```
.pl 60
.mb 3
.po 0
.op
```

THE LABELLER HANDBOOK

Release 2
August 1989

This document has been prepared by:
L.Chiappetti, IFCTR

The software described in this document has been contributed by the following persons :

L.Chiappetti IFCTR, CNR Milano

```
.he$PAG The Labeller Handbook (August 89)
Contents
.pa
.pn 1
.foContents
```

Table of content

.pa

Amendment history

February 1986	v 1.0	release 1 of software associated documentation limited release
November 1988	v 2.0	unofficial release of new version
August 1989	v 2.0	release 2 of software associated documentation official release

.he$PAG The Labeller Handbook (release 2)
Page #
.pn 1
.pa
.fo The Labeller Handbook (August 89)
Section 0

0. Foreword

This document is the users' guide to the release 2 of The Labeller. Details on the new release, issued in August 1989, are contained in section 0.1. The previous release of The Labeller was issued in February 1986, and a semi-official documentation briefly afterwards. Section 0.2 contains the original foreword to release 1.

0.1 Foreword to release 2

The present version of this handbook assumes the reader has some familiarity with the plotting facilities at IFCTR, and possibly with the usage of the previous release of The Labeller itself. In this respect reading of the foreword to the previous release of the handbook (now section 0.2) may be helpful.

A general introduction to The Labeller is otherwise contained in section 1 below. This foreword mainly points out the changes and improvements since the previous release.

First of all, the PABLO plotting program, to which The Labeller is complementary has now its own extensive documentation (L.Chiappetti, The PABLO Handbook, released in December 1988, with minor updates in February and August 1989).

Secondly, a new program, the Plot Editor, now exists and provides an interactive interface to both PABLO, The Labeller and the contouring program MCONT, with automatic generation of the command files. Such program has been in semi-official existence since February 1988 and is currently being official released with notable improvements and with related documentation.

Therefore The Labeller may be used autonomously, or driven by the Plot Editor, which takes care of calculating nasty things like scales, tic steps etc. The usage of the Plot Editor is normally recommended, but manual editing of the Labeller command files may help for "special effects" not supported by the default choices of the Plot Editor.

Here follows a list of the major improvements in this second version of The Labeller with respect to the previous one :

Log scales covering a non-integral number of decades are now supported, consistently with PABLO Version 2.

New formats for numeric labels are supported (including exponential format, a choice of decade (e.g. 10^{-5}) or log (e.g. -5) value for log scales and suitable default provisions)

.cp3

New options allow to control the positioning and orientation of numeric labels with respect to tics and axes.

Axis captions allow now escape sequences. Furthermore a new **OFFSET** command allows to control caption positioning.

A new **DIRECTION** command allows to control text orientation directly, and not only through escape sequences.

The **TEXT CURSOR** command now gives the correct, reproducible position (bug has been fixed).

Paper is now by default not ejected by the plotter at the end, but this can be changed by the new **EJECT** command.

0.2 Foreword to release 1

You all have got somewhat familiar with the PABLO plotting program (although it is totally undocumented, and not very helpful, even if totally flexible and versatile). You also have noticed that PABLO (actually the graphics library used by it, do not blame the authors of PABLO !) uses awful software-generated characters (how often have you put a 0 instead of an O - by the way in PABLO an O looks like a 0 and a 0 like an O - to improve the aestethics of your text ?). Also the axis routine (which has been greatly improved on the original already !) has an unpleasant attitude about writing tics only every inch, using funny formats for the labels (and exponents also), overlapping the origin labels for the x- and y-axis, etc. etc.

Therefore the need of a program like The Labeller, which just makes labels with tics at an user defined step, in many nice character fonts, allows lower and upper case, user controlled formats, different scales on different axes of the same frame, etc. etc. Although The Labeller is generally used in conjunction with PABLO, you can use it also to produce text outputs (e.g. for transparencies). In this respect it is different from BRUNO (try it if you do not know it). BRUNO is very helpful and easy to use, and allows interactive editing of a text "slide". Also BRUNO has xery nice software character fonts, like Roman and Gothic. On the other hand it is very difficult to use BRUNO to label a pre-existing plot (it was not written for that). On the other hand The Labeller allows an easier control of things like exponents and subscripts, change of pen colour in-between a text string, change of orientation (you can write a string at any angle, not just horizontally) etc., all using the normal HP editor, even if at the cost of the use of some (not so) complicated escape sequences.

.cp10
.fo The Labeller Handbook (August 89)
Section 1

<u>1. Introduction</u>

The Labeller (**LABLER**) is a plotting program operating on an HP computer with RTE 6, equipped with an HP 7550 8-pen plotter. The main purpose of the program is to annotate a pre-existing plot (typically one produced with the PABLO plotting program) with a set of neat labels in an user defined format, using all the opportunities built in a 7550 plotter (like slanted characters, hardware character fonts, etc.).

The Labeller is not intended to be portable, therefore it is written (in HP Fortran FTN7X) using a number of system-dependent peculiarities and facilities (like CALL EXEC and CALL REIO). All graphic calls are made directly generating the proper HP-GL commands (as a consequence the program outputs only to the 7550 plotter, and to none of the other graphics devices).

Persons interested in programming details are welcome to have a look at the program listing. No programmers' sheets are provided, as the extent of comments in the listing is felt to be adequate documentation. The source code of The Labeller is in file **&LABLR::LC**, and is self-contained, in the sense that all external non-system references (subroutines) are resolved within the same file. However the Fortran code for the two main common blocks is now contained in include files **&LABCO::LC** and **&LABC1::LC**.

The program can be reloaded when necessary with the procedure :

```
     TR,*CLK2,&LABLR,B,LABLER
```

.cp 7

2. Users' instructions

You may use The Labeller before or after you produce your plot (you may also use The Labeller as a standalone program to produce a nicely plotted text, like a transparency, although better results are obtained with other programs, e.g. BRUNO), but you should note the following points.

If you intend to annotate a PABLO plot, you should produce it with no labels (i.e. use a "character size" of 0.0, which also implies no tics). The same applies to MCONT contour maps.

As a prerequisite, you must be reasonably familiar with the operation of the 7550 plotter, in particular with paper loading, and the use of the front panel keyboard. Refer to the plotter manuals, or ask someboby.

The plotter must be switched on, cleared (do a **CN,38** as first thing after you have switched the plotter on) and paper must be loaded (if paper is not loaded The Labeller hangs up until you load the paper).

.cp5
.fo The Labeller Handbook (August 89)
Section 2

The plotter must also be set to the right paper orientation (PABLO leaves the paper orientation unchanged, but other programs may not do so; note that the Plot Editor automatically sets the correct orientation you have specified to it.) To change the paper orientation, press the **NEXT DISPLAY** key until you get **ROTATE-0** or **ROTATE-90**. You can then use the **ROTATE** key to toggle between the two rotation possibilities.

Also the pens you intend to use must be in the carousel, and the proper carousel (P for normal pens and T for transparencies) must be loaded. You may set the speed and force of the pens, using the appropriate keys (refer to the plotter manuals). I recommend lowest speed (10) and force (1 or 2) in all cases.

Use everyday pens and xerox paper for tests, use new pens and HP smooth paper (available, though normally safely stowed away: ask Bruno or Lucio) for final plots (the results are much better). Also thick (0.7) pens are available.

You may use the auto-feed option, or load the paper manually. Note that The Labeller does not unload the paper at completion (unless you have issued an **EJECT ON** command): remember that in auto-feed mode unloading means paper is <u>ejected</u>. You should note that the plotter position reproducibility is quite good, while your manual repositioning (if you have removed the paper) generally is not. Therefore, for best results:

.cp 2

load the paper manually (and preferably run **LABLER** before you run your plotting program : this is what the Plot Editor does, for instance)

.cp 2

if the first program unload the paper (which is the case of PABLO and MCONT in all instances, and of **LABLER** only if EJECT ON has been issued), load it again (just press **LOAD**, dont' touch the paper)
then run your plotting program or The Labeller (accordingly to what you have done before).

For tests it is also acceptable to load the paper automatically, then switch the auto-feed off (press **AUTO-FEED** until the asterisk disappears), do the first plot or **LABLER** run, reload the paper manually, and do the second plot (or **LABLER** run as appropriate) : note however the alignment of tics and labels with the plot will be worse in this case.

In the case you are interested of producing many plots, say with the same layout and different data files, you may take advantage of the auto-feed capability of the plotter and have the whole run done automatically while you are away as follows :

Prepare standard command files for PABLO and The Labeller (e.g using the Plot Editor). Make sure the **LABLER** command file does not contain an EJECT ON command.

Prepare a FMGR transfer file trfile, which contains a "unit" consisting of the **:RU**, commands for **LABLER** and PABLO (in this order, so that paper is not ejected in the meanwhile), followed by a number of **:RU,EDIT** commands which change your standard command files for the next plot (e.g. change the file name and plot title). Duplicate and edit a number of such "units" according to how many plots you want. Terminate the file with a **:TR** command.

Make sure the plotter loading tray contains enough paper.

Set the plotter into auto-feed mode.

Do **TR,**trfile : your plots will be done and labelled automatically, and a fresh sheet automatically loaded for each new plot.

Pray that the paper never jams (and repeat crushed plots as necessary).

To run The Labeller, just do :

LABLER,commandfile

where commandfile is the name of a file containing labeller commands. It is suggested to start the filename with an 'at'-sign (**@**) and end it with the letter **L** (this is not necessary at all). It is compulsory to have a command file. Labeller commands are described in the next section.

The program then runs automatically, clears the screen and echoes the line number currently being processed. Error messages may also appear (and remain) on the screen, with reference to offending line number. Errors may be input errors (usual HP runtime errors, i.e. something wrong in the file, or format errors), plotter errors (HP-GL errors, see plotter manuals, like coordinates out-of-range, etc.), or internal errors generated by the program. They are described below. Any line in the command file which does not contain a valid command is also displayed (line number only) as "unknown command",

and is simply ignored by the program.

Note that the program may be "neatly" interrupted at any time (better than aborting it with **OF**), by breaking it (press any key to get the system attention and the **S=nn?** prompt, and give a **BR,LAB**nn where nn is the terminal session number). Give time that the plotter buffer empties. When the program is terminated by break, paper is ejected or not according to the last EJECT command executed.

.cp10
.fo The Labeller Handbook (August 89)
Section 3

.cp 5

3. Labeller commands

The command file contains a number of labeller commands (one per line, except the **TEXT** and **COMMAND** commands, which may span more lines) in a plain, self-explanatory format like :

.cp 4

PEN 2   or

AXIS X 1. 1. LIN 6. 100. 25.

Commands may be followed (as in the example above) by parameters. Parameters are generally in HP free-format, and may be separated by blanks or commas. Any number of blanks is acceptable, while in the case of commas, two consecutive commas indicate a parameter is omitted and the default value should be used (this option is for numeric parameters only and has not been tested, so you should experiment yourself to see what the default is, in most cases it may just be the preceding setting for that parameter. However note no guarantee !). If you have used a comma, the next parameter <u>shall not</u> be preceded by leading blanks.

Comment lines may be freely inserted in the command file, and have an asterisk (*****) in column one.

.cp 3
Commands are used to perform the following functions : general set up, set up for an axis set, plotting actions for an axis (tics, numeric labels, text labels), plotting a

complex text (including change of pen, character font, slant, subscripts and superscripts, etc.), sending direct HP-GL commands.

The complete list of commands with the correct syntax is given in a table in appendix below. Here we just give an introductory overview.

3.1 Set-up commands

A first group of set-up commands is used to give the coordinate system. These commands are just a single keyword, specifying the units for the coordinates, to be used until the end of the command file, or until another command of the same class is given. They are just single keywords, like **INCHES**, **CM**, **MM** and **PLOTTER**. The first three are self-explanatory, the latter are plotter units of 0.025 mm. You will be generally using **INCHES** in connection with plots produced with PABLO. Note that, in conformity to PABLO, paper origin is at (0,250) plotter units (with current paper orientation). If no unit choice is made, the default is INCHES.

You may then select the pen with the **PEN** command, you may select the character size with the **SIZE** command, the character set with the **SET** command, the slant of characters with the **SLANT** command and the orientation in which text is written with the **DIRECTION** command. All these commands refer to all text produced, until another command of the same type is sent. It is also possible to change the settings in-between of a **TEXT** string (using escape sequences, described below).

Pen numbers are 0 to 8, other values are ignored (no-operation). Reference to not mounted pens is processed by the plotter (according to what described in the plotter manual).

Character sizes are specified both in x and y, in the current units (**PABLO** users : even if using INCHES, there is no immediate relation with the PABLO character sizes ! It takes a bit of experience to find a nice size, try 0.07 at first). If y-size is not specified, it is 1.38 times the x-size. If no size parameters are specified, the command is a no-operation. If no SIZE command is given at all, 0.187 x 0.269 cm is used.

Character slant and text direction are specified in degrees. No slant or direction parameter is equivalent to slant zero or direction zero (horizontal).

.cp5

Character set is specified by a number or a file name. Allowed set numbers are 0-19 and 30-49, plus -1, as specified in the plotter manual. Reference to an illegal set is processed by the plotter, according to the specifications in the manual.

A file name is used to load user-defined character sets (later to be referred at as set number -1). Character set -1 may not be invoked, if not previously loaded once by name. This option has not been tested (no user-defined character sets exists yet), though it is planned to have at least a Greek character set.

A "download in progress" message appears while reading in the character file. Other messages tell you how many characters are in the file, and what are the ASCII equivalents.

There is a table in the plotter manual (Appendix A), which gives the appearance of all hardware character sets (both with fixed and variable spacing), and the correspondence with ASCII values : e.g. in character set 7, ASCII letter P correspond to the Angstrom sign. A SET command without parameters is equivalent to request set number 0 (default). The requested character set is loaded as "standard" character set (this is in contrast to the imbedded escape sequence see below - which loads a set as "alternate").

.cp10

You may also select whether the paper is ejected at the end of the run, with the **EJECT** command. If you specify EJECT OFF or do not specify anything, paper is not unloaded. If you specify EJECT or EJECT ON, paper is unloaded. The command may occur at any place in the command file, and only the last one executed takes effect.

3.2 Axis set-up

Axis set up commands are mainly contained in the **AXIS** command itself. Exceptions are the **HOUR** and **STEP** commands, which must be called before AXIS, and the **OFFSET** command (see 3.3 below), which must be called before the LABEL command to which it applies. The AXIS command itself must be called before any other axis plotting command may be used.

HOUR or **HOUR ON** sets a flag (removed with **HOUR OFF**) to enable the axis labels to be drawn in hours,minutes,seconds (with formats like 13:30:45 etc.) with the special H format described below.

STEP indicates every how many current units a tic must be drawn for a given axis. If called without parameters or not called at all, the default is 1. (inch), which is compatible with PABLO. Note however you have the maximum flexibility to put tics (also multiple sets of tics with different size) wherever you want.

The **AXIS** command has the most complicated syntax. Note that no plotting at all is performed by this command !!! Here goes the syntax :

```
AXIS X x-origin y-origin LIN length start delta code
     Y                   LOG
```

where the second keyword may be **X** or **Y** (obvious meaning), and the fifth **LIN** or **LOG** (also obvious). If the axis is LIN then you specify start and step as delta and omit code, while if it is LOG the business is more complicate and explained below. Note that the above syntax does not imply at all all X-axes are LIN and all Y-axes are LOG !
For LOG axes code may assume three different values, and the meaning of start and delta are different in the three cases (the complication is unfortunately necessary to ensure backward compatibility with previous versions of PABLO and of the Plot Editor).

code=**OLD** corresponds to the original usage in PABLO : start is the starting decade exponent and delta the number of decades. Both numbers shall be integer. A log 2 scale extension is applied (like in the original version of PABLO). If not specified, code defaults to **OLD**, which ensures compatibility with older command files. This mode is however not recommended.

code=**DECADES** corresponds to the new usage allowed in PABLO version 2. start and delta are still the starting decade and number of decades, but are allowed to be real. Again log 2 scale extension is applied. This is consistent with the new PABLO, but is a bit awkward to use (compute the fraction of decades is annoying).

code=**EXACT** is the recommended and more friendly mode. Here start and delta are the start and end values of the axis, the more natural thing to say (if you want a scale from 1000 to 20000 it is easier to say so, than "starting decade 3" and "1.30103 decades long"). This is consistent with the usage in PABLO Version 2 where the PLT.PF lines 30-31 are set to a negative value to force an exact log scale without log 2 extension. However here The Labeller is even more friendly as it requires start and end, and not start and step.

By the way all this is totally transparent to Plot Editor users, for which also the consistency between PABLO and LABLER command files is automatically guaranteed.

The **AXIS** command means : be prepared to annotate an X- (or Y-) axis with origin at (x-origin,y-origin) (in the current coordinates, typically inches). The axis is LIN (or LOG), is length current units (again typically inches) long, the value of the label at the origin is start and we want to increment it by delta at every tic (for LIN axes, see above for LOG axes). Note that start and delta are in world coordinates (keV, erg/s, apples, km/man/year etc.). It is your responsibility to ensure consistency between the tic distance (given in the **STEP** command) and the delta in world units.
For LOG axes, there is no such problem, just specify the starting decade, and the number of decades, or whatever as described above. The consistency is automatically ensured for command files generated by the Plot Editor.

Note for PABLO users: x- and y-origin are like lines 19 and 20 of PLT.PF, length is like lines 17 or 18 (according to the axis orientation), start like lines 8 or 9. delta is like lines 10 or 11 for linear scales <u>IF the tic interval</u> <u>selected by STEP is 1 inch</u>, otherwise must be scaled appropriately.

The AXIS command does not do any plotting, but loads a common block with values likely to be used later for labels and tics. These are computed assuming X-axes run from left to right, with labels outside (below) and tics inside (upwards), and Y-axes run from bottom to top, with labels outside (left) and tics inside (right). Other arrangements may be obtained by defining AXIS at an origin offset with respect to the original. Although use of negative values for length and delta has not been tested, it is likely to work and give some interesting results for axes with unusual orientations.

.cp6

3.3 Axis plotting

Despite the name, no Labeller command exists to plot an axis. Commands instead exist to draw tics to an axis (**TICS**), write numeric labels beside the tics, or even without tics, (**NUMBER**) and to write a text label or caption (**LABEL**), eventually at a specified offset (**OFFSET**).

All such commands retrieve the necessary information about where to place the text, etc., from a common block set by the latest **AXIS** command (this command must have been called !).

The **TICS** command draws tics along the current axis at the positions set (in command **STEP** for the interval between tics, and in command **AXIS** for the origin and length). The only parameter is the tic size (in current units). No parameter defaults to half of the current character y-size. This applies to small (unit) tics for LOG scales (decade tics are 1.5 times larger). Of course the **STEP** command information is not used for LOG scales. The tic size may be specified as a negative number to draw tics on the outside part of an axis (i.e. below an x-axis or left of a y-axis). Contrary to HP-GL, tics on both sides of the axis are not allowed (may be drawn separately with two TICS commands).

Hint: you may draw tics of different heights at different steps (i.e. small tics every 5 world units, and large tics every 10), by issuing the appropriate **STEP** command, followed by **AXIS** and **TICS**. (It is necessary to repeat the AXIS command only if you want to plot labels too).

Note that for LOG scales of less than one decade, tics are drawn at 1,2,3,4...9 (in whatever decade, e.g. at 40 50 60 for a log scale from 35 to 65) in an unique size. Also (in the EXACT and DECADE mode) if there are more then 10 decades only big tics at decades are drawn.

The **NUMBER** command may be called before or after or independently of **TICS**. It will place a numeric label at the computed tic position. If e.g. in your **AXIS** command you have specified start at 15. and step as 2.5, the numeric labels will be 15., 17.5, 20. etc. Note that an axis length needs not to be an exact multiple of the interval between tics. You may also specify a length smaller than the actual one, to avoid the last label to be written (e.g. when labelling adjacent frames, where the last label of a frame may overlap the first one of the next frame). The above does not apply to LOG scales in the OLD or DECADE mode, which are always extended by 20% of the last decade.

.cp3

The **NUMBER** command has now a longer syntax than before:

NUMBER format alignment orientation.cp5where the alignment parameter specifies how the numeric labels are justified (aligned) with respect to the tic position, and orientation specifies how the numbers are written. If omitted, both parameters assume suitable default values.

The possible values for alignment are:

CENTRED : labels are centred at the position of the corresponding tic (nice, but the first and last label may protrude outside of the frame)

PROTECTED : this is the default setting : labels are centred as above, but the first and last are forced to remain within the frame boundaries (good choice when plotting adjacent frames)

GRADUAL : this correspond to the setting used in the previous release, where label position was gradually adjusted to protect against protrusion (not so nice as current solutions)

LEFT : labels are left justified to tic positions

RIGHT : labels are right justified to tic positions

The possible values for orientation are:

NORMAL : labels are written horizontally for x-axis, and vertically for y-axis (this is the default, corresponding to the usage in the previous release)

ROTATED : labels are rotated by 90 degrees (then vertical for x-axis and horizontal for y-axis ; the latter case may improve graph readability)

The format parameter is optional for LOG scales. Normally it may assume the following two values : **EXP** or **LOG**, and only tics corresponding to decades are annotated.

In the **EXP** case, which is also the default, and corresponds to the usage in the previous release, the label is always in the form of a "10" raised to the appropriate exponent (in smaller characters).

In the **LOG** case, the label is just the exponent (e.g. 10^{-3} is written as -3; usual for labelling scales as "log flux")

If the LOG axis has less than two complete decades (and typically less than one), not only the decade big tics are annotated, and the above formats may not be adequate. You are allowed to use the ritten at some appropriate place, centered in the axis length (as specified in AXIS command). Note that imbedded escape sequences (see below) are now allowed in the text string. Note however that if you insert escape sequences which alter the character size, the centering of the label may be affected.

The label (axis caption) is written at a position such not to overwrite the numeric labels at tics. This may not always be easy to mantain when the numeric labels are ROTATED (see **NUMBER** above), although the program keeps suitable defaults. In such case, as in any other case you may want to control the displacement of the label from its axis (the displacement along the axis is always such to centre it), make use of the **OFFSET** command.

The **OFFSET** command has a single parameter, which is the number (also real) of characters (in the current size) the label has to be offset from the axis. A positive offset indicates a label on the usual side (below an x-axis and on the left of an y-axis), but a negative offset can be used to place a label on the other side (sometimes useful when one frame has more units scales).

The **OFFSET** command shall be called after **NUMBER** (since it is there that the proper offset for the ROTATED case is set) and before the relevant **LABEL** command, and applies to all following labels, until a new **NUMBER** or **AXIS** command is called (they reset the default offset).

General hints: if you want to fully annotate a plot, you must include the following commands (example only) in your command file :

```
STEP size
AXIS X x1 y1 LIN lx sx1 sx2
TICS tsize
NUMBER format
LABEL This is the X-axis
AXIS X x1 y2 LIN lx sx1 sx2
TICS -tsize
STEP newsize
AXIS Y x1 y1 LOG ly ny1 ny2
TICS tsize
NUMBER
LABEL and this is a log y-axis
AXIS Y x2 y2 LOG ly ny1 ny2
TICS -tsize
```

where y2 = y1 + ly and x2 = x1 + lx. The AXIS and TICS commands are repeated with negative tic size, to write tics on the upper and right portions of the frame. Labels are on the lower and left part of the frame. This is just a (simple) example. You may think of more complex things, like two different scales at the top and at the bottom of the frame, a scale offset from the axis origin (provided you do all the annoying scale calculations in user units) etc. All the rest is up to you !

Consider now carefully the owing three examples of command sequences :

```
 (1)                 (2)                 (3)

STEP 1.0            STEP 1.0            STEP 1.0
AXIS .....          AXIS .....          AXIS ......
TICS 0.15           TICS 0.15           TICS 0.15
```

```
* no numbers        NUMBER .....        * no numbers
STEP 0.5            STEP 0.5            STEP 0.5
AXIS ....           * nothing here      *
TICS 0.08           TICS 0.08           TICS 0.08
* the end           * the end           NUMBER ....
```

All three draw a set of big tics every inch, and a set of small tics every 0.5 inch, and do it correctly. Case 1 is all right if you are not interested in numbers, case 2 numbers correctly the major tics, but case 3 is wrong, as it numbers the minor tics with the values corresponding to the major ones, as no AXIS command has been inserted !

Note for PABLO users: the length of one decade in inches for LOG scales in OLD mode, is given by the axis length divided a magic number, which is (guess what) :

The number of decades + 0.30103

.cp8

3.4 Text plotting

This is accomplished via a single command **TEXT**. The command has either two parameters (the x and y position of the text in current units), or is in the form **TEXT CURSOR** for manual (interactive) positioning. The command is followed by a number of lines of text (at least one), and is terminated by a line having the capital word **END** as the first three characters. The lines of text are plotted exactly as written (i.e. on the same number of lines, left justified), except for a number of escape sequences used to perform control functions.

.he$PAG The Labeller Handbook (Release 2)
Page #

In the case you use the TEXT CURSOR command, you get a prompt on the terminal ("Position pen on wished point") : you should then walk to the plotter, which is ready to be used in digitize mode. If no pen is currently loaded, it is suggested you load one (pressing one of the numeric keys, e.g. 2 for pen number 2 etc.). It will be automatically put away at the end, and will not interfere with the plotting. Then you move the pen to the wished point with the arrow keys (use the **FAST** key to move faster). When you are in the right position, press **ENTER** (as indicated by the message on the liquid crystal display).

The digitized position is echoed to the terminal, both in current user units (inches, cm, etc.) and in "revised" plotter units ("revised" just means they are note raw plotter units, but refer to the software origin at (0,250)). The value in current user units may then be written in the command file instead of the **CURSOR** keyword, to reproduce the plot later).

.cp3

Text escape sequences are useful to change pen, character set, slant, direction and character size in the middle of a string, as well as to write subscripts, superscripts or symbols. They may look cumbersome at first, and they certainly are, although they are easier to use than, e.g., HP terminal control escape sequences, since only printable characters are used.

Escape sequences are initiated <u>and terminated</u> by a single escape character, currently defined as the "at" character (**@**). In order to write a single @ in your text, this must be escaped, i.e. doubled (**@@**, like '' in Fortran format statements). The escape sequence between the @s consist of a single letter command code, followed by zero or more parameters. Parameters may be separated by blanks or commas (they are parsed by the same parsing routine used for commands), except that it is not allowed to have an intervening comma between the command code and the first parameter. Allowed escape sequences are listed in a table below and briefly explained here.

The **@S** xsize ysize**@** sequence is used to change character size (identical to command **SIZE**) for the next character to be plotted onward (until another @S@ sequence or a SIZE command).

.cp3

The **@P** pen**@** sequence is used to change pen number (identical to **PEN** command).
The **@A** angle**@** sequence is used to change the character slant (identical to **SLANT** command).

The **@C** set**@** sequence is used to change character set. It is similar but not identical to the **SET** command. The difference is that the new character set is loaded as "alternate", and co-exist with the latest set loaded by a SET command. A **@C@** sequence (without parameters) performs a switch to the "standard" set (i.e. the one loaded with the last SET command, even if you do not remember which it is). If you want just to change character set in the middle of a string and go on using it, the difference is transparent to you. However if you switch to an alternate set to plot a funny character (e.g. the Angstrom symbol) and then switch back to normal, this way is faster.

By the way, to plot the Angstrom symbol (P in character set 7) you use the sequence: @C7@P@C@. Easy, isn't it ?

The **@D** angle**@** sequence is used (mainly in front of a text string, use as imbedded sequence may have unexpected effects) to change the direction of the text <u>from now onwards</u> until the next @D@ (or **DIRECTION** command, to which it is similar). Use an angle of 90 degrees for normal vertical writing. The command **@D@** without parameters is equivalent to an angle of 0 degrees, i.e. resets horizontal writing.

.cp5

The sequences **@E@** and **@e@** are used respectively to initiate and terminate a superscript (exponent). Exponents are written in half the current character size, a quarter of character upwards. Everything between @E@ and @e@ is written as exponent.

The sequences **@I@** and **@i@** are used respectively to initiate and terminate a subscript (index). Subscripts are written in half the current character size, a quarter of character downwards. Everything between @I@ and @i@ is written as subscript.

The sequence **@*@** is used to enter symbol mode. This option is not yet available.

Note for advanced users: if you play around with direct HP-GL commands to change text writing characteristics, like e.g. reset the "label-origin" setting (LO command), you may have unpredicted results when imbedded escape sequences are used in a text string, like unexpected "carriage returns". Single text strings (without imbedded escape sequences) should be immune from problems.

.cp10

3.5 Direct HP-GL commands

This section concerns advanced users, who are familiar with HP-GL (see the plotter manual), and want to use HP-GL commands for particular plotter enhancements (e.g. filled polygons, hatching, etc.). A single **Labeller** command caters for all possible HP-GL command sequences (of course you must note that HP-GL commands work in absolute plotter units only).

.cp5

The command is **COMMAND**, and is followed by a number of lines, each one containing one or more HP-GL commands (separated by an allowed separator, see plotter manual). The last line must contain in the first three characters the word **END** (the same as for the TEXT command).

4. Changes from version 1 and future improvements

The following improvements, planned but not implemented in version 1, are now available (see also the foreword):

The possibility of using escape sequences (hence super scripts and subscripts) in labels.

The possibility of exponential format for numeric labels.

.cp3

The possibility of different formats for LOG scale numeric labels (i.e. have the exponent only, instead of 10 raised to the exponent, also horizontal letters for Y-axis, also have tics for decades, or decades and half-decades only; the latter will be useful for scales with many decades).

The possibility for the user to control the offset used to center (or not to center a numeric label with respect to the tics).

The inclusion of an actual command to reset the direction of a text (instead of the current escape sequence).

The following improvements, planned in version 1, still remain unimplemented :

The inclusion of an actual command to draw an axis.

The generation of user-defined character sets for the commonly used symbols.

.cp10
.fo The Labeller Handbook (August 89)
Appendix 1

Appendix 1 : Labeller commands

* comment   used to indicate comment lines

INCHES
PLOTTER
CM
MM

HOUR   in the forms: HOUR = HOUR ON, HOUR OFF

STEP size   with default size = 1.

PEN pen   with pen in the range 0-8 (no or illegal argument have no effect)

SIZE x y   also accepted SIZE x (character aspect ratio 1.38), while no arguments cause no operation

SLANT a   with default a = 0. degrees

DIRECTION d   with default d = 0. degrees

SET n   with n equal to a valid character set number
SET filename   to load an user defined character set
SET -1   to use the latest user defined set loaded

EJECT   in the form EJECT = EJECT ON, EJECT OFF

.cp3
COMMAND   followed by one or more lines of HP-GL commands, and terminated by a line containing END as first 3 characters

TEXT x y   followed by one or more lines of text, and terminated by a line containing END as first 3 characters. x y must be specified except in the case

TEXT CURSOR as TEXT but the text position is determined manually using the cursor keys.

AXIS with the following syntax possibilities

```
AXIS xy xorig yorig LIN length start step
AXIS xy xorig yorig LOG length other-info
```

where xy may be X or Y

where other-info may be in one of the forms

.cp3

```
startexp ndecades OLD
startexp ndecades DECADES
start    end      EXACT
```

OFFSET nchar where nchar is the number of character of which the axis label should be offset from the axis (and assumes suitable defaults, according to the orientation specified in **NUMBER**)

LABEL text

NUMBER with the following syntax

```
NUMBER format alignment orientation
```

where the accepted formats for LIN axes (or for LOG axes with less than two decades) are: Iw, Fw.d, Ew.d, Sw.d.e, Hw. The latter is valid only if HOUR is ON. The accepted formats for LOG axes are EXP and LOG, with default to EXP. the accepted alignment values are CENTRED, PROTECTED (the default), GRADUAL, LEFT and RIGHT.

the accepted orientation values are NORMAL (the default) and ROTATED

TICS height where height defaults to half character height

.cp50

.fo The Labeller Handbook (August 89)

Appendix 2

Appendix 2 : Text escape sequences

They are now supported by the TEXT and LABEL commands.

@S x,y @	set character size
@Pn@	select pen n
@D a @	select text direction angle
@A a @	select slant angle
@Cn@	select alternate character set
@C@	return to standard set
@*@	symbol mode (not yet supported)
@E@	start superscript
@e@	end superscript
@I@	start subscript
@i@	end subscript
@@	escape for a single "at" (@)

```
.CP50
.fo The Labeller Handbook (August 89)
Appendix 3
```

Appendix 3 : Error codes

Error codes in the range 400-700 are normal HP runtime errors (input/output) according to the usual codes. See the HP Quick Reference Guide, or the panel on the wall.

Plotter errors (lt 100) are HP-GL errors. See the plotter manual.

Errors in the range 900-1000 are LABLER errors as specified below:

901 HOUR not ON nor OFF. This is a syntax error in the HOUR command. Check it!

902 user defined character set not loaded yet. You have asked for set g-1, but have not yet loaded a character set

by filename !

903 (commissioning error : SET not ALTERNATE nor STANDARD)
You should NEVER get this error !

904 no x y given for TEXT position. You have specified a TEXT command without x or y (or both).

905 AXIS not X nor Y
906 no axis origin given
907 AXIS not LIN nor LOG
908 no axis lenght,start,step etc.
These are all syntax errors in the AXIS command. Check it !

909 too many decades or divisions. You are asking for more than 200 tics (or 20 decades in LOG OLD mode). Check your step and axis length. Unpredictable results on plotter.

910 AXIS not yet defined. This occurs if you call LABEL, NUMBER or TICS without having called AXIS at least once.

911 no format given for axis NUMBER. You have not specified a format in the NUMBER command.

912 illegal format specifier. Valid specifiers are I,F,E,S,H, EXP, LOG. Any other causes error.

913 H format invoked with HOUR OFF. You can't use H format without first calling HOUR ON.

914 invalid w.d format specifier. There is something wrong in your F or E format (probably a missing dot)

915 orientation is not NORMAL or ROTATED
916 alignment is not PROTECTED, CENTRED, LEFT, RIGHT or GRADUAL
917 log scale specifier is not OLD, DECADES or EXACT

.cp10
.fo The Labeller Handbook (August 89)
Appendix 4

Appendix 4 : Demo files

The included figures show the result of the use of The Labeller to annotate axes in a different number of ways, and namely :

The usage of different formats
The usage of escape sequences in axis labels

The usage of log scales with a large number of decades
The usage of log scales with less than 2 decades
The usage of the OFFSET command

The different number alignment for an x LIN axis

The same for a y LIN axis

The effect of rotation for a y LOG axis

More demo files are available on request to show the effect of alignment in different combinations of axis type and orientation, or to show the usage of log axis annotation in conjunction with PABLO.

.pa
.pa
.pa
.pa
.pa