) and type

voice <Enter>

(Note that in this document, all text which you must type is printed in a

different typeface, e.g. voice, <ALT>+H. Also in describing what you must type the notation <Enter> refers to pressing the corresponding single key, as opposed to typing the word *Enter*).

The computer will say, "Voice now loaded. Press Alt+H for help". You will now find that anything you type on the keyboard is spoken. As mentioned above, Voice works alongside other programs. At the moment you are working with the MS-DOS operating system, and Voice is effectively 'looking over the shoulder' of MS-DOS and speaking out the things you are typing.

If you have not already done so, this is the point at which you should take a backup of your Voice Master Disc (Appendix E). Having done that, remove the Master disc from the computer and put it somewhere safe.

3. Reading text

Now you have got Voice loaded and running, you are in a position to hear it speaking and experiment with controlling it. If you have followed the instructions so far you will now be at the MS-DOS command level. That means there should be a **prompt** on the screen, probably as shown here:

A>

The letter displayed *may* be different. It tells you which disc drive you are currently logged on - in this case drive A. The presence of the prompt that the MS-DOS system is ready for you to type commands. If you do not understand what it means to be logged on to a disc drive - and you want to - you should consult your MS-DOS documentation.

Try typing in a line of text - but do *not* press the <Enter> key. For example, you might type,

hello world

As you type each letter is spoken (including the space). The things you are typing are acted upon by MS-DOS - which is why you should not have pressed <Enter> after typing hello world, in the above example. If you had pressed <Enter> then the line would have got passed to MS-DOS and it would have tried to execute a command called *hello*, which would have caused an error.

Sometimes, though you need to communicate with Voice, to give it commands. Often those commands are special key presses from the keyboard which are

generated using the Alt key plus another key. The Alt key works rather like the Shift key, in that you hold down the Alt key, and while still holding it down you press another key, and then you release them both. For instance, one command involves pressing Alt and L like this, and in this manual we use the notation <Alt>+L to describe such a two-key combination. Note that in describing key presses in this manual we use capital letters (e.g. the L in <Alt>+L), but this does not mean that you must press the Shift key, unless explicitly stated. We use capitals because it looks clearer.

If you make a mistake while you are typing in, you can delete the most recently typed letter using the backspace key. If you do that, Voice will say *backspace* followed by the deleted letter. Once you have typed a few words (and not pressed <Enter>), press <Alt>+L and everything on the current line will be spoken. That is to say, you will hear something like, "A, greater-than, hello world." The "A greater-than" is the **prompt**, which MS-DOS has displayed.

The key combination <Alt>+L is just one of the commands that Voice recognizes. As you go through this tutorial and the Technical Report you will learn about the others. All of the commands used by Voice are listed in Tables 1 and 2 in the Technical Report. You should notice that the letters used in Voice should help to remind you of what their associated command does. For instance, <Alt>+L reads the current **Line**, <Alt>+H gives you **Help**, and so on.

The part of the screen which is spoken always depends on the location of the **cursor**. When you are interacting with another program (such as the MS-DOS operating system or an application program) the cursor is under the control of that program, and you are limited to having the text which happens to be around the cursor spoken. When working in this way, Voice is said to be in **live mode**. However, Voice also allows you to *browse* around the screen, effectively by taking control of the cursor. To see how Voice can be used to examine screens-full of information in this way, firstly you must fill the screen.

If you now press <Enter> you will hear:

Bad command or file name. A greater-than.

This is an error message, generated because *hello* is not an MS-DOS command. It illustrates the fact that Voice speaks output from the computer as well as the things you type. It also makes the system ready for you to enter

a proper command – as signalled by the A> prompt. Make sure (a copy of) the Voice disc is in your disc drive, and type:

```
type sample.txt <Enter>
```

(Notice that the full-stop is pronounced as "period". See Appendix A for a description of how some non-**alphanumeric** characters are pronounced). A screen-full of text, taken from this manual, will be printed on the screen. You should now enter the **browsing mode** by typing <Alt>+B. Voice will say "Browsing". The relationship between Voice and MS-DOS has now changed. MS-DOS is effectively frozen, and anything you type now is assumed to be intended for Voice. This means that you do not need to use the Alt key with command letters, and in fact you *must not* use <Alt>. However, the same letters have the same meanings, and there are several more useful ones now.

When you entered **browsing mode**, Voice will have positioned the cursor at the top left-hand corner of the screen. Typing L now will read the current line (i.e. the top one on the screen). There are a number of ways in which you can examine the screen. If you type <Shift>+L then the current line will be spoken, and then the cursor will be moved down one line, so that you can then read the next line in the same manner.

Alternatively you can listen a word at a time. The command W will speak the current word and <Tab> will move the cursor to the next word. <Shift>+W will read the word *and* move the cursor on to the next one. When you get to the end of a line the computer will give a little 'beep' to warn you. If you press <Tab> the computer will beep again - but the cursor will move to the beginning of the next line. Now press W or <Shift>+W again to hear the first word, and so continue through the screen.

You can experiment with listening to the screen in this way. When you have had enough and you want to exit **browsing mode** and return to MS-DOS, press the <Esc> key. Voice will say "Returning".

If you now type <Alt>+H you will get a *help* screen of information about the letter commands. Voice will say:

Voice help screen. Press Escape to exit. Press R to read this screen or L to read line-by-line.

If you follow these instructions, and enter R you will hear all the contents of the help screen. While you are listening to this screen of help information it might be a good opportunity for you to adjust the volume and the pitch of the speech to suit yourself, using the two knobs on the speech synthesizer. You will also hear Voice say "spaces". This signifies that a line is blank. (See also the Technical Report). To get out of the help screen press the Escape (or Esc) key - even if it has not yet spoken the whole screen. This is a general convention with Voice, if you want out of anything, press <Esc>.

4. Using Voice with an application

To find out how you can really make the most of voice you must use it with an application program. At the MS-DOS level start up an application you are going to be using, such as your word processor. To do that, take out the Voice disc and insert a disc containing your word processor and type the word processor name. Once your word processor application is running, there are *four* kinds of input you can type:

1. text that you want to go into your word processed document;
2. commands to the word processor;
3. live commands to Voice;
4. browsing commands to Voice.

Word processors vary in the details of their operation, but normally you must put the application into an *insert mode* and thereafter anything you type on the keyboard is treated as input to your document text. Most word processors recognize commands as *control codes*, that is single letters which are typed in conjunction with the Control (or Ctrl) key, such as <Ctrl>+X.

So, in **live mode** normal typing (i.e. not modified by use of the Ctrl or Alt keys) is input to your application (1, above), Ctrl codes are usually commands to your application and Alt codes are commands to Voice. In **browsing mode** you use only unmodified single keys and these are commands to Voice.

You should now experiment. A challenging exercise would be to do as follows. The file LETTER.TXT on your Voice disc contains the outline of a letter, which you are going to complete. It is a form letter, which you might use to take care of your Christmas thank-you correspondence, but it needs to be modified before being sent to an individual; it needs their name inserting after *Dear*

(e.g. *Dear Auntie Mabel*) and *your* name at the end.

Load that file into your word processor and, using browsing mode, listen to its contents. The easiest way to hear the whole document is to proceed as follows:

<u>You type:</u>	<u>Voice says:</u>
<Alt> + B	<i>browsing</i>
R	<i>read window</i>
0	

(This will cause Voice to read the contents of window number zero - which is the whole screen - but do not worry about what that means for now, it will all be explained later). The first word spoken by Voice may be *dear*, but it may be that you hear a lot of other speech first. This will imply that your word processor is displaying information about itself on the screen. For now, just listen to the whole screen, and eventually you should hear the following text spoken:

Dear

Thank you very much for the present you so generously sent for Christmas. It was just what I needed.

I hope you had a very happy Christmas, and good luck for the New Year.

There are a few things which voice will say which may seem odd. It will pronounce the punctuation, and it may say something after it has spoken all of the letter. That again will be information from the word processor.

In order to insert the name of the appropriate relative you will have to locate the line containing the word *Dear*. Follow the procedure below:

<u>You type:</u>	<u>Voice says:</u>
<Esc>	<i>returning</i>
<Alt> + L	<i>Dear</i>

So, the cursor is on the right line. Move the cursor to the end of the word *Dear*. How you might do that will vary depending on the word processor you are using, but most will allow you to use the right-arrow cursor key. You can

check where the cursor is as you move it if you follow the procedure below:

<u>You type:</u>	<u>Voice says:</u>
<Alt> + C	<i>capital D</i>
→	<i>cursor right</i>
<Alt> + C	<i>E</i>
→	<i>cursor right</i>
<Alt> + C	<i>A</i>
→	<i>cursor right</i>
<Alt> + C	<i>R</i>
→	<i>cursor right</i>
<Alt> + C	<i>spaces</i>

Voice says "spaces" at this point because the rest of the line is blank. You can now type in the name. Assuming the present was from Auntie Mabel, you would proceed as follows:

<u>You type:</u>	<u>Voice says:</u>
<space>	<i>space</i>
A	<i>capital A</i>
u	<i>U</i>
n	<i>N</i>
t	<i>T</i>
i	<i>I</i>
e	<i>E</i>
<space>	<i>space</i>
M	<i>capital M</i>
a	<i>A</i>
b	<i>B</i>
e	<i>E</i>
l	<i>L</i>
,	<i>comma</i>

If you now enter <Alt> + L, you will hear that whole line:

Dear Auntie Mabel comma

Now you must close the letter. To do that you must firstly move the **cursor** to the end of the letter. Again the easiest way to do this, regardless of the particular word processor you are using, is to use a cursor key.

You type:	Voice says:
↓	<i>cursor down</i>
<Alt> + L	<i>spaces</i>
↓	<i>cursor down</i>
<Alt> + L	<i>Thank you very much for the present you so generously sent for</i>
↓	<i>cursor down</i>
<Alt> + L	<i>Christmas period It was just what I needed period</i>
↓	<i>cursor down</i>
<Alt> + L	<i>spaces</i>
↓	<i>cursor down</i>
<Alt> + L	<i>I hope you had a very happy Christmas comma and good luck for the New Year period</i>
↓	<i>cursor down</i>

Now you can enter your signing off, e.g.

Lots of love,

Les

Why not listen to the whole thing again? Just repeat the procedure you went through to hear the letter before you modified it. You may quit the word processor now. It is up to you whether you save the file on disc first.

So, you have now experimented with some of the features of Voice. When you feel you are ready, and you want to know what more it will do you can read the following Technical Report, which describes all its features.

Technical Report

1. Introduction

Voice is a screen-reading system which will work in conjunction most application programs on any computer running the MS-DOS operating system.² It has an advantage over systems in that it also works with applications which display text in a *bit-mapped* mode. (See Appendix B for a description of bit-mapped technology). Also Voice offers an easily accessible on-line help facility. Voice was developed at the Open University. The initial development was funded by the University's Research Committee. All of the programming was carried out by Peter Grove.

There are two components of the system: the (hardware) synthesizer, which produces the speech, and the software, which senses the contents of the computer screen and sends the appropriate data to the synthesizer. The user controls the speech by interacting with the software. The beauty of screen readers is the fact that they enable blind users to use the same applications as sighted users. This is achieved by having the two pieces of software in the computer's memory at the same time, the screen reader and the application. This is called being *co-resident*. Voice is loaded into memory by typing

```
voice <Enter>
```

It then remains in memory until you switch the machine off.

2. Connecting the speech synthesizer to the computer

The speech synthesizer can be connected to the computer either through a **serial** link or a **parallel** one. Computers differ in the number and types of ports they have. At a minimum they usually have one **serial** and one **parallel** port, but sometimes they may have two of either or both. The Tutorial Guide includes a detailed description of how to connect via the **serial** port. Essentially this involves connecting the synthesizer to (one of) the computer's **RS232** serial outputs. You may wish instead to use a parallel port. This might be the case,

² Our tests have not yet uncovered any PC-compatible computers with which Voice does not work. Similarly, there are very few programs which are not compatible with Voice. The only ones we know of are those which do not use the standard text fonts. We would be grateful to be notified of any users uncovering any other incompatibilities. See Appendix C for a fuller discussion of compatibility and incompatibility.

for instance, if you are using the computer as a terminal connected to another computer. Such a connection would be made through a serial port, and so if your computer has only one serial port, you can use a parallel port instead for the synthesizer.

Making the connection to the parallel port is not difficult. You will need a parallel, ribbon cable which will plug in to a parallel port of the computer (often used to connect to the printer) and then into the back of the synthesizer. (See Appendix A for a full description of the layout of the synthesizer). When you load Voice you will have to specify that the synthesizer is connected to the parallel port, and you do that by typing

```
voice p <Enter>
```

(The letter p in this line is a *command-line parameter*. See Section 9 for a description of other such parameters).

3. Adjusting the speech

Voice gives you the facilities to adjust the volume and the speed of the speech output. You will want to adjust the speech to suit your own requirements. As you should already have found, the two knobs on the synthesizer unit control volume and pitch. The right-hand one controls pitch, and if you turn it you will find it have a small 'click' at a point of average pitch. Speed of the speech is controlled by the command S (i.e. <Alt>+S in **live mode** or S in **browsing mode**). This has three levels: *slow*, *medium* and *fast*. In **live mode**, if you press <Alt>+S, Voice will set the speed to the next level and you will be told which setting it is now on. The settings are cyclic, so that repeatedly entering this command will produce: "Slow, medium, fast slow, medium..." and so on.

You should experiment with the speed adjustment to find the setting which is best for you. You will also find it helps audibility to adjust the pitch control to match the speed. If you are new to using synthetic speech you will probably want fairly slow setting, which will make the speech easier to understand. Later, when you are more used to the speech, you will want to get through the speech more quickly, and so may sacrifice some clarity for greater speed.

4. Browsing

Enter the **browsing mode** by typing <Alt>+B. Voice will say "Browsing". The relationship between Voice and your application program has now changed. The application is effectively frozen, and you can browse around the screen

and read what is displayed there. Since the application is suspended anything you type now is assumed to be intended for Voice. This means that you do not need to use the Alt key with command letters, and in fact you *must not* use Alt. However, the same letters have the same meanings, and there are several more useful ones now.

D	Define a window or label. (Sections 5)
<Tab>	Move the cursor to the start of the next word.
<Shift>+<Tab>	Move the cursor to the start of the previous word.
<PgUp>	Move the cursor to the top of the screen.
<PgDn>	Move the cursor to the bottom of the screen.
↑	Move the cursor up one line.
↓	Move the cursor down one line.
<Home>	Move the cursor to the start of the current line.
<End>	Move the cursor to the space after the last character on the line.
<Esc>	Return to live mode .
<Enter>	Return to live mode .

Table 1. Voice commands available only in **browsing mode**. Letters in the descriptions have been emboldened to emphasize the intended mnemonic cue in the command. See Table 2 for a summary of commands available in both **live** and **browsing mode**.

When you enter **browsing mode** the cursor goes to the top left-hand corner of the screen. You can now move the cursor around, using the cursor keys (↑, →, ↓, ←) and the Tab key, and have portions of the screen spoken, using the commands A, C, L, U W, or you can produce speech *and* move the cursor with <Shift>+C, <Shift>+L and <Shift>+W. (See Section 7 and Tables 1 and 2). To return to **live mode** press <Esc> or <Enter>. Voice will say "Returning". The cursor will be back in the same position as before you went into browsing mode, and your application will be ready to accept input.

Notice that Voice cannot distinguish between the cursor being at a blank point on the screen and somewhere that a space character has been displayed. So,

A	Speak the A scii code or the attribute of the current character.
B	Enter b rowsing mode. (Live mode only!) (Section 4)
C	Speak the current c haracter.
<Shift>+C	Speak the current c haracter and move the cursor to the next character.
E	Toggle e choing of capital letters
F	Load or save a definition f ile. (Section 5).
G	G o to a window or label. (Section 5)
H	Display h elp screen (Section 6).
I	Ignore next Alt key combination.
K	Alter k eyboard echo level.
L	Read the current l ine.
<Shift>+L	Read the current l ine and move to the next line.
N	N on-alphanumeric key toggle.
O	Open a window (Section 5).
P	Toggle speaking of p unctuation.
Q	Q uiet! Silence the current speech.
R	R ead a window (Section 5).
S	Change the s peed of the speech. (Section 3)
T	Speak the current t ime.
U	Read the current line u p to the cursor.
V	Set v ideo echo on or off.
W	Say the current w ord.
<Shift>+W	Say the current w ord and move on to the next word.
X	Say the row and column number of the cursor.
Z	Z ap - remove Voice from memory.

Table 2. Summary of Voice commands. Letters in the descriptions have been emboldened to emphasize the intended mnemonic cue in the command. Note that all the above commands work in both **live** and **browsing mode**, but in **live** mode they must be typed with the Alt key. See Table 1 for a summary of commands available only in **browsing mode**.

for example, if you have the following line displayed on the screen:

hello world

and you move along and read it character-by-character using the cursor key and the C command or <Shift>+C, then each of the letters of *hello* will be spoken, then "space", then each of the letters of *world*. Thereafter the rest of the line will be spoken as "spaces". Notice, though, that when reading character-by-character, Voice treats any occurrence of more than one space in the same way, so that it is difficult to distinguish the following lines, for instance:

Is there any more

Is there any more on this line

The Tab key can be used to move the cursor from one word to the next. In other words, entering <Alt>+W <Tab>, is equivalent to entering <Shift>+W. <Tab> recognizes the end of the line, so that it gives a 'beep' at the end of the line, but if you press <Tab> you will hear another beep, but the cursor will be moved to the start of the next line. <Shift>+<Tab> moves the cursor *back* to the previous word. There are a number of other cursor-movement commands available in **browsing mode** and these are all described in Table 2.

5. Windows and labels

Many **applications** use particular areas of the screen to display particular information. For instance, one rectangle at the top of the screen may be reserved for help information. Voice has a facility to enable you to access such information quickly by defining *windows* corresponding to such areas. Once you have defined such a window, then three key-presses will take you into it. You can define up to nine such windows. Also, having defined some windows, you can store their definition in a file, so that the next time you run the same application you can use the same windows without having to re-define them.

It is easier to define a window in **browsing mode**, because you can move the cursor anywhere on the screen. You define a window by specifying two diagonally opposite corners, using the command D. This is best described through an example. You can define up to nine windows, and each is identified by a number in the range 1 to 9. Suppose you want to set up a window as in Figure 1, bounded by rows 5 and 11, and by columns 10 and 30,

and you want to call it window 2. You would enter commands, and Voice would speak as below (in **browsing mode**). Firstly, you must define the corner at position (10, 5). Move the cursor to that position (you can check the position using the X command).

<u>You type:</u>	<u>Voice says:</u>
D	<i>define</i>
W	<i>window</i>
2	<i>two</i>
S	<i>start at row 5, column 10.</i>

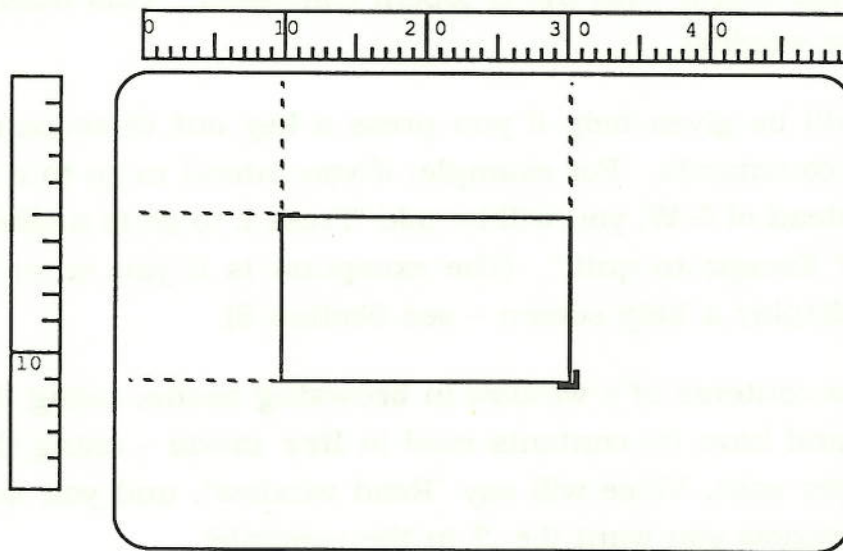


Figure 1. A window defined between columns 10 and 30 and rows 5 and 11.

Now you must define the bottom left-hand corner. So, move the cursor to the appropriate position, and you will go through a similar sequence, except that you will enter E for the *end* of the definition.

<u>You type:</u>	<u>Voice says:</u>
D	<i>define</i>
W	<i>window</i>
2	<i>two</i>
E	<i>end at row 11 column 30.</i>

Your window is now defined. You can make the cursor go to it using the command G and you can have all its contents read, through the command R.

Going to a window is generally only useful in **browsing mode**, when you can go to the window and browse through its contents. For example, to go to the window created above, the sequence would be as follows:

<u>You type:</u>	<u>Voice says:</u>
G	<i>go to</i>
W	<i>window</i>
2	<i>two</i>

Cursor movement commands (<Tab>, ↑, →, ↓, ←, <Shift>+L, <Shift>+W, <Shift>+C) will be confined within the window, as if it is a small screen. You can get out of a window by opening it with the O (open) command. This makes the whole screen the active window.

Note that you will be given help if you press a key out of sequence in any of these multi-key commands. For example, if you intend to go to a window, but you type G E instead of G W, you will be told "Press L to go to a label, or W to go to a window, or Escape to quit." (The exception is if you enter H, in which case Voice will display a help screen - see Section 6).

You can read the contents of a window in **browsing mode**, using the command R, or skip to it and have its contents read in **live mode** - using the command <Alt>+R. In either case, Voice will say "Read window", and you must type the number of the window you want (i.e. 2 in the example).

Voice defines the whole screen as window 0, so you can use the same commands to access the screen. For example, from **live mode** <Alt>+R 0 will read the whole screen.

The example above is a little atypical because you would not normally think of the limits of a window in terms of its row and column coordinates. You are rather more likely to define the window in terms of items on the screen. In other words, you would use commands such as W and C to locate the appropriate points on the screen for the window corners.

You can also *label* parts of the screen as a way of getting around quickly. Defining a label is very similar to defining a window. Labels are identified by letters, instead of the numbers used for windows, so you can define 26 labels. Again this is easier to do in **browsing mode**. Assuming you want to create a

label with the letter Z at row 10, column 20, move the cursor to that point and then proceed as follows:

<u>You type:</u>	<u>Voice says:</u>
D	<i>define</i>
L	<i>label</i>
Z	<i>at row 10, column 20.</i>

You can go to a label in a similar way to going to a window. In other words, having defined a label as above, in **browsing mode**, you could enter G L Z (go to label Z), and the cursor would jump the defined point, from which you could continue your browsing. (Appendix C on incompatibilities explains that this feature may not work properly with some applications).

It is *not* necessary to re-define windows and labels every time you use Voice. Once you have defined a set which work well with a particular application you can save that set in a file, which you can then use again next time you use that application. For example, to save the current definitions in a file called *window* from **live mode**, proceed as follows:

<u>You type:</u>	<u>Voice says:</u>
<Alt>+F	<i>File load or save?</i>
S	<i>Enter name of file to save</i>
window <Enter>	<i>window.vdf saved.</i>

The suffix *.VDF* is automatically appended to the file name and distinguishes Voice definition files.

To reload that definition file in a subsequent session you would do as follows:

<u>You type:</u>	<u>Voice says:</u>
F (or <Alt>+F)	<i>File load or save?</i>
L	<i>Enter name of file to load.</i>
window <Enter>	<i>window.vdf loaded.</i>

6. Help

You can always get help on using Voice, via the H command. This will fill the screen with information which you can browse through using the same

commands as in **browsing mode**. The help provided is sensitive to what you have been doing within Voice. For instance, if you are in **browsing mode** you will receive help on browsing. Similarly, if you forget how to define a window you can get help as you define it. For example, suppose having entered the *D* command to define a window you cannot remember what to do next so you enter *H* to get help, as below:

<u>You type:</u>	<u>Voice says:</u>
D	<i>Define</i>
H	<i>Voice help screen (press Escape to exit) showing help on defining things.</i>

You may now read through the help information, using the normal browsing commands. When you have fished, you exit the help by entering <Esc> and you can pick up from where you left off on defining the window, as shown:

<u>You type:</u>	<u>Voice says:</u>
<Esc>	<i>Leaving help screen.</i>
W	<i>Window</i>
2	<i>two</i>
S	<i>start at row 5, column 10.</i>

7. Command summary

Care has been taken to make the operation of Voice consistent in **browsing** and **live modes**. So, for instance the command to hear the current line in **live mode** is <Alt>+L and in **browsing mode** it is L. Table 2 summarizes the commands, and those which have not been explained already are described below.

A: Ascii/Attribute. All characters (letters, numbers, punctuation and some not normally visible on a screen) are represented within a computer as a number. That number is assigned according to a standard known as **Ascii** (pronounced 'askee'). This command will tell you the **Ascii** code for the character at the current cursor position. For instance, for a capital A it will say "Ascii six five", the space character is "Ascii three two" and so on. Some characters are used to control the display and are usually not visible, but these can be 'read' as their **Ascii** codes.

A character displayed on a screen also has *attributes*, which describe its

appearance. Entering the A command a second time will cause the attributes of the current character to be described. Attributes include the colour and brightness of the foreground and background. For instance, the default attribute for text displays is *white on black*.

C: say character. The character at the current cursor position is spoken. If you press C again without moving the cursor the letter will be pronounced using the phonetic alphabet (*alpha, bravo, charlie*, etc). This can be used to distinguish letters which sound similar. <Shift>+C will cause the current character to be read and move the cursor along to the next character. This use of the <Shift> key is consistent with the L and W commands, (see below). Some characters displayed are not letters or numbers, but **graphics characters**, which are used to build pictures on the screen. These cannot be read out by Voice, and are pronounced as "Graphics character".

E: toggle echoing of capital letters. Normally when reading a character, Voice will explicitly state if it is a capital letter (e.g. it would say "Capital G"). This can be switched off (so that it would simply say "G") with the E command. Pressing E again switches the option back on. This kind of two-state switch is known as a **toggle**.

I: ignore Alt key. In live mode commands to Voice are all Alt key combinations (e.g. <Alt>+L, <Alt>+W etc.) Some applications also use Alt keys, so to pass an Alt key code to an application first press <Alt>+I. This has the effect of telling Voice *not* to act on the next Alt key pressed, but to pass it on to the application. So, for example, if you must pass the command <Alt>+A to your application, you will have to type <Alt>+I <Alt>+A.

K: keyboard level. When you type Voice can speak back (or *echo*) each word you type, or each letter or not speak at all. This command causes Voice to cycle through these possibilities. Supposing Voice is in **live mode** and currently echoing every character (which is the default level). If you enter <Alt>+K Voice will say "Key echo off", which means that if you type now nothing will be spoken. This can be useful if you are a good touch typist and Voice does not speak quickly enough to keep up with your typing. If you press <Alt>+K again, Voice will say "Echo words", and if you now type, nothing will be spoken until you complete a word (i.e. press the space bar, or any non-letter key) and then the whole word you just typed will be spoken. Pressing <Alt>+K again brings you back round to each key being spoken. To recap, the options

are:

Echo characters
Key echo off
Echo words.

L: read the current line. All of the current line will be read. Any unreadable characters on the line will be pronounced as "graphics character" - or "graphics characters" if there are more than one together. <Shift>+L will read the current line and move the cursor on to the next line. Blank lines are read as "spaces".

N: toggle non-alphanumeric key echo. Letters and numbers are known as *alphanumeric* characters. Their keys are essentially the same as those you find on a typewriter, but computer keyboards can generate other characters - which are normally not displayed on the screen. These are generated by the cursor-control keys, and by combinations of the control key plus a letter (e.g. <Ctrl>+G etc). Normally Voice *will* tell you when you have pressed one of these keys. For instance if you press the *cursor-right* key (→) it will say "Cursor right". You can disable this by pressing <Alt>+N, in which case Voice will say, "Non-alpha echo off", and now nothing will be spoken when you press any non-alphanumeric keys. To switch that option back on press <Alt>+N again.

P: toggle speaking of punctuation. Normally Voice will speak all punctuation characters. Thus, *Hello, world!* will be spoken as "Hello comma world exclamation." The P command allows you to have only the words spoken. Having the speaking of punctuation on is useful for proof-reading, but it is more natural to have it off when having long passages read out.

Q: quieten speech. This command will interrupt the current speech.

T: speak the current time. This is a speaking clock.

V: set video echo on or off. Text appears on the screen when you type on the keyboard, but it is also generated by the programs you use without you doing any typing (error messages, **prompts** etc). By default Voice says all the output from programs (i.e. it echoes the video output). For instance, if you type

echo hello <Enter>

Voice will (normally) say "e-c-h-o space h-e-l-l-o enter" and then will say the output from the command (i.e. "hello"). You can disable the speaking of the

output with this command. So, if you are in **live mode** and you type <Alt>+V, Voice will say, "Video echo off". Now if you type

echo hello<Enter>

Voice will say "e-c-h-o space h-e-l-l-o enter", but not the "hello" output from that command. Switch it on again with another <Alt>+V.

W: say the current word. The current word will be spoken. Alternatively, <Shift>+W will read the current word *and* move the cursor to the next word. Any unreadable characters will be pronounced as "graphics character".

X: say cursor position. To help you navigate around the screen, Voice will tell you the current position of the cursor. So, for instance, in live mode, if you press <Alt>+X it will give you the current row and column number, e.g. "Row 25, column 28."

Z: Zap - remove Voice from memory. Voice can be removed from memory. This implies that nothing more will be spoken. This command can only be called from MS-DOS (i.e. when you have a prompt like A> on the screen), and not from within an application. Voice will check that you really want to abandon its use, by asking you to press <Enter> to confirm that you do want to remove Voice or any other key to abort the unloading and to continue using Voice.

Tab: move to next word. In **browse mode**, pressing <Tab> will move the cursor on to the next word. It is thus possible to move through the screen word-by-word. When you get to the end of a line Voice will give a little 'beep' to warn you. If you press <Tab> again the computer will beep again - but the cursor will move to the beginning of the next line. Now press W again to hear the first word, and so continue through the screen. <Shift>+<Tab> moves the cursor *back* one word.

Shift: read and move. The commands L, W and C read the current line, word and character respectively, without moving the cursor. Using them with the shift key, i.e. <Shift>+L, <Shift>+W and <Shift>+C will also read, but move the cursor on to the next position. So, <Shift>+L reads the current line and positions the cursor ready to read the next line, <Shift>+W reads the word and moves on to the next word, and <Shift>+C works the same way with characters. Note that these are available in both **browsing** and **live mode**. In **live mode** you

would have to type <Alt>+<Shift>+L and so on.

8. Hal emulation

Voice Command	Hal Command	Action
<u>Live mode</u>		
<Alt>+B	<Alt>+J	Enter browsing (reading) mode.
<Alt>+C	<Alt>+I	Speak the current character.
<Alt>+F	<Alt>+F	Load or save a definition file.
<Alt>+I	<Alt>+N	Ignore next Alt key combination.
<Alt>+L	<Alt>+L	Read the current line.
<Alt>+Q	<Alt>+M	Silence the current speech.
<Alt>+R	<Alt>+P	Read a window.
<Alt>+S	<Alt>+S	Change the speed of the speech.
<Alt>+U	<Alt>+K	Read the current line up to the cursor.
<Alt>+W	<Alt>+O	Say the current word.
<Alt>+X	<Alt>+G	Say the row and column number of the cursor.
<u>Browsing (reading) mode</u>		
<PgUp>	T	Move the cursor to the top of the screen.
<PgDn>	B	Move the cursor to the bottom of the screen.
↑	U	Move the cursor up one line.
↓	D	Move the cursor down one line.
<Home>	S	Move the cursor to the start of the current line.
<End>	E	Move the cursor to the space after the last character on the line.
<Enter>/ <Esc>	<Enter>	Return to live mode.

Table 3. Summary of Hal equivalents to Voice commands.

Hal is another screen reader, available from Dolphin Systems. Though similar to Voice, *Hal* is less well adapted for use with **bit-mapped** displays (it will read *some* text displayed in bit-mapped mode). *Hal* uses the same protocol of using Alt keys as commands to the screen reader. However it uses a completely

different set of command letters. For the benefit of users who are accustomed to using Hal, but need to move to Voice, Voice will work with a limited subset of Hal's commands. The commands and their Voice equivalents are presented in Table 3. In order to make Voice recognize Hal commands instead of its own set it must be loaded with a command-line parameter, as explained in Section 9.

9. Command-line parameters

There are a number of options available when you load Voice. As mentioned in the previous section, it is possible to run it in such a way that it accepts Hal commands, and it is also possible for it to be connected to different output ports on the computer (See Section 2). The way that you specify any such options is through *command-line parameters*, which you enter when you load voice. The parameter to make Voice emulate Hal is *h*, so to load Voice to run in this manner you would type

```
voice h <Enter>
```

The complete set of available command-line parameters is as follows:

h	Hal emulation
s2	use the serial port number 2
p1	use parallel port number 1
p2	use parallel port number 2

Command-line parameters can be combined, so, for instance, to load Voice with Hal emulation, using parallel port 2, you would enter

```
voice h p2 <Enter>
```

10. Making up a Voice system disc

The easiest way to use Voice is to have all files associated with it on the same disc as your MS-DOS system. That is to say that you should copy the file `voice.com` from your Voice disc (the backup, not the master) onto your system disc. It makes no difference whether you have a hard disc or a floppy disc as your system disc. You will probably want to have Voice loaded automatically every time you **boot** your system. To do that you should add a suitable command to your `AUTOEXEC.BAT` file. That simply involves editing the `AUTOEXEC.BAT` file (using a word processor or other text editor) to add a line containing the word:

```
voice
```

Normally this should be the last line in the file, so that Voice is loaded after any other memory-resident software. This will ensure that Voice does not clash in its memory allocation with any such programs. See your MS-DOS documentation for details of the AUTOEXEC.BAT file.

Acknowledgements

Voice was developed under a grant from the Open University Research Committee. Valuable help was given during its development by David Calderwood. Dr Tom Vincent assisted with the writing of this manual, providing useful comments on an early draft.

Glossary

- application** (See *application program*)
- application program** A program written for a specific user application, such as a word processor, spreadsheet, database etc.
- Ascii** An acronym for *American Standard Code for Information Interchange*. This is the most commonly used coding system for representing textual data in computers. In fact it is also a European, International Standards Organization (ISO) standard.
- backup** A copy of a disc or file, kept in case the original should become damaged. Also used as a verb, meaning to create a backup copy. The original disc is usually referred to as the *master* copy.
- bit-mapped display** Computer screen technology based on breaking images down into dots. See Appendix B.
- boot** A verb meaning to start up your computer system. It is derived from the idea that the system *pulls itself up by its own bootstraps*.
- browse mode** This is the style of interaction which enables you to examine the contents of the whole screen. The application or operating system programs are essentially frozen and you can issue commands directly to Voice to specify sections of the screen to be spoken. Letter commands in browse mode *must not* be typed in combination with the Alt key.
- co-resident software** More than one piece of software being loaded in computer memory at a time. This is the mechanism through which a screen reader can interact with an **application program**.
- cursor** A marker on the computer screen, indicating the current active area of the screen (i.e. normally where the next character typed will appear).
- DIN** A connector, based on a standard devised by the German Standards Organization, DIN (Deutsche Industrie Norm).
- floppy disc** Computer software and data are usually stored on magnetic discs. These come in two main categories: *floppy* and *hard*. Floppy discs (or just *floppies*) are cheaper, but have less storage capacity and are more

prone to damage. They can also be easily transferred from one computer to another.

- graphics character** A character which is not a letter or number but a shape, which can be used to build up pictures. Graphics characters cannot be read by Voice, and are pronounced as "Graphics character" - or "Graphics characters" if there are more than one together.
- live mode** This is the form of interaction where you are principally using an application program, or the operating system, with Voice giving some spoken help as to what is happening on the screen. The alternative is **browse mode**. All commands to Voice in live mode are combinations of the Alt key plus a letter.
- parallel port** An output socket from a computer which transmits data in several (usually eight) bits simultaneously. This is commonly the way that printers are connected. Cables for parallel connections can usually be easily recognized as they consist of a flat 'ribbon'.
- prompt** A signal from the computer that it is ready for the user to type some input. In MS-DOS the normal prompt is some thing like A>, which also signifies that the current disc drive is drive A.
- RS232** An approved standard for voltage interfaces, commonly used for **serial** transfer of data.
- serial port** An output socket on a computer which sends data one bit at a time. Often also referred to as **RS-232**, which means it conforms to a certain standard regarding the connection.
- speech synthesizer** A device which will produce speech-like sounds.
- toggle** In electronics, a two-position switch. In software this has come to be used to describe a two-valued state, such as *on* and *off*. Also used as a verb: to toggle between the two allowed states.

References

- Edwards, A. D. N. (1987) *Adapting user interfaces for visually disabled users*. PhD Thesis, The Open University, July 1987 (On Microfilm at the British Lending Library, Shelf number DX 80409).
- Edwards, A. D. N. (1988) *Soundtrack: an auditory interface for blind users*, Human-Computer Interaction (accepted for publication, to appear in a special edition on the use of sound in interfaces).

Appendices

Appendix A

The Dolphin Mimic Speech Synthesizer

Voice is designed to work with the *Mimic*, manufactured by Dolphin Systems. It is one example of the low-cost synthesizers currently available, and is one already widely used in Britain by visually disabled users, using the *Hal* software. Although Voice has been written to work with the *Mimic*, but is by no means tied to that hardware; it would be a relatively simple programming job to re-configure the software to work with other synthesizers.

There are four major components to the system:

- the speech unit;
- a cable to connect the unit to the computer;
- a battery charger;
- a pair of headphones.

The unit is about the size of a thick paperback book. There two round knobs, about the size of ten-pence pieces on its upper surface. Behind them the unit rises in a wedge shape in the centre of which is a rectangular slot with a fabric grille, covering the speaker.

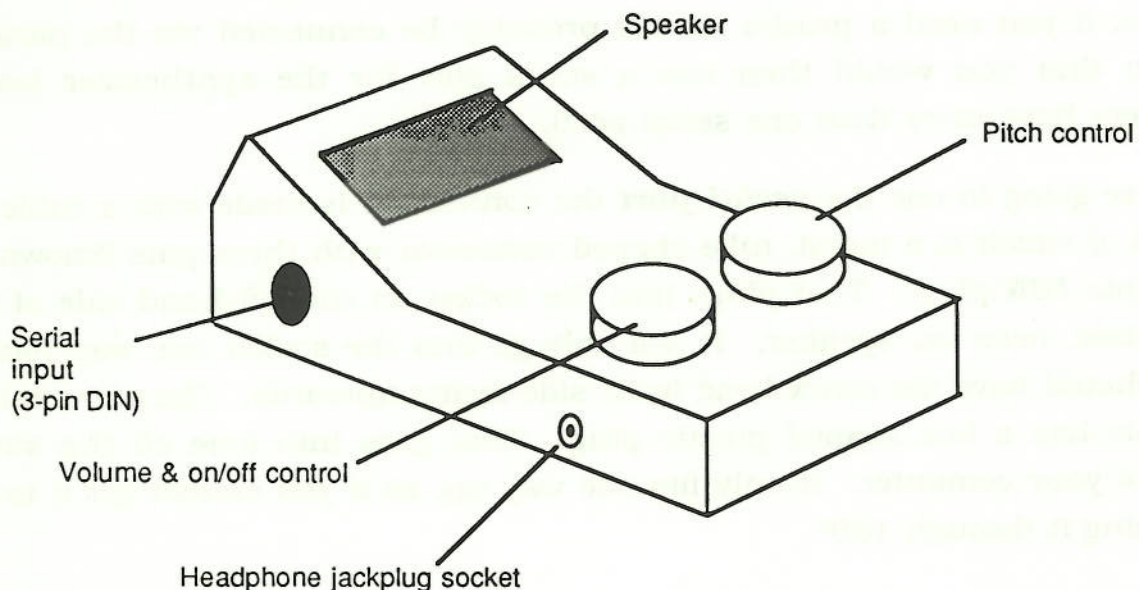


Figure A-1. The Dolphin Mimic speech synthesizer

On the left-hand edge, approximately in line with the knobs is a small jack-

plug socket. This is where you plug in the headphones, should you want to use them. Further back on the left edge is another socket. This is a hole about the size of a penny. There are two ways of connecting the unit to the computer and one of them (the **serial** connection) uses this socket. The second possible connection (the **parallel** one) is via a socket at the back. This is a plastic and metal slot, about 50 cm x 1 cm (2 x 1/2 inches). On either end of it are two hinged metal loops which are used to hold the connector in place. Near the back on the right-hand edge is another hole. That is the socket for the battery charger.

These connections and their uses are described below.

Headphones

The speech through the headphones is generally more audible than through the speaker; it is also less distracting for anyone else in the room! The Mimic includes a pair of personal-stereo-style headphones. To use them simply plug them into the socket, as identified above. Plugging in the headphones disconnects the built-in speaker.

Connecting to the computer

There are two ways to connect the synthesizer to the computer: either through a **serial** or a **parallel port**. Which you use will depend on your own needs, whether any of the computer outputs are needed by other devices. For instance, if you need a printer, it will probably be connected via the parallel port, so that you would then use a serial port for the synthesizer (some computers have more than one serial port).

If you are going to use the **serial port** the connection is made with a cable on one end of which is a metal, tube-shaped connector with three pins (known as a *three-pin DIN* plug). That plugs into the socket on the left-hand side of the synthesizer, near the speaker. It will only go into the socket one way round, and it should have the screw head in its side facing *upwards*. The other end of the cable has a box-shaped plastic plug. That goes into (one of) the **serial** port(s) of your computer. It only fits one way up, so if you cannot get it to fit, try turning it through 180°.

To connect through the **parallel port** you must use a ribbon cable. This connects to the parallel output socket so your computer and into the parallel input socket at the back of the synthesizer. To load Voice so that it will run

through the parallel port, you must type

voice p

Power supply

The unit can be powered in three ways:

1. directly from the mains, via the power pack transformer;
2. on built-in, rechargeable batteries;
3. by an external, non-rechargeable battery.

If you connect the unit to the power pack transformer - and plug the transformer into the mains - the unit will run on the mains current. If you leave it in when not in use, the mains will continue to recharge the internal batteries. As long as the internal batteries are charged, the unit can be run off them without a mains connection. Alternatively, you can fit an external battery. To fit an external battery, turn the unit over. On the bottom (now facing upwards) in the centre, about 4 cm (1 1/2 inches) from the front edge are three horizontal lines, and a small triangle). Place a finger there and, pushing down slightly, draw your finger towards you. This should open a sliding cover, with a 'click'. This will reveal a compartment inside which is a wire and a fitting which clips onto the top of a PP3 battery (or equivalent). Clip the battery on, place it and the cable in the compartment and slide the lid back into place. The unit uses a lot of power, so you should use an alkaline battery.

You will know when the battery power is getting low because the unit will start to make a horn-like noise. You should then either replace the external battery, or plug in the power unit, depending on whether you use the internal or external battery.

The speed with which a computer sends data through a communications port is measured in *baud*. By default, the Mimic synthesizer expects to be connected to a port running at 9600 baud. That is why, when switched on it says "Ninety-six hundred". It is possible to reset it to work at 1200 baud, if that is the output speed of your computer. To do that you will have to open up the synthesizer by unscrewing the base. There is a switch on the left-hand side, near the DIN socket. When pointing to the right the setting is 9600, switching it to the left will change it to 1200.

The Mimic's vocabulary

Being of American manufacture, the Mimic sometimes pronounces characters differently from how a British user might expect. Also, some characters are used in more than one way, which can cause confusion. For instance a dash may be used as a minus sign. Notice, too, that numbers are spoken digit-by-digit. So, for instance, 42, would be pronounced as *four-two*, rather than *forty-two*. The Carriage-return key is pronounced as "Enter".

Below is a table of how the Mimic pronounces non-alphanumeric characters.

<u>Character</u>	<u>Pronunciation</u>	<u>Character</u>	<u>Pronunciation</u>
-	dash	(left parent (short for left parenthesis)
[left bracket)	right parent
]	right bracket	_	underline
'	apostrophe (may be used as a single-quote mark)	{	left brace
.	period	}	right brace
/	slash	"	quotes
\	backslash	~	tilde
*	star	<	less-than (may be used as a diamond bracket, as in this manual)
!	exclamation	>	greater-than
@	at	?	question mark
#	pounds		vertical line
\$	dollars	%	percent
^	up-arrow	&	ampersand

Appendix B

Display Technology and screen readers

Text-based displays

Computers are often described as enormous calculators, doing arithmetic on numbers, very quickly. At one level that description is correct, but it is also misleading. Numbers can be used to represent many different forms of information. So, for example, computers can manipulate written text, and this is a very important area of their application. However, written text does have to be converted into a numerical form before a computer can manipulate it. That is to say that the letters of the alphabet are encoded as numbers. The commonest encoding scheme is the *American Standard Code for Information Interchange* (known as **Ascii**). Table B-1 shows an extract of the encoding scheme. For instance, the letter *a* is represented by the number 97, *b* is 98, and so on.

<u>Letter</u>	<u>Code</u>	<u>Letter</u>	<u>Code</u>
a	97	A	65
b	98	B	66
c	99	C	67
d	100	D	68
e	101	E	69
f	102	F	70

Table B-1. An extract of the Ascii character set.

In this way textual information can be encoded in computers. Groups of letters (usually known as *strings*) can be represented simply by combining their letters. For example, referring to Table B-1, the string "Cab" would be represented by

67 97 98

Additionally the computer must be signalled that it is dealing with text and not

numbers. That implies that different rules apply to the way it will manipulate the information. For instance, it makes no sense to add together two letters, but it would be legitimate to search the computer's memory for a particular letter by looking for a code equal to the appropriate value. Thus, if the user requested the computer to search for a *B* it would translate that request such that it actually tested for a value of 66.

Older computers store the information displayed on the screen in a special area of their memory, using this form of encoding. So, for example, if the word *Cab* is displayed on the computer screen, then if the appropriate part of the computer's memory is examined (by a program) then the numbers shown above would be found. Programs display output on the screen by putting the appropriate codes in the right part of memory. The computer hardware converts these codes into the corresponding letters, which it then displays on the screen.

This is how conventional screen readers work. The application program (word processor) displays text on the screen in the manner described. The screen reader can then examine the area of memory corresponding to the screen and send the appropriate strings of Ascii codes to the speech synthesizer. The speech synthesizer has built-in rules which enable it to pronounce English words. So, if the synthesizer receives the codes 67 97 98, it generates the sound *cab*.

Bit-mapped

A computer screen is a form of cathode ray tube, as also used in televisions. The screen can be thought of as an array of picture elements or *pixels*. On a screen each pixel can be either *off* or *on*, and will correspondingly be either black or white (on a black-and-white screen). If one was to examine a computer screen with a magnifying glass, it would be possible to see that the objects on the screen are made up of such dots. For example, the letter *O* might appear as below:



Figure B-1 shows that same letter, magnified many times. Each square represents a pixel and the black squares are off, while the white ones are on.

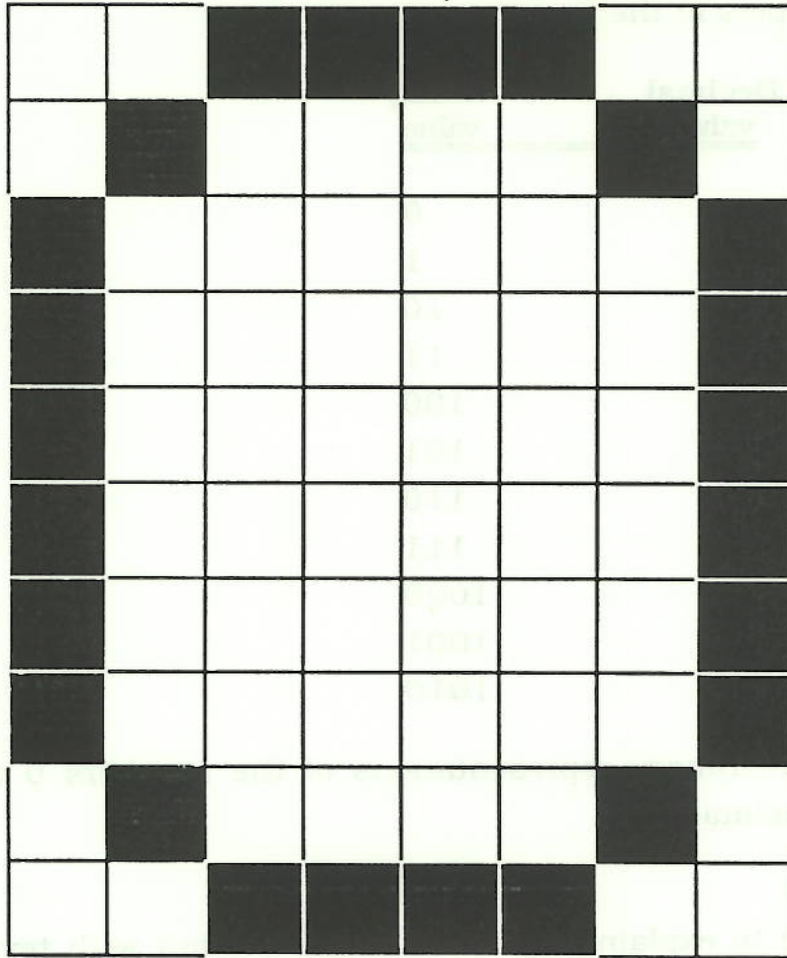


Figure B-1. The pixel pattern of the letter O.

It was stated above that computers deal with numbers. This is true, but it was also implied that they use *decimal* numbers, that is numbers that can be represented by the digits 0 to 9. That is not strictly true, in fact computers use *binary* numbers, which are constructed using just two digits, 0 and 1. The details of this number system are not important here, but the binary representations of the decimal numbers 0 to 10 are given in Table B-2.

It may become clearer in the discussion below why it is convenient for computers to work in the binary number system. The on and off pixels of the screen can be represented by binary numbers: a 1 representing an on pixel and a 0 for one which is off. Figure B-2 shows the same letter O represented in that way.

If each row of Figure B-2 is treated as a binary number, it is possible to calculate its decimal equivalent, as on the right of the figure. If the letter O is displayed on a computer screen, it would be represented within the computer

as the set of numbers in the figure.

<u>Decimal value</u>	<u>Binary value</u>
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Table B-2. Binary representations of the numbers 0 to 10 (decimal).

Screen readers

It is now possible to explain why screen readers work with text displays but not with bit-mapped ones. In a text-based display each character is represented by a single, unique number. The letter O is represented by the number 79. A screen reader can examine the display memory, and decode all the numbers. However, on a bit-mapped display the letter O will be represented by a set of numbers, as shown in Figure B-2. If a user attempted to use a conventional screen reader with such a display it would not work at all. For instance, the second row of the O has the numerical value 66, which it would treat as the letter B, clearly an error.

0	0	1	1	1	1	0	0	00111100 = 60
0	1	0	0	0	0	1	0	01000010 = 66
1	0	0	0	0	0	0	1	10000001 = 129
1	0	0	0	0	0	0	1	10000001 = 129
1	0	0	0	0	0	0	1	10000001 = 129
1	0	0	0	0	0	0	1	10000001 = 129
1	0	0	0	0	0	0	1	10000001 = 129
1	0	0	0	0	0	0	1	10000001 = 129
0	1	0	0	0	0	1	0	01000010 = 66
0	0	1	1	1	1	0	0	00111100 = 60

Figure B-2. The pixels representing the letter 0, represented as binary numbers.

Appendix C

Software compatibility

There is a wider range of applications programs which can be used with Voice than with other screen-readers, because Voice has the capability to speak text even when it is displayed in a **bit-mapped** format. (See Appendix B for more technical details). However there are still limitations, which depend on the programming techniques which have been applied by the creators of the applications programs.

Selection of the area of the screen to be spoken is based around the cursor position. Voice relies on a record of the current cursor position which is held in the system memory to know where the cursor is. However, some applications move the cursor without telling the system. This means that it is possible when working in **live mode** for the application and Voice to have different ideas as to where the cursor is. This may cause odd effects. It is not a problem in **browse mode**.

Also some applications use a cursor which is entirely unconnected with the system cursor. In that case, it is impossible for Voice to know the location of the cursor used by the application. This means, for instance, that it is not possible to use the <Alt>+U command to read back the line up to the cursor. *Framework* is an example of a program with such a built in cursor.

Most programs use text fonts which are supplied as part of the computer system. However, it is possible for programs to use their own, internal fonts. It is not possible for Voice to recognize these fonts and so it will not speak them. Voice will speak these as **graphics characters**.

Gem is an example of such a program. However, Voice will work with applications which have been launched from *Gem* - as long as *Gem* is loaded *before* Voice. On the other hand, *Gem* uses such a visual form of interaction that it is unlikely that any blind people will use it.³ Other examples are found in the *Help screens* in *Framework*.

³ Work is being carried out at the Open University into how such visual interfaces might be adopted for use by blind people. See Edwards (1987 & 1988).

Appendix D

Trouble-shooting

Symptom	Possible cause and remedy
Computer gives no 'beeps' at ends of lines.	Speaker volume is turned down. Turn up the volume control.
Speech synthesizer makes constant beep.	Batteries are low. Either replace battery or plug in to recharger.
Speech synthesizer does not speak.	Volume turned too far down: adjust volume knob. Batteries flat. Either replace battery or plug in to recharger. Not connected to computer. Check both ends of cable.
Voice says the same thing twice.	Voice reads the characters as you type them, but they are sometimes displayed separately on the screen, and at that point Voice speaks them again. This is unavoidable.
Computer prints the message <i>Bad command or file name</i> .	You have entered a command which is not recognized by MS-DOS. Did you press <Enter> after typing <i>hello world</i> ?

Appendix E

Backing-up a floppy

The procedure for making a backup copy of a floppy disc varies a bit, depending on whether you have a single or double floppy disc drive, and whether you have a hard disc. The procedure below should, however, work on all configurations. The possibility of damaging a disc can be reduced by ensuring that the computer cannot write anything on it by accident. This is known as *write-protection*. Floppy discs have a small notch in their edge (see Figure E-1). If you hold the floppy as if you are going to put it into its drive, with your thumb on the label and feel along the left-hand edge, you should feel the notch about 3 cm (2 1/4 inches) along. If you cannot find the notch, the disc may already be write-protected. If you do explore floppy discs with your hands, do be careful not to touch the recording surface. It is exposed though a slot just beyond the central hole. Touching it (with sweaty fingers) can corrupt the data on the disc.

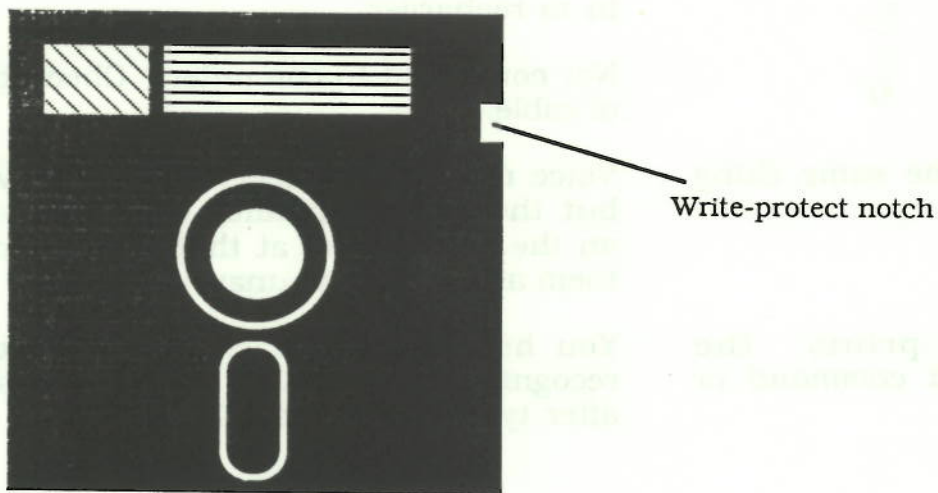


Figure E-1. A floppy disc, showing the write-protect notch

The computer can detect if that notch is covered up, in which case it cannot write to the disc. The notch can be covered by a metallic sticker and these are normally supplied with new discs. Figure E-2 shows a write-protected floppy.

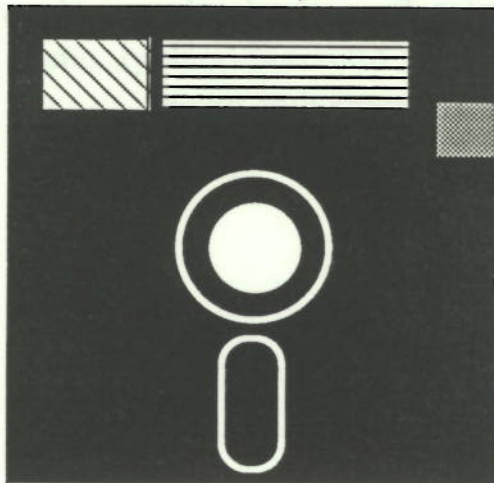


Figure E-2. A write-protected floppy disc.

It is safer if you write-protect a disc before making a backup copy of it. So, write-protect your *Voice Master Disc* - if this has not been done already. Then, with your *MS-DOS* floppy in the floppy drive type:

```
diskcopy <Enter>
```

You will be prompted:

```
Insert SOURCE diskette in drive A:
```

```
Press any key when ready...
```

Insert the (write-protected) *Voice Master Disc*, as instructed and press a key. There will be some messages about the amount and format of the data, then the disc drive will whirr for a while. Eventually you will get a message:

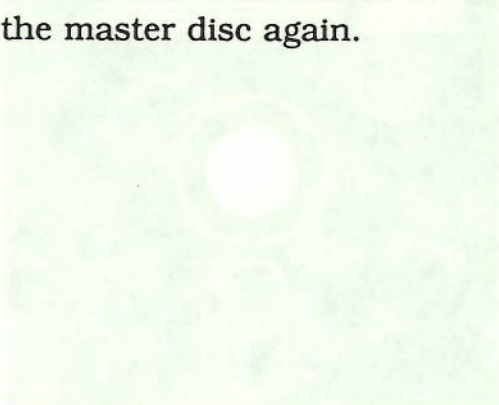
```
Insert DESTINATION diskette in drive A:
```

```
Press any key when ready...
```

Take out the master disc, insert a blank *non-write-protected* floppy in the drive and press a key. The drive will whirr again and eventually you will get a message that the copy is complete and you will be asked whether you want to copy any more disks. Enter N (for *no*).

Now go and put your master disc somewhere safe, where it will not get lost, or too hot - or too cold, and away from any magnets (such as the one in the back of your computer monitor and your television).

You can make further copies from the backup. Many experienced computer users adhere to the adage, *Why take one backup, when six will do?* Ideally you should never have to use the master disc again.



CENTRE FOR INFORMATION TECHNOLOGY IN EDUCATION

List of CITE Reports

These reports may be obtained from:

Hansa Solanki, Institute of Educational Technology, The Open University, Walton Hall,
MILTON KEYNES, MK7 6AA, England.

<u>Report No.</u>	<u>Title and Author</u>
1	A.T. Vincent, (1985) Computing and the Blind.
2	A. Jones, G. Kirkup, J. Morrison (1985) A Trial of Home Based Computer Terminals.
3	Gill Kirkup, (1985) The Present and Potential Use of Ceefax in the Open University.
4	Mark Elsom-Cook, (1986) Artificial Intelligence and Computer Assisted Instruction.
5	Mark Elsom-Cook, (1986) A Pascal program checker.
6	Simon Holland, (1986) How computers are used in the teaching of music and speculations about how Artificial Intelligence could be applied to radically improve the learning of composition skills.
7	Simon Holland, (1986) Design consideration for a human-computer interface using 12-tone three-dimensional harmony space to aid novices to learn aspects of harmony and composition.
8	Alison Petrie-Brown, (1987) The Influence of Context and Coherence as a Foundation for Dialogue Research.
9	Eileen Scanlon, Randall B. Smith (1987) A Rational Reconstruction of a Bubble Chamber Simulation Using The Alternate Reality Kit.
10	Mark Elsom-Cook, (1987) Intelligent Computer-Aided Instruction research at the Open University.
11	Mark Elsom-Cook, (1987) Towards a framework for human-computer discourse.
12	Mark Elsom-Cook, (1987) MATILDA AND IMPART: Lisp tools.
13	Mark Elsom-Cook, (1987) Guided discovery tutoring and bounded user modelling in Intelligent Computer Aided Instruction ed. J. Self, Chapman-Hall 1987.
14	A.M. Petrie-Brown and M.T. Elsom-Cook, (1987) An Examination of an AI model of indirect speech acts.
15	A. Edwards, (1987) Integrating Synthetic Speech With Other Auditory Cues In Graphical Computer Programs For Blind Users.
16	S. Holland, (June 1987) A knowledge-based tutor for music composition.

- 17 S. Holland, (June 1987) New Cognitive Theories of Harmony Applied To Direct Manipulation Tools for Novices.
- 18 M. Baker. (July 1987) Intelligent Computer-Aided Instruction and Musical Performance Skills.
- 19 M. Baker, (August 1987) Proposed Research Directions for Intelligent Computer-aided Instruction in Musical Performance Skills.
- 20 A D N Edwards, (August 1987), Adapting interfaces for visually disabled users.
- 21 M Elsom-Cook, (September 1987), Acquisition of computing skills.
- 22 M. Baker, (September 1987), Computational Analysis of musical grouping structures.
- 23 M Baker, (June 1987), Automated Analysis of Musical Grouping Structures as a Basis for a Guided Discovery Environment for Interpretation of Music.
- 24 D Laurillard, (October 1987), The different forms of learning in psychology and education.
- 25 A.D.N. Edwards, (November 1987) Modelling blind users" interactions with auditory computer interface.
- 26 P. Fung, (November 1987) Novice Prolog Programmers.
- 27 P. Fung, B. DuBoulay & M. Elsom-Cook, (November 1987), An initial taxonomy of novices' misconceptions of the Prolog interpreter.
- 28 G. Kirkup, (November 1987), Considering the effect on women students of an increased use of microcomputers in distance education.
- 29 Sara Hennessy, Rick Evertsz, Dave Ellis, Phil Black, Tim O'Shea, Ann Floyd, Design Specification for 'Shopping on Mars' a computer-based Educational Activity.
- 30 A D N Edwards, The Use of home computers by disabled students at the Open University. Part 1: Previous use of computers in courses
- 31 Michael E. Fox (March 1988) A Report on Studies of Motivation, Teaching and Small Group Interaction with special reference to Computers and to the Teaching and Learning of Arithmetic.
- 32 Michael E. Fox (March 1988) Theory and Design for a Visual Calculator for Arithmetic.
- 33 M. Baker (February 1988) A Cognitive Model for Perception of Musical Grouping Structures
- 34 M. Baker (February 1988) An Artificial Intelligence Approach to Musical Grouping Analysis
- 35 P. Fung (December 1987) Novices' predictions of Prolog's control flow: A report on an empirical study
- 36 M. Elsom-Cook and F. Spensley (April 1988) Knowledge representation in a tutoring system for procedural skills.

- 37 F. Spensley & M. Elsom-Cook (April 1988) *Dominie: Teaching and Assessment Strategies.*
- 38 M. Baker (May 1988) *Tutoring with Incomplete and Uncertain Knowledge*
- 39 R. Moyses (May 1988) *Multiple Viewpoints for Intelligent Tutoring Systems.*
- 40 P. Whalley (May 1988) *Hyper Technic - a graphic object-oriented control environment.*
- 41 P. Whalley (June 1988) *Cued recall as a Measure of argument integration.*
- 42 L. Alpay (June 1988) *A Survey and Examination of Intelligent Tutoring Systems in Medicine*
- 43 M. Elsom-Cook (July 1988) *Introduction to the ECAL system.*
- 44 F. Spensley (July 1988) *Dominie: Trainer Interface.*
- 45 M. Baker (July 1988) *An Architecture of an Intelligent Tutoring System for Musical Structure and Interpretation.*
- 46 E. Scanlon and R. Sibbitt (April 1988) *Proceedings of CAL Conference 88.*
- 47 L. Alpay (September 1988) *Medical Problem-Solving and Intelligent Tutoring Systems: Proposed Research Directions.*
- 48 S.A Cerri, M.T. Elsom-Cook, M. Leoncini (September 1988) *TRILL: The Rather Intelligent Little Lisper.*
- 49 R. Evertsz, S. Hennessy & R. Devi (September 1988) *GADL: a Graphical Interface for Mental Arithmetic Algorithms.*
- 50 P. Fung, (October 1988) *A formalisation of novices' errors in Prolog programs.*
- 51 M. Baker, (October 1988) *Arguing with the tutor: a model for tutorial dialogue in uncertain knowledge domains.*
- 52 (October 1988) *Information Technology in Education: Conceptual change in Science.*
- 53 T. O'Shea, C. O'Malley & E. Scanlon (October 1988) *Magnets, Martians and Microworlds: Learning with and Learning by OOPS.*
- 54 (November 1988) *Directory of Research 1988.*
- 55 P. Fung (November 1988) *Automated Diagnosis of Prolog control flow errors: a first evaluation.*
- 56 R. Mason (December 1988) *The use of Computer-Mediated Communication for Distance Education at the Open University, 1988.*
- 57 A.D.N. Edwards & P. Grove (December 1988) *A User's Guide to the Voice Screen Reader.*

