DVItoVDU User Guide

Version 1.0 for VAX/UNIX
Andrew Trevorrow, July 1986

This document explains how to use DVItoVDU, a TEX page previewer. The program allows you to look at TEX-generated DVI pages on a variety of commonly available VDUs. It is assumed you are already familiar with TEX under VAX/UNIX.

Contents

University of Adelaide Computing Services

Introduction.

Let's assume you've just run TEX and created a file called `foo.dvi`. Even if TEX did not find any errors, you cannot be sure that the document is formatted correctly. There might be bad line breaks, bad page breaks, alignment problems, pages too big for your printer's paper, missing fonts—not to mention spelling mistakes.

The lack of visual feedback is one of the major disadvantages of using a document compiler like TEX. Without a previewing program, the only way to check `foo.dvi` is to continue through the TEX cycle and look for errors in the printed output. This can be time consuming and is a waste of paper. Now you can use DVItoVDU to look at `foo.dvi` on a variety of commonly available terminals.

DVItoVDU is an interactive program; you can enter commands to select a particular page for display, look at the overall format of the entire page and then request a smaller region for closer examination. The manner in which the page is displayed can be varied from a full, accurate representation to a terse, fast display for when fine details are unimportant. All the commands are described in detail in a later section.

To look at `foo.dvi`, just type '`dv foo`'. Some command options may be necessary if DVItoVDU is to work properly. In particular, the `-v` option must correctly describe the type of VDU you are using.

Command options.

The `dv` command has a number of options, each of which must be followed by a value. If the same option appears more than once then the last value will be used. If a particular option does not appear then a default value will be used:

option	default value	meaning
`-v`	value of `TERM`	type of VDU
`-r`	`300`	printer resolution in pixels per inch
`-x`	`8.3in`	assumed paper width
`-y`	`11.7in`	assumed paper height
`-m`	use the DVI magnification	replace the DVI magnification
`-f`	`/usr/local/lib/tex/fonts`	font directory containing `pxl` files
`-d`	`/usr/local/lib/tex/fonts/amr10.1500pxl`	dummy font used if one is missing
`-h`	`/usr/local/lib/dv/dvitovdu.hlp`	help file read by `?` command

The last three options are really for system wizards; their default values have been set up so that most users need never worry about changing them.

Let's look at all the options in more detail. Each option is followed by a value of a certain type: *string* is a string of alphanumeric characters, *i* is a positive integer, *file* is a file or directory, and *dimen* is a positive integer or real number followed by a two-letter unit: `in`, `cm`, `mm`, `pc`, `pt` or `px`. (Most of these should be familiar from TEX. DVItoVDU provides an additional unit, `px`, which stands for "paper pixels". These two-letter sequences can later be used as commands to change the units of dimensions—see page 7 for details.)

`-v` *string*

This option explicitly tells DVItoVDU what type of VDU you are using. The value of the `TERM` environment variable is used if `-v` is not given. For example, if you're using a VISUAL 550 terminal and `TERM` does not equal `vis550` then you'll need to type '`dv -v vis550 foo`'.

The current version of DVItoVDU will accept the following *string* values in upper or lower case:

`AED483`	(AED 512 with 512 by 483 screen)
`AED512`	(AED 512 with 512 by 512 screen)
`ANSI`	(any ANSI compatible VDU; synonym = `VT100`)
`REGIS`	(any ReGIS compatible VDU; synonyms = `GIGI`, `VK100`, `VT125`, `VT240`)
`VIS240`	(VISUAL 240; synonym = `VIS241`)
`VIS500`	(VISUAL 500)
`VIS550`	(VISUAL 550)
`VT100132`	(any VT100 compatible VDU in 132 column mode)
`VT220`	(VT220 using down-loaded chunky graphics)
`VT640`	(VT100 with Retro-Graphics)

It is assumed your terminal has been set up to obey XON/XOFF flow control. This is the normal setting for terminals connected to a VAX so don't worry about it unless problems occur while DVItoVDU is displaying a page.

Some VDUs require special settings for DVItoVDU to work properly:

- The `VIS500` and `VIS550` terminals have function keys which control how information on the screen is to be displayed. DVItoVDU assumes graphic images and alphanumeric text can be seen at the same time. To ensure that this is possible, you may need to

  — hit the `SET-UP` key
  — hit the `F6` function key until `F6=BOTH` appears
  — hit the `SET-UP` key to return to normal operating mode.

  You can do these steps while DVItoVDU is running. Both terminal types should also have the Scale Factor set at 3 : 4.

- The `REGIS` VDUs use ANSI escape sequences to update the dialogue region, and the `VIS550` VDU is assumed to be emulating a VT100. Change the appropriate `SET-UP` value if your screen becomes full of junk.

- If you want to select `VT100132` you have to switch to 132 column mode yourself. The `VT220` VDU will, however, automatically switch to 132 columns (and then to 80 columns when you quit).

`-r` *i*

DVItoVDU treats the imaginary sheet of paper on which a DVI page will appear as a two-dimensional array of tiny dots called paper pixels. *i* is a positive integer that defines the number of paper pixels per inch (horizontally *and* vertically). This value should match the resolution of the device that will be used to print your document. The Apple LaserWriter, for instance, has a resolution of 300 dots per inch.

`-x` *dimen* and `-y` *dimen*

These two options define the width and height of the paper upon which your document will eventually be printed. Note that their default values specify A4 paper (`8.3in` wide and `11.7in` high). Every time you select a page, DVItoVDU uses these values to check that all the printed material will fit on the paper.

`-m` *i*

This option allows you to replace the magnification used in the DVI file with some other value; *i* is a positive integer 1000 × the desired magnification. The given value should be chosen carefully so that the new font sizes still correspond to existing PXL files.

Unless you know exactly what you are doing you should avoid changing the DVI magnification, especially if your TEX source file uses `\magnification` *and* `true` dimensions. You should only supply a replacement magnification if you intend to print the DVI file with the same override.

`-f` *file*

DVItoVDU gets all its font information from PXL files. These files should be kept within the directory specified by this option.

`-d` *file*

TEX gets all its font information from TFM files. For each TFM file there are usually a number of corresponding PXL files, each of which contains the character shapes for a font at a particular size. Although TEX allows you to scale a font to virtually any size, it is obviously impossible to provide an infinite number of PXL files. That is why it is best to stick to the pre-defined `\magstep` values when scaling fonts—you're much more likely to stay within the range of existing PXL files.

DVItoVDU will warn you if your document attempts to use a non-existent PXL file. Rather than abort, it will load the PXL file specified after `-d` and continue so you can look for more errors. Paragraphs using this dummy font are likely to have ragged right margins.

The quickest way to check your DVI file for missing fonts is to type the `S` command. The resulting display will indicate any PXL files that do not exist. You should not attempt to print a DVI file using such fonts.

`-h` *file*

The `?` command reads the help file specified by this option. The default help file contains a brief summary of all the commands.

You might like to include some lines like these in your `.login` file:

```
alias look500  'dv -v vis500'           # for use on a VISUAL 500
alias lookland 'dv -x 11.7in -y 8.3in'  # for rotated A4 paper
```

The dialogue region.

If your command line is correct and if the `-v` value matches the type of terminal you're actually using, then DVItoVDU will clear the screen and display

```
Total pages=n   DVI page=0   TeX page=[0]   Next=>   Terse
Window at (h,v) wwd by wht   Page at (minh,minv) pwd by pht   IN

Command:
```

These lines represent the "dialogue region"; the rest of the screen is called the "window region" and should be blank at this stage.

The top two lines show status information. The first status line shows:

- n, the total number of pages in the DVI file.
- The current DVI page and its corresponding TeX page counters. Particular pages can be selected by their position in the DVI file (1 to n) or by the value of their TeX counters.
- The direction in which the `N` command will select consecutive pages; initially `>`. The `>` and `<` commands allow you to move forwards or backwards through the DVI file.
- The current display mode; initially `Terse`. The `T`, `B` and `F` commands allow you to switch between `Terse`, `Box` and `Full` display modes.

The second status line shows:

- The current location and size of the window. h and v are horizontal and vertical coordinates that define the current paper position of the top left corner of the window region. The following section has details on the coordinate scheme used by DVItoVDU. The `W` command allows you to move the window to an absolute position; the `U`, `D`, `L` and `R` commands allow relative positioning. *wwd* and *wht* represent the current width and height of the window region; their values are changed by the `H` and `V` commands. The initial window size depends on the `-r` value and the size and resolution of your VDU screen.
- The location and size of the "page rectangle". *minh*, *minv*, *maxh* and *maxv* define the smallest rectangle containing all the rules and characters in the current page. (*minh*,*minv*) is the top left corner of this rectangle; it is *pwd* units wide and *pht* units high.
- The current units; initially inches. All numbers shown in this line are dimensions in terms of these units. The two letters shown at the end of the line correspond to the commands that allow you to switch between inches (`IN`), centimetres (`CM`), millimetres (`MM`), points (`PT`), picas (`PC`) and pixels (`PX`). Pixel values are shown as integral numbers and represent exact dimensions; all other values are shown as real numbers rounded to one decimal place.

Until a page is selected, most of the status values are meaningless and set to zero.

The third line is initially blank. DVItoVDU displays messages of various kinds in this line. Some of these messages appear only briefly but may convey helpful information. Others are more important and indicate some sort of problem, such as an invalid command or a page that won't fit on the paper; in these cases DVItoVDU will prompt you to hit the `RETURN` key before continuing.

The last line in the dialogue region is for entering commands. The first thing you normally want to do is choose a particular page for display. For example, typing '`1`' will select the first page in the DVI file. Many commands can be entered in the one command line. Hit `RETURN` to execute the command(s).

The window region.

The number of paper pixels per inch is given by the `-r` option. A pixel can be either black (corresponding to a tiny blob of ink) or white (no ink). A typical DVI page contains characters from one or more fonts, and perhaps a few rules. A rule is simply a rectangular region of black pixels, usually in the shape of a thin horizontal or vertical line. A character is usually a more complicated pattern of black and white pixels. Every character and rule has a paper position—or "reference point"—defined by a pair of pixel values (h,v) where h is the horizontal coordinate and v is the vertical coordinate. DVItoVDU uses a coordinate scheme in which the position (0,0) is a pixel one inch in from the top and left edges of the paper. Vertical coordinates increase down the paper, horizontal coordinates increase to the right. Confused? The illustration on page 9 may help clear things up.

The window region is used to view the current DVI page. DVItoVDU treats this region of the VDU screen as a two-dimensional array of dots called "screen pixels" (to distinguish them from the paper pixels described above). A screen pixel is the smallest possible area on a VDU screen that can be drawn or erased; the greater the number of screen pixels in a given area, the higher the resolution of the VDU.

The initial size of the window region depends on the VDU; the width and height are set so that each paper pixel corresponds to exactly one screen pixel. The most accurate representation of a page will occur at these values since neither horizontal nor vertical scaling is necessary:

VDU	initial window size (width by height in pixels)
`AED483`	512 by 442
`AED512`	512 by 471
`ANSI`	80 by 20
`REGIS`	768 by 400
`VIS240`	800 by 500
`VIS500`, `VIS550`	1024 by 688
`VT100132`	132 by 20
`VT220`	132 by 100
`VT640`	1024 by 650

Use the `H` command to change the width (the Horizontal size), and the and `V` command to change the height (the Vertical size). Try to avoid weird aspect ratios: '`H0.1 V100`' is a perfectly legal command string, but produces a very distorted display! The higher the resolution of the VDU, the greater the accuracy of such scaled displays. The command string '`HV`' will restore the window size to the above unscaled dimensions.

Note the very low resolution of the `ANSI` and `VT100132` VDUs. Since the individual dots making up their screens cannot be turned on and off, DVItoVDU has to define a screen pixel to be an entire character position. An `ANSI` screen typically consists of 24 lines of 80 characters, therefore the initial window region is 80 pixels wide and 20 pixels high (the top 4 lines are used for the dialogue region). At more useful window sizes the resulting displays will be extremely crude. Nevertheless, a variety of formatting errors can still be detected on such terminals—just don't try proofreading your document!

The size and location of the window region are automatically set every time a page is selected. DVItoVDU tries to show as much of the paper (and presumably the page) as possible, but without too much distortion. After comparing the shape of the paper with the shape of your VDU's *initial* window region, and depending on the location of the page, DVItoVDU may show the entire paper, or the top or bottom half, or the left or right half. If any part of the page is off the paper then the entire paper *and* the entire page will be shown. Since most paper sizes have portrait dimensions (width < height), and most VDU screens have landscape dimensions (width > height), DVItoVDU will normally show the top half of a sheet of paper containing the selected page.

The commands.

In response to the '`Command:`' prompt you can enter one or more of the following commands in upper or lower case. Multiple commands are processed in the order given but the window region is only updated, if necessary, at the end. For example, if you type '`NFD`' then DVItoVDU will get the Next page, switch to Full display mode, move the window Down, and only then display the page. If an illegal command is detected, any further commands are ignored. Most commands consist of only one or two characters; some can be followed by parameters. Spaces before and after commands and parameters are optional.

MISCELLANEOUS.

`?` Display brief help on the available commands.

`S` Show the option values being used, as well as statistics about the number of fonts, characters and rules used on the current page. You'll also be warned about any PXL files that don't exist.

`Q` Quit from DVItoVDU. (`^C` will return you to the '`Command:`' level and `^Z` will suspend `dv`.)

SELECTING A PAGE.

i Select the *i*th DVI page. *i* must be a positive integer from 1 to *n* where *n* is the total number of pages in the DVI file.

`[`i_0`.`i_1`.` ⋯ `.`i_9`]`

Select the DVI page whose TEX page counters match the given specification. i_0 to i_9 are integers separated by periods. Any number of these integers may be absent and trailing periods may be omitted. An absent integer will match any value in the corresponding counter. If more than one DVI page matches, the lowest will be chosen. E.g., `[]` is equivalent to `[.........]` and will select the first DVI page, even though the request matches every possible page.

TEX stores the values of `\count0,\count1,...,\count9` in each DVI page. Plain TEX only uses `\count0` to control page numbering; the remaining counters are set to zero. Some macro packages may use the other counters for section or chapter numbering; e.g., `[2.5]` might mean "select page 2 from chapter 5". When showing the current TEX page in the top status line, DVItoVDU always shows the value of `\count0`, but trailing counters with zero values are not shown. Note that plain TEX uses negative values in `\count0` to indicate page numbers in roman numerals (such as in a preface). In most documents, the *i*th DVI page will be the same as TEX page `[`*i*`]`.

`N` Select the Next DVI page according to the current direction (`>` or `<`). Before any page has been requested, `N` will select the first DVI page if the current direction is `>`, or the last DVI page if the current direction is `<`.

`>` Future `N` commands will now select DVI pages in ascending order. If *i* is the current DVI page, an `N` command will get page $i + 1$ (unless *i* is the last DVI page).

`<` Future `N` commands will now select DVI pages in descending order. If *i* is the current DVI page, an `N` command will get page $i - 1$ (unless *i* is 1).

CHANGING THE WAY A PAGE IS DISPLAYED.

The `T`, `B` and `F` commands provide three different ways of displaying a page. The choice of display mode depends on the capabilities of your VDU and the level of detail you wish to see. Whatever the display mode, the window region is always updated in the following manner: Visible paper edges are drawn first, followed by visible rules. Visible characters are then shown on a font by font basis; those fonts with the least number of characters on the page are drawn first.

While the window region is being updated, DVItoVDU will check to see if you've typed something at the keyboard. You can hit the `RETURN` key to abort the display, or you can change the display mode by hitting the `T`, `B` or `F` keys (without hitting `RETURN`). There might be a slight delay between the time you hit a key and the time something actually happens.

`T` Display a Terse representation of characters. On high resolution VDUs the TEX text fonts should be quite readable; characters will be in approximately the right position and may even be about the right size. The display on low resolution VDUs can be a bit confusing; characters get overwritten if they are too close together. Don't worry too much if you suddenly see many more spelling mistakes than usual! Note that the text fonts will all look alike. The only way to distinguish between different fonts is to note the order in which characters are displayed (the less used fonts are drawn first). Most VDUs assume all characters come from a TEX text font and then attempt to map them into similar-looking ASCII characters. Characters from non-text fonts, such as math symbols, will usually appear incorrect.

`B` Display Box outlines of the smallest rectangles containing all black pixels in characters. The reference point of a TEX character is usually located near the bottom left corner of one of these boxes. Box mode is intermediate in speed between Terse and Full modes; it is a quick and accurate way of checking the alignment of entries in a table, for example.

`F` Display a Full representation of all pixels in characters. This display is the most accurate but may take some time; hit `RETURN` or switch to Terse or Box mode if you get bored. The fact that the display mode can be changed *while the window is being updated* can be very useful. A good compromise between speed and accuracy is to start off in Full mode so that math symbols and any other special characters are displayed correctly, and to switch to Terse mode (by hitting `T`) when the bulk of the text begins.

CHANGING THE UNITS OF DIMENSIONS.

All the numbers in the second line of the dialogue region are dimensions in terms of the units shown at the end of the line. The parameters following some commands are also dimensions in terms of these units. Unlike the dimensions in TEX, you don't explicitly type the units when you need to specify a dimension after a DVItoVDU command—simply enter an integer value or real value. This value is rounded up internally to the nearest paper pixel based on the current units and the conversion factors shown below. For example, '`CM H5 V3.4`' will change the units to centimetres and set the window size to be 5 cm wide and 3.4 cm high.

`IN` Dimensions are assumed to be in terms of inches (`-r` defines the number of paper pixels per inch).

`CM` Dimensions are assumed to be in terms of centimetres (2.54 cm = 1 in).

`MM` Dimensions are assumed to be in terms of millimetres (10 mm = 1 cm).

`PC` Dimensions are assumed to be in terms of picas (1 pc = 12 pt).

`PT` Dimensions are assumed to be in terms of points (72.27 pt = 1 in).

`PX` Dimensions are assumed to be in terms of paper pixels.

MOVING THE WINDOW.

The window region can be moved to any position over the current page. You will be told if the window moves entirely outside the page rectangle defined by *minh*, *minv*, *maxh* and *maxv*. If this does happen, movement is restricted to *just outside* the edges to make it easier to get back over the page using only the `U`, `D`, `L` and `R` commands. Note that the location of the window is automatically set every time a page is selected. This position will normally be the top left corner of the paper; i.e., $(-1, -1)$ in inches. The parameters h and v are dimensions ranging from -480 inches to $+480$ inches (for those of you TEXing billboards).

`W` h`,`v

Move the Window region's top left corner to the given paper position. h is the horizontal coordinate, v is the vertical coordinate. If h and v are absent then the window is moved to (*minh*,*minv*), the top left corner of the page rectangle.

`U` v

Move the window Up v units. If v is absent then move up an amount equal to the window's current height.

`D` v

Move the window Down v units. If v is absent then move down an amount equal to the window's current height.

`L` h

Move the window Left h units. If h is absent then move left an amount equal to the window's current width.

`R` h

Move the window Right h units. If h is absent then move right an amount equal to the window's current width.

CHANGING THE SIZE OF THE WINDOW.

The `H` and `V` commands are used to change the width and height of the window region. It is up to you to maintain a sensible aspect ratio. The location of the window will not change unless the page becomes invisible. Note that the width and height of the window are automatically set every time a page is selected; the values chosen will depend on the paper dimensions. The parameters *wd* and *ht* are dimensions ranging from 1 pixel to 480 inches.

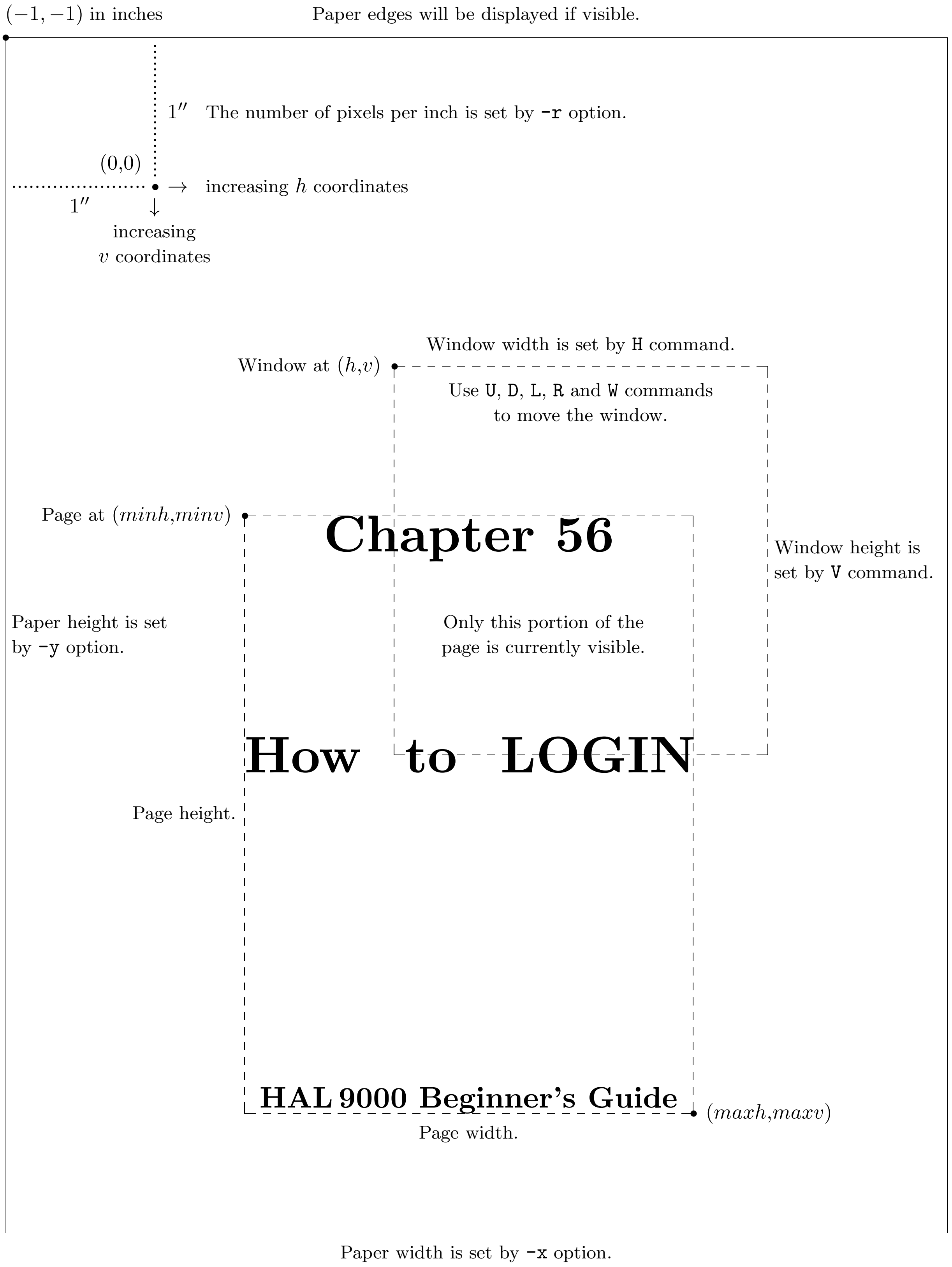
`H` *wd*

Set the Horizontal size of the window to the given width. If *wd* is absent then set window width to its initial, unscaled value.

`V` *ht*

Set the Vertical size of the window to the given height. If *ht* is absent then set window height to its initial, unscaled value.

An illustration.


$(-1,-1)$ in inches
Paper edges will be displayed if visible.
1″
The number of pixels per inch is set by -r option.
(0,0)
→ increasing h coordinates
1″
↓
increasing
v coordinates
Window width is set by H command.
Window at (h,v)
Use U, D, L, R and W commands
to move the window.
Page at $(minh,minv)$
Chapter 56
Window height is
set by V command.
Paper height is set
by -y option.
Only this portion of the
page is currently visible.
How to LOGIN
Page height.
HAL 9000 Beginner's Guide
$(maxh,maxv)$
Page width.
Paper width is set by -x option.

Some examples.

For clarity, commands are separated by spaces and typed in upper case; neither condition is mandatory.

```
< N N N
```

If these commands are given before a page has been requested, they will select the 3rd last page for display (assuming there are at least 3 pages). `<` tells DVItoVDU to select pages in reverse order; the following `N` selects the last page; the remaining `N`s go back another 2 pages. Note that only the 3rd last page will be displayed. The intervening pages are still processed though, and you'll be warned about any problems with them (such as a page off the paper).

```
1 F W H V
```

This command sequence is very useful for looking carefully at the results of a small TeX experiment. You might, for instance, want to look closely at the appearance of two characters separated by a small `\kern`. `1` selects the first DVI page (usually the only page in such experiments); `F` sets the display mode to full; `W` moves the window to the top left corner of the page; `H` sets the horizontal size of the window to its unscaled value and `V` sets the vertical size of the window to its unscaled value. Remember that this unscaled window size produces the most accurate display, since each screen pixel corresponds to exactly one paper pixel.

```
R9999 L D9999 U
```

Sometimes you want to move quickly to the right edge of the page to have a look at line breaks, or you might want to go to the bottom and look at where the page was broken. This particular command sequence will move the window's *bottom right* corner to the bottom right corner of the current page; i.e., (*maxh*,*maxv*). `R9999` moves the entire window to the right of the page, but only just; `L` then moves the window left by its current width so that the *right* edges of the window and the page coincide; `D9999` moves the entire window below the page, but only just; `U` then moves the window up by its current height so that the *bottom* edges of the window and page also coincide.

```
IN W-1,-1 V12 H16 T
```

It is often useful to get an overview of the positioning of the page within the entire paper. These commands will do just that on all the VDUs currently implemented, assuming the default A4 paper size. (The `S` command shows the paper dimensions if you can't remember them.) `IN` sets the current units to inches just in case they were something else; `W-1,-1` moves the window to the top left corner of the paper (recall that (0,0) is one inch in from the top and left paper edges); `V12` sets the window height to slightly more than the default paper height; `H16` sets the window width to a value that will ensure the paper shape has roughly the right proportions (most VDUs have screens with widths greater than their heights); `T` sets the display mode to terse since you're probably not concerned with finer details when looking at the entire paper.

```
[.56] W IN H2.5 V2.6 R1 U1 F
```

This is the likely command sequence that led to the window display shown in the illustration on the previous page. The author has been clever enough to define a `\chapter` macro that sets `\count1` to the given chapter number. `[.56]` will thus select the appropriate DVI page (`\count0` is ignored); `W` moves the window to the top left corner of the page; `IN` sets the current units to inches; `H2.5` and `V2.6` set the window dimensions; `R1` moves the window right 1 inch; `U1` moves the window up 1 inch; `F` sets the display mode to full.

⋆ ⋆ ⋆