

THERMOCODE SERIES 2

ESCAPE SEQUENCES

ISSUE 2.0

Designed and manufactured by;

**OPEN DATE EQUIPMENT LIMITED
Units 8 & 9 Puma Trade Park,
145 Morden Road,
Mitcham.
Surrey CR4 4DG
ENGLAND**

TEL:- 0044 (0)208 655-4999

FAX:- 0044 (0) 208 655 4990

1	General Operation of the Printer	1-5
1.1	How to use escape codes	1-5
1.1.1	Defining a label	1-5
1.1.2	Controlling the printer	1-7
1.2	Escape code listing	1-8
1.3	USB specification for Open Date Printers	1-11
1.4	ETHERNET or TCP/IP Networking for Open Date Printers	1-11
1.5	Opencode Custom Networking (RS 485 Variation)	1-12
1.6	PASSWORDS	1-13
1.6.1	Setting Passwords (Activation, Descriptions & Password codes)	1-14
1.6.2	Requesting Individual Passwords	1-14
2	Commands used to define a format	2-15
2.1	Loading a Font	2-15
2.2	Print Format Description (Used within a format)	2-15
2.3	Print Parameter Variables (Used within a format)	2-15
2.4	Text Lines	2-17
2.5	Line / Box Field	2-18
2.6	Barcode Field	2-18
2.6.1	2D Barcode (Datamatrix)	2-19
2.6.2	2D Barcode (Datamatrix) New Variable File	2-21
2.7	Graphics Field used in a Format (max 100 graphic fields per format)	2-22
2.8	Graphics Field loaded to a printer	2-22
2.9	Variable Insertion Fields	2-23
2.9.1	Variable Insertion field (Fixed text) type 0	2-24
2.9.2	Variable Insertion field (using limits) type 1	2-25
2.9.3	Variable Insertion field (24 Hour Time types) type 3	2-25
2.9.4	Variable Insertion fields (Counter Fields) type 4	2-26
2.9.5	Variable Insertion field (Day types) type 5	2-27
2.9.6	Variable Insertion field (Year types) type 6	2-27
2.9.7	Variable Insertion field (Month types) type 7	2-28
2.9.8	Variable Insertion field (12 Hour time types) type 8	2-29
2.9.9	Shift Codes (hours of shift) Type 9	2-29
2.9.10	Variable type W	2-30
2.10	Updating Existing Variable fields	2-31
2.11	Specifying Rollover times of Automatic Date Fields	2-32
2.12	Specifying Daylight Saving Dates	2-32
2.13	Requesting Daylight Saving	2-33
2.14	Special control variables used within Automatic Date Codes	2-33
2.15	Auto Print Command (from remote computer or similar)	2-34
2.16	Quantity Command	2-34
3	Commands to Control and Monitor the Printer	3-34
3.1	Delete all user specified Global Variable Text Fields	3-34
3.2	Delete Individual Global Variable Text Field	3-34
3.3	Delete all Global Graphics	3-34
3.4	Delete Individual Global Graphic	3-35
3.5	Change System Variables	3-35
3.6	Select Format	3-35
3.7	De-Select Format	3-35
3.8	Request Acknowledge	3-35
3.9	Request Format Data	3-36
3.10	Delete all Formats	3-36
3.11	Delete Individual Format	3-36
3.12	Request System Variables	3-37
3.13	Request Individual System Variable	3-37
3.14	Delete System Variables Changes	3-37
3.15	Delete Individual System Variable Changes	3-37
3.16	Request Global Text Variable Names	3-37
3.17	Request Global Text Variable Information	3-38
3.18	Request Printer Font Names	3-38

3.19	Request Printer Font Information	3-38
3.20	Delete all Printer Fonts	3-39
3.21	Delete Individual Font.....	3-39
3.22	Request Global Graphic Names.....	3-39
3.23	Request Global Graphic Information	3-40
3.24	Print Quantity (Used within a format only).....	3-40
3.25	Request Print Quantity	3-40
3.26	Request Total Prints	3-41
3.27	Printer output to communications line	3-41
3.28	Pause Printing	3-41
3.29	Resume Print.....	3-41
3.30	Request Memory	3-41
3.31	Request Format Name	3-42
3.32	Request Printer Status	3-42
3.33	Request Errors	3-42
3.34	Request a range of variables and parameters	3-42
3.35	Clear Errors	3-43
3.36	Initialise Part Printer	3-43
3.37	Initialise All Printer	3-43
3.38	End of Format Command	3-44
3.39	Insert and overwrite a local Variable Text field	3-44
3.40	Update and overwrite a local graphic field	3-44
3.41	Program Printer	3-44
3.42	Loading a file to the Printer.....	3-44
3.43	Request All Files.....	3-45
3.44	Request individual file information.....	3-45
3.45	Delete a file from file store.....	3-45
3.46	Selecting a new Burn File.....	3-46
3.47	Text Line output to Communications Port	3-46
3.48	Request Mechanical Variables	3-47
3.49	Request Individual Mechanical variable	3-47
3.50	Request all Print Parameters.....	3-48
3.51	Request an individual Print Parameter.....	3-48
3.52	Request format variables.....	3-48
4	How to use the printer in special situations.....	4-50
4.1	How to make a Thermocode 2 printer faster.....	4-50
4.1.1	The number of prints per minute.....	4-50
4.1.2	The time to download new information.....	4-50
4.1.3	The time to update the format information and to generate the initial image	4-50
4.1.4	The cycle time for a single print	4-51
4.2	Using the printer on a cross-web mounting	4-51
4.3	How to use multiple image planes to print multiple images	4-51

1 General Operation of the Printer.

The Thermocode 2 printer is controlled by a computer or other communicating device via a communications link. All the commands used start with an Escape character followed by command instructions and this manual describes how, by the use of escape codes to control and manage the printer.

The manual starts by describing the general use and layout of the escape codes and then gives details of all the available codes and control variables.

1.1 How to use escape codes.

There are two things that we can do with escape codes with the printer:

- Define the image for the label to be printed
- Control the operation of the printer

1.1.1 Defining a label.

The label to be printed is defined in a format. This consists of a number of lines of escape codes between a start line, containing the name of the format and an end line. This named format is saved within the file store of the printer and can be selected for printing either with an escape code or from the hand held terminal on the printer.

The escape code lines of a format will contain a range of information:-

- Print parameters that control the burn level, the speed, the head height etc. (Esc0P)
- Local text variable definitions that create text for things like counters, dates and time. (Esc0E)
- Local system variables to give precise printer control specific to a format. (Esc0X)
- Local pieces of graphic that stored within the format. (Esc0V)
- Line/Box drawing commands. (Esc0L)
- Text line drawing commands. (Esc0T)
- Barcode line definitions. (Esc0B)
- Graphic file use lines. (Esc0W)
- Split memory plane controls. (Esc0E...W)
- Label count control. (Esc0Q)
- Text output control lines. (Esc0Y)

A typical format would look like this:-

```

Esc0FTestLabel
Esc0PBURN1      0450
Esc0ECounter0 40001,1,1,1,9999
Esc0TArial  0010010016001Start text\0Counter\0End text
Esc0WGRAPH1  0126000001
Esc0B00010000300050040110123654321
Esc0L000000000400030000100010
Esc0Q000100
Esc0K
Esc0STestLabel

```

This defines a label:-

- Called TestLabel
- A print parameter BURN1 set to 450.
- A counter variable called Counter0.
- A line of text in Arial font with fixed start and end text and a counter.

- A graphic.
- A barcode.
- A Line/Box
- A count of 100
- The label is selected and the printer goes on-line

You can store any number of formats within the printer only limited by the size of the formats and the available space within the printer.

The use of text variables allows lines of text and barcode in a format to have any required level of complexity. Each variable defines a small bit of text, such as the current day number or a counter with 3 digits or a shift code. Once define these can be used in the format any number of times and combined with fixed bits of code.

Here is an example:-

```
Esc0Eday1  51
Esc0Emonth1 71
Esc0Eyear1 63
```

These define:-

- A variable day1 of type day with 2 numeric digits.
- A variable donth1 of type month with 2 digits.
- A variable year1 of type year with 4 digits.

These can then be built into a text line:-

```
Esc0Tarial  0100010010000Date=\0day1\0:\0month1\0:\0year1\0
```

Which will print the following:

```
Date=01:01:2006
```

In arial font, at position 100*100.

In the same label we could also have the following text lines:-

```
Esc0Tarial  0100020010000The current month=\0month1\0
Esc0Tarial  0100030010000The current year=\0year1\0
```

Which will print:-

```
The current month=01
```

At 100*200

And

```
The current year=2006
```

At 100*300

Most of the text variables automatically update when printing and give the format designer complete control over the text and barcodes on a label.

1.1.2 Controlling the printer

Many of the escape codes help to control and monitor the operation of the printer. Typical escape codes will:-

- Initialise the printer.
- Load fonts to the printer memory.
- Load global graphics to the printer memory.
- Stop and start printing.
- Request one or many of the system settings.
- Request information from the file store
- Return one or more system setting to their default.
- Delete one or more files from the file store.

These are all one line commands that can be sent at any time, for example:-

```
Esc0CINew  
Esc0AArial 0034578  
Esc0GVGRAPH2 02000200000000  
Esc0GECOUNTER0 40000  
Esc0GP
```

These define the following functions:-

- Full initialisation of the printer.
- Load the font Arial into the file store.
- Load the named graphic GRAPH2 to the file store.
- Create the global variable named Counter0 of type counter with 4 digits.
- Do an immediate print.

In addition to these direct commands there are a large number of system and mechanical variables within the printer that control every aspect of printer operation. Each of these variables is named and all have default values that allow the printer to be used in normal conditions but which can be modified to improve system performance or fit the printer to a particular application. For example:-

```
Esc0XSYSHEIGHT 2.0  
Esc0XPACCVERT 4000  
Esc0XSYSIPADDR 192.168.1.21
```

These control the following:-

- The head height is set to 2.0mm
- The vertical acceleration is defined set to 4000mm/s²
- The IP address is set to 192.168.1.21

Some functions require a number of system variables to control and you need to set all the relevant variables to get correct operation.

Use the escape code listings to help find the command you need. Look in the special sections that explain particular aspects of the printer operation. Look at the lists of system and mechanical variables for one that helps your application. (See separate manuals for full System & Mechanical listings)

1.2 Escape code listing

Esc Code	Qualifier	Command Global	Description	Section
A	None	Yes	Loading Fonts to Printer File Store.	2.01
B	None	No	Loading Barcodes within a Format.	12.6
C	See below	Yes	Command Code.	-
C	C	Yes	Pause Printing.	3.28
C	R	Yes	Resume Printing.	3.29
C	INEW	Yes	Deletes fonts, variables, graphics, and formats from filestore and memory, restores all default settings.	3.37
C	INIT	Yes	Deletes fonts, variables, graphics, and formats from filestore and memory	3.36
C	IPROG	Yes	Resets printer and allows new code to be downloaded.	3.41
D	See below	Yes	Delete Command	-
D	A	Yes	Delete all Fonts from filestore	2.52
D	A (font name)	Yes	Delete specific Font from filestore	3.21
D	E	Yes	Deletes all Global Variables from filestore & memory	3.01
D	E (global name)	Yes	Deletes specific Global Variable from filestore & memory	3.02
D	F	Yes	Deletes all Formats from filestore & memory	3.10
D	F (format name)	Yes	Deletes specific Format from filestore & memory	3.11
D	H (File name)	Yes	Deletes file from memory.	3.45
D	V	Yes	Deletes all Global Graphics from filestore & memory	3.03
D	V (graphic name)	Yes	Deletes specific Global Graphic from filestore & memory	3.04
D	X	Yes	Restores all system variables back to default .	3.14
D	X (variable name)	Yes	Restores a specific system variables to default.	3.15
D	Z	Yes	Clear all Errors	3.35
E	none	No	Variable Insertion Fields.	2.9
F	none	No	Format name. (file name)	2.2
G	See below	Yes	Global Graphics & Global Variable Text fields	-
G	E	Yes	Global Variable text fields loaded to filestore.	2.09
G	P	Yes	Automatic, Immediate Print Command	2.15

Esc Code	Qualifier	Command Global	Description	Section
G	Q	Yes	Remote Quantity Command	2.16
G	V	Yes	Global Graphic file loaded to filestore	2.08
H	none	Yes	Loading files to printer	3.42
I	none	Yes	Overwriting of existing local Variable Insertion Fields	3.39
K	none	No	Identifies end of Format file	3.38
L	none	No	Line & Box Field	2.05
O	none	No	Password Identification Number (Future use)	
P	none	No	Print Parameters.	2.03
Q	none	No	Quantity of prints Command within a format.	3.24
S	none	Yes	Deselect current format.	3.06
S	S(name)	Yes	Select format from filestore and load to Printer memory.	3.07
T	None	No	Text Fields	2.04
U	See below	Yes	Updates	
U	V	Yes	Updates a local graphic	3.40
U	E	Yes	Updates a local variable insertion field	2.10
V	None	No	Local graphics fields loaded within a format.	2.08
W	None	No	Graphic used in a format, including co-ordinates etc.	2.07
X	None	Yes/No	System Variables.	3.05
Y	None	No	Text Line output to Communications Port	3.47
Z	None	Yes	Request Printer type & Software Version.	3.08
Z	A	Yes	Request Font names from the filestore	3.18
Z	A (font name)	Yes	Requests Font file from filestore	3.19
Z	E	Yes	Requests Global Text field names from the filestore	3.16
Z	E (global name)	Yes	Requests Global Text field file from the filestore.	3.17
Z	F	Yes	Requests Format names from filestore	3.09
Z	F (format name)	Yes	Request Format file from the filestore	3.09
Z	H	Yes	Requests all files in Memory, including identification.	3.43
Z	H (file name & Identification)	Yes	Request actual file from memory.	3.43
Z	I	Yes	Request all Variables in Selected Format	3.52
Z	M	Yes	Request available memory of filestore.	3.30
Z	N	Yes	Request Name of Format being printed	3.31
Z	P	Yes	Requests all parameters of the Format being printed	3.50

Esc Code	Qualifier	Command Global	Description	Section
Z	P (name)	Yes	Requests individual print parameter	3.51
Z	Q	Yes	Request quantity that has been set and amount of prints that have been done. (current image only)	3.25
Z	R	No	Reserved Future Use	
Z	S	Yes	Requests Status line of printer	3.32
Z	T	Yes	Request Total Prints. (cycles in life of Printer)	3.26
Z	V	Yes	Request Global graphic names from filestore.	3.22
Z	V (graphic name)	Yes	Requests Global Graphic file from the filestore.	3.21
Z	X	Yes	Request System Variables settings from the memory.	3.12
Z	X (system name)	Yes	Requests a specific system Variable value.	3.13
Z	Y	Yes	Requests Mechanical Variables settings from the memory.	3.48
Z	Y (mechanical name)	Yes	Requests a Mechanical system Variable value.	3.49
Z	Z	Yes	Requests all Errors	3.33
Z	8	Yes	Reserved for future use	
Z	9	Yes	Requests a range of variables and parameters	3.34

General Notes

1. All lines prefixed by an Escape code (character 27) and a zero (character 48)
2. All lines terminated by carriage-return (character 13) and a line feed (character 10)
3. Null lines allowed. (carriage return and line feed with no other data).
4. The maximum characters on each line within a format is 150.
5. Names of Formats, can have up to 15 alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character.
6. The names of Fonts, Graphics & Variable Insertion Fields, can have up to 10 alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character.

Within a dedicated name there must be no space characters. The only valid characters used within a name:-

Alphabetical (upper or lower case) A -Z
 Numerical, 0 -9
 Underscore, _

1.3 USB specification for Open Date Printers.

The USB system international standard has had a number of revisions 1.0, 1.1 and 2.0.

At each revision, the maximum speed has been increased, but full compatibility has been maintained with the previous versions.

The current version is 2.0 and it defines 3 possible transfer speeds, (The speeds are defined as bits per second):

Low speed	=	1.5Mb/sec
Full speed	=	12Mb/sec
High speed	=	480Mb/sec

The Open Date printers meet the USB 2.0 specification for full speed operation at 12Mb/sec (1200000 bits per second).

We do not support operation at 480Mb/sec, because we cannot manage the high data rate in the printer.

Because of the way the specification works, our printers will plug into any USB enabled computer, regardless of the USB version that the computer runs (1.0, 1.1 or 2.0) as long as the correct driver is used.

Specifically, they will plug into a computer with a USB 2.0 specification port, and will run at a maximum 12Mb/sec. (depends on computer, windows version and printer operations)

1.4 ETHERNET or TCP/IP Networking for Open Date Printers

Printers that include a USB & Ethernet card need the following information, either pre-set or programmed to the printer:-

Port Number	=	9100 (Fixed in printer)
MAC Address	=	Pre-programmed on board by manufacturer.
IPADDRESS	=	Customers own identification. (Separate number each printer)
IPGATEWAY	=	Customers own identification. (Should match the network)
IPMASK	=	Customers own identification. (Should match the Network)

The system has been designed to work at 10 Megabit / second specification, but will connect to most 10/100 hubs to work with equipment running at 100 Megabit / second.

1.5 Opendate Custom Networking (RS 485 Variation)

The simple RS485 network that is supported allows a number of printers to be connected to a single computer. To identify which printer the computer is talking too at any time a network number is added to the command strings. The rest of this manual shows the commands without the added network number, but the following example shows how a network number and comma (,) is added to commands to address a particular printer:

Example of Standard format:-

```
Esc0FTest Label
Esc0PSPEED          0050
Esc0PBURN1          0450
Esc0PBURN2          0450
Esc0PPRESSURE       20
Esc0PROTATE         0
Esc0PHOMOFF         26.0
Esc0TARIALBD        0050012012000Open Date Equipment
Esc0Q000000
Esc0K
```

Example of Standard format being sent to Network Printer (number 12)

```
Esc12,0FTest Label
Esc12,0PSPEED       0050
Esc12,0PBURN1       0450
Esc12,0PBURN2       0450
Esc12,0PPRESSURE    20
Esc12,0PROTATE      0
Esc12,0PHOMOFF      26.0
Esc12,0TARIALBD    0050012012000Open Date Equipment
Esc12,0Q000000
Esc12,0K
```

1.6 PASSWORDS

Up to 9 password levels with individual descriptions and access codes may be programmed within the printer, the levels allow the user to specify which staff can access or program specific types of data. Normally the higher level of password will mean stricter security requirements. Utilising the levels correctly will allow a Manager to restrict his personnel to dedicated functions only. Passwords are numeric values between 1 and 8 characters long.

Default Password Settings

Password	Default Code	Default Name	Functions
Level 1 SYSPASS1	1111	OPERATOR	Selecting Formats. Editing Home Offset. Editing Quantity of Format. Can be disabled.
Level 2 SYSPASS2	2222	CHARGEHAND	Access to Level 1. Editing Text Fields (fixed & variable) Saving Format as another name. Can be disabled.
Level 3 SYSPASS3	3333	ENGINEER	Access to Levels 1 & 2. Editing Format Parameters. Editing Service Functions. Editing Format "X" & "Y" positions. Can be disabled.
Level 4 SYSPASS4	4444	MANAGER	Access to Levels 1, 2 & 3. Editing Barcode variables Editing Counter Variables Can be disabled.
Level 5 SYSPASS5	5555	SUPERVISOR	Access to Levels 1, 2, 3 & 4. Access to Supervisor Menu. Can be disabled.
Level 6 SYSPASS6	XXXX	OPENDATE SERVICE	Access to Levels 1, 2, 3, 4 & 5. Allows Service Engineer Access to customer machines
Level 7 SYSPASS8	XXXX	PROGRAMMING	Access to Levels 1, 2, 3, 4, 5 & 6. Access to Password setup menu's.
Level 8 SYSPASS8	XXXX	OPEN DATE TECHNICIAN	Allows access to Level 8 only. Programming of Printer Setup Menu, building & testing of machines only.
Level 9 SYSPASS9	XXXX	SECURITY	Allows access to all Levels.

The above security levels are used by the mini-terminal, to allow or restrict access to specific functions of the Printer. Editing of the Formats or changing the Printer settings, can then be restricted to prevent unauthorised access.

Once the required password has been entered correctly, access is then allowed, to only those specific functions of the program that are preceded with your security level number or less. Whichever level of password that is entered, will depend upon the type of access allowed. Entering a level 4 password will allow access to level 4

security codes and all functions of the lower security levels. (1 to 3)

1.6.1 Setting Passwords (Activation, Descriptions & Password codes)

Using these system variable codes, allows users to specify the name and pass-code of a specific System password. The passwords are system variables and only used by the mini-terminal display.

Esc0 {XSYSPASS1} {1} {,} {DDDDDDDD} {,} {NNNN..}

where:-

{XSYSPASS1}	=	System Password name.
{1}	=	enable/disable flag.
{,}	=	field separator. (coma)
{DDDDDD...}	=	password description. 18 alpha-numerical characters max.
{,}	=	field separator. (coma)
{NNNN..}	=	Password, 1 to 8 numeric characters.

The whole string (excluding the XSYSPASSN) can be up to 34 characters long allowing a name up to 23 characters long.

1.6.2 Requesting Individual Passwords

Esc0 {ZSYSPASS1}

where:- {ZSYSPASS1} = System Variable name for password for level 1.

printer responds with:-

Esc0 {XSYSPASS1} {1} {,} {DDDDDDDD} {,} {NNNN..}

where:-

{ZSYSPASS1}	=	Escape code to request password for level
{1}	=	enable/disable flag.
{,}	=	field separator. (coma)
{DDDDDD...}	=	Password description. 18 alpha-numerical characters max.
{,}	=	field separator. (coma)
{NNNN..}	=	Password, 1 to 8 numeric characters.

2 Commands used to define a format

2.1 Loading a Font

The Printer can store and utilise either TrueType (*.ttf) or Bit Map (*.sfp) fonts, for the printing of text characters, automatic dates and counter fields.

The format for loading a font into the printer is as follows:-

Esc 0A {name} {ss} {FFFFFF} {CRLF} Esc {font}

{name}	=	The name of the font: this field must be ten characters long with printable ASCII characters followed by trailing spaces.
{ss}	=	00 for scalable fonts; point size for bit mapped fonts.
{FFFFFF}	=	size in bytes of the font file. (000000) always six digits.
{CRLF}	=	Carriage Return, Line feed.
Esc	=	Escape.
{font}	=	font file contents, {FFFFFF} bytes.

2.2 Print Format Description (Used within a format)

The format for sending a new print format is as follows:-

Esc0F {name}

{name}	=	The name of the format: this field can be up to 15 characters long, with printable ASCII characters, followed by trailing spaces. (No spaces allowed within name)
--------	---	---

2.3 Print Parameter Variables (Used within a format)

The Print Parameters Variables must be entered before any other data in a format. The Parameters may be sent in any order.

These sequences will set the values of the print control parameters of the format. The Variable types must be named and entered correctly. See list below:-

Esc0P {name} {setting value}

Where:-

P	=	Escape code for print parameter variables.
{name}	=	Name of parameter to change. (Names of fields are pre-defined as list below)
{setting value}	=	Actual value required.

Note !.

Formats do not have to contain the print parameter variables, as the printer has default settings. You may however wish to include some of the parameter variables if required, these will override the printer default settings. If you need to edit values using the hand held terminal, then the parameters should be included in the format because terminal editing is only be allowed on parameters included in the format.

Standard Parameter Field Names

SPEED	=	Print speed in millimetres per second. (0050 = 50 mm/sec)
BURN1	=	Burn Value in microseconds, for intermittent A to B. and Continuous style printer. (0450 = 450 microseconds)
BURN2	=	Burn Value in microseconds, for intermittent B to A only (0450 = 450 microseconds)
PRESSURE	=	Pressure value in Newtons. (20 = 20 Newtons or 2Kg)
PEEL	=	Peel height in millimetres. (05.0 = 5 millimetres)
OFFSET	=	Distance in millimetres of print position from actual print signal. (050.0 = 50 millimetres) Continuous models only.
HOMOFF	=	Print position offset from Home Datum. See notes below. (05.0 = 5 millimeters)
ROTATE	=	Rotation of the Format, either 0 = 0° or 2 = 180°.

Home position offset. (HOMOFF)

This feature allows the user to specify within a format an offset in dots from the programmed Home position. This could save having to move the printer on the "Y" axis of a frame, enabling a new print position of the format to be achieved.

See the Standard printer Default setting sheets for valid ranges of the Parameter variables.

Example of Print Parameter Variables within a Format:-

```

EscOFFFormatName
EscOPSPEED      0050
EscPBURN1       0450
EscPBURN2       0450
EscPPRESSURE    20
EscPPEEL        07.0 (Not normally used)
EscPROTATE      0
EscPOFFSET      050.0 (Continuous Machines only)
EscPHOMOFF      26.0 (Intermittent machines only)
EscOTARIALBD    00500120120000pen Date Equipment
EscOQ000000
EscOK

```


2.4 Text Lines

(Maximum 120 Text Lines per format, 150 text fields per format, 100 characters maximum per text field. Also see file store variables that can include large amounts of data)

Esc0T {ffffff} {xxxx} {yyyy} {ss} {o} {r} {P} (\1) {ppp} (\1) {a..a} (\0) {variable_name} (\0) {b....b}

NOTE (\0) means binary all zeros or the NULL character.

{ffffff}	=	The name of the font: this field must be ten characters long with printable ASCII characters followed by trailing spaces.
{xxxx}	=	horizontal position, in dots.
{yyyy}	=	vertical position, in dots.
{ss}	=	scaling factor. (point size nominal)
{o}	=	orientation:- (Clockwise rotation) 0 = 0°, 1 = 90°, 2 = 180°, 3 = 270°.
{r}	=	Reverse outline option (not implemented) set at 0.
{P}	=	Proportional flag, 0 = fully proportional, 1 = monospace variable. (not implemented)
(\1)	=	Binary Character 1 (start of percentage of Font width)
{ppp}	=	Percentage width of Font, must be 3 digits. (may be left out if not required)
(\1)	=	Binary Character 1 (end of percentage of Font width)
{a...a}	=	Fixed ASCII text , may be left out if not required.
{variable_name}	=	Name of pre-defined variable. Must have a \0 before and after variable_name.
{b...b}	=	Fixed ASCII text, may be left out if not required.

Note!

When using multiple variable insertion fields within a format, you must still have a \0 before and after each variable_name. See following examples.

Example 1:-

Esc0TARIALBD 0050005012000**5212**\0**variable1**\0

Fixed text (5212) before a variable named field (variable1)

Example 2:-

Esc0TARIALBD 0050005012000**5212**\0**variable1**\0\0**variable2**\0

Fixed text (5212) before a two variable named fields (variable1 & variable2)

Example 3:-

Esc0TARIALBD 0050005012000\Øvariable1\Ø5212

Fixed text (5212) after a variable named field (variable1)

Example 4:-

Esc0TARIALBD 0050005012000\Øvariable1\Ø5212\Øvariable2\Ø

Fixed text (5212) between two variable named fields (variable1 & variable2)

2.5 Line / Box Field

(No maximum limits on number of fields)

Esc0L {xxxx} {yyyy} {www} {hhh} {vvv} {rrr}

- {xxxx} = Horizontal position, in dots (X axis).
- {yyyy} = Vertical position, in dots (Y axis).
- {www} = Horizontal width, in dots (X axis).
- {hhh} = Vertical height, in dots (Y axis).
- {vvv} = Width thickness, in dots (X axis).
- {rrr} = Height thickness, in dots (Y axis)

Together, these parameters may describe vertical lines, horizontal lines, filled blocks or boxes, with sides of different thickness.

Lines or filled blocks are described by supplying width and height parameters as required and setting both thickness parameters (vvv) + (rrr) to all zero's. Boxes are described by supplying all parameters as required, where the thickness parameters set the thickness of the vertical and horizontal sides of the box respectively.

2.6 Barcode Field

(maximum 20 barcode fields per format)

Esc0B {ss} {xxxx} {yyyy} {hhh} {o} {b} {r} {h} {c} {s} {a..a} (\Ø) {variable_name} (\Ø) {b...b}

NOTE (\Ø) means binary all zeros or the NULL character.

{ss}	=	Style	Symbology	Style	=	Symbology
		00	= EAN 8	09	=	CODABAR
		01	= EAN 13	10	=	MSI
		02	= EAN 128	11	=	CODE 93
		03	= UPC-A	12	=	EXTENDED CODE 93
		04	= UPC-E	13	=	UCC - 128
		05	= CODE 39	14	=	HIBC
		06	= CODE 128	15	=	UPCE (6, SYSTEM 0)
		07	= ITF CODES	16	=	UPCE (6, SYSTEM 1)
		08	= EXTENDED CODE 39	20	=	DATAMATRIX
		{xxxx}	= horizontal position			
		{yyyy}	= vertical position			
		{hhh}	= height (not including human readable characters)			
		{o}	= orientation:- (Clockwise rotation)			

		0 = 0°
		1 = 90°
		2 = 180°
		3 = 270°
{b}	=	Narrow Bar width in dots. (between 1 to 9)
{r}	=	styles (element thickness & ratios):-
		0 = 3:1 ratio
		1 = 2.5:1 ratio
		2 = 2:1 ratio
{h}	=	Human readable option, (1 = ON, 0 = OFF).
{c}	=	Checksum Digit Automatic, (1 = ON, 0 = OFF).
{s}	=	Speed flag, 0 = normal, 1 = boundary move (future use)
{a...a}	=	Fixed ASCII text, may be left out if not required.
{variable_name}	=	Name of pre-defined variable. Must have a \Ø before and after variable_name.
{b...b}	=	Fixed ASCII text, may be left out if not required.

Note!

When using multiple variable insertion fields within a format, you must still have a \Ø before and after each variable_name. Please see following examples.

Example 1:-

```
Esc0B01005000200150040115212\Øvariable1\Ø
```

Fixed text (5212) before a variable named field(variable1)

Example 2:-

```
Esc0B01005000200150040115212\Øvariable1\Ø\Øvariable2\Ø
```

Fixed text (5212) before a two variable named fields (variable1 & variable2)

Example 3:-

```
Esc0B0100500020015004011\Øvariable1\Ø5212
```

Fixed text (5212) after a variable named field(variable1)

Example 4:-

```
Esc0B0100500020015004011\Øvariable1\Ø5212\Øvariable2\Ø
```

Fixed text (5212) between two variable named fields (variable1 & variable2)

Note! For EAN type barcodes, the style parameter will be ignored.

2.6.1 2D Barcode (Datamatrix)

```
Esc 0B {ss} {xxxx} {yyyy} {qq} {i} {o} {m} {a} {cc} {rr} {bb} {dd} {a..a} (\Ø) {variable_name} (\Ø) {b...b}
```

NOTE (\Ø) means binary all zeros or the NULL character.

{ss}	=	Style	Symbology
------	---	-------	-----------

	20	=	Data Matrix
{xxxx}	=	horizontal position	
{yyyy}	=	vertical position	
{qq}	=	ECC Quality Printer (represents)	ECC Code Standard
	00		0
	01		10
	02		40
	03		50
	04		60
	05		70
	06		80
	07		90
	08		100
	09		110
	10		120
	11		130
	12		140
	13		200 EAN UCC Standard.

{i}	=	Character set	[1] to [6]
	1		1..9' ' (500c)
	2		A..Z ' ' (500c)
	3		A..Z 1..9' ' , . - / (500c)
	4		A..Z 1..9 (500c)
	5		ASCII 128 Keyboard (500c)
	6		ISO 8 – Bit International (500c)

{o}	=	orientation:- (Clockwise rotation)
	0	= 0°
	1	= 90°
	2	= 180°
	3	= 270°

{m}	=	Mirror
	0	= Normal
	1	= Mirror

{a}	=	Automatic size
	1	= Automatic size; SizeCol and SizeRow are disabled

{cc} = SizeCol [8 – 99]

{rr} = SizeRow [8 – 99]

{bb} = Border size [1 – 15]

{dd} = Dots multiplier [1 – 99]

Fixed and variable data fields:-

{a...a} = Fixed ASCII text, may be left out if not required.

{variable_name} = Name of pre-defined variable. Must have a \Ø before and after variable_name.

{b...b} = Fixed ASCII text, may be left out if not required.

Example:-

```
Esc 0B {ss} {xxxx} {yyyy} {qq} {i} {o} {m} {a} {cc} {rr} {bb} {dd} {a..a} (\Ø) {variable_name} (\Ø) {b...b}
20 0165 0009 13 6 0 0 1 00 00 01 04 QWER \ØBAR\Ø 12q34
```

2.6.2 2D Barcode (Datamatrix) New Variable File

NOTE: The file type variable described below can also be used for text lines.

If a variable field is more than 40 bytes (characters) a normal variable won't be able to cope. We therefore allow a special kind of variable that gets its contents from a file within the file store. There are two parts to these variables:

Storing the data in the file store:

Use the 'H' command that allows a file to be written directly to the file store. The following example shows all the QWERTY characters written into a file called fileTHT00001. The number '9' is the type of the file, (in this case type txt) and should always be 9.

```
Esc0HfileTHT00001 9000034
EscQWERTYUASDFG123456789LKJHGFDSAZXCB
```

Using the file data on a format:

To use the file data you must first define a variable of type 'A'. In this example the variable is called 'Bar' is of type 'A' and its value is the name of the file in the file store "fileTHT00001". You can then use the variable in a barcode definition.

```
Esc0FDocument1
(Speed, Burn(s), Pressure, Homeoff and Rotate are removed for clarity. )
Esc0EBar AfileTHT00001
Esc0B20016500091300100000104A123456\ØBar\ØAAABBB123456
Esc0Q000000
Esc0K
```

Note!

When using multiple variable insertion fields within a format, you must still have a \Ø before and after each variable name. Please see following examples.

Example 1: -

```
0B20016500091300100000104A212\Øvariable1\Ø
Fixed text (A212) before a variable named field (variable1)
```

Example 2: -

```
0B20016500091300100000104A212\Øvariable1\Ø\Øvariable2\Ø
Fixed text (A212) before a two variable named fields (variable1 & variable2)
```

Example 3: -

```
0B20016500091300100000104\Øvariable1\ØA212
Fixed text (A212) after a variable named field(variable1)
```

Example 4: -

```
0B20016500091300100000104\Øvariable1\ØA212\Øvariable2\Ø
Fixed text (A212) between two variable named fields (variable1 & variable2)
```

Recommendations: -

Type “200” is the EAN™ UCC™ standard.	“{qq} 13”
Character set 5 or 6	“{i} 5 or 6”
Use Automatic sizing	“{a} 1”

2.7 Graphics Field used in a Format (max 100 graphic fields per format)

Graphics data is saved in the file store either globally in named files or locally as part of a format. Once they have been saved they can be used within the format. The ‘W’ command places a graphic onto the label specifying its position, orientation and scale factor. A single graphic can be used many times in a format.

Esc0W {nnnnnnnnn} {xxxx} {yyyy} {s} {o}

{nnnnnnnnn} = Name of Global Graphic file stored within Printer filestore.

{xxxx} = Horizontal position, in dots.

{yyyy} = Vertical position, in dots.

{s} = Scaling factor:-

0 = 1:1

1 = 2:1

2 = 4:1

{o} = orientation:- (Clockwise rotation)

0 = 0°

1 = 90°

2 = 180°

3 = 270°

Where a scaling factor of N:1 applies, each dot and each row are copied N times into the image memory, so that the final image size is N* width x N* height.

2.8 Graphics Field loaded to a printer.

There are two commands for loading graphics into the printer:

Esc0GV {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {d...d}

Esc0V {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {d...d}

They are almost identical, but, the first loads a global graphic and the second loads a local graphic. Apart from the extra ‘G’ they are the same.

A global graphic is loaded into the printer before the formats are loaded and can be used by any of the formats in the file store. This is ideal for company logo’s etc. You can load as many global graphics as the file store will take.

A local graphic is loaded as part of a format and can only be used from within that particular format. This is good for something specific to a format, such as a picture of a particular product.

{nnnnnnnnnn}	=	Name of Graphic image, 10 alphanumerical characters.
{wwwwww}	=	width (bits)
{hhhhh}	=	height (bits)
{S}	=	Style of Graphic image. 0 = *. BMP files 1 = *. PCX files 2 = *. BMP files (fast special applications) 3 = *. LZ Compressed files. (special applications)
{fffff}	=	Size in bytes of the graphic. Note that this is not just width times height, unless the graphic is rectangular.
{CR}	=	Carriage Return.
Esc	=	Esc.
{d...d}	=	Content (binary data).

No terminator character is used, as the data is in binary format.

The BMP format is a simplified form of the Microsoft BMP format as follows:

The data is single bit per pixel monochrome.

The binary data is arranged as horizontal lines of the image, each line encoded separately.

Each horizontal line of the bit-mapped image is encoded as a word count of the number of data bytes in the line, followed by the binary data starting with the left most pixel. The word count is two bytes MSB, LSB.

Each line is output with subsequent lines running top to bottom until the full height of the image is completed.

The use of the byte count in each horizontal line allows lines with no data at the end to be shortened, ie the last bytes of data do not need to be stored.

The PCX format is the standard PCX format as supported by many graphics programs.

The LZ format uses a special form of compression to reduce the download size of a graphic file.

!!!Thierry to add detail!!!

2.9 Variable Insertion Fields

(Maximum 100 Variable insertion fields Global + Format, 60 characters per field generated, 140 characters line length)

Named variable insertion fields create text or numbers that become part of text or barcode lines. Variables are of different types and each type creates its own sort of information. Many types create data that changes with time, such as counters or dates. By combining a range of variables and fixed text it is possible to produce any kind of text or barcode line.

Variables can be Global, defined outside a format or Local, defined inside a format. Global variables can be used by any format in the printer. Local variables can only be used within the format in which they are defined.

Each variable can be used as many times as required (or not at all). This would allow a barcode and a text line to contain the same variable information.

Variables can be edited using the terminal.

Some of the later parameters to a variable may be optional. This is indicated in the descriptions.

Local variables use the form:-

Esc0EName

and are part of a format.

Global variables use the form:-

EscGEName

and are define outside a format.

This section shows the local format, but the global format is also valid.

For variable Insertion Field Names see the general notes, section 6 for naming conventions.

Types of Variable Insertion fields

Variable Insertion Field types	Description	Variable Insertion Field types	Description
0	Fixed Text	6	Year types
1	Text using Limits	7	Month types
2	Reserved Future use.	8	12 hour times
3	24 Hour times	9	Shift Codes
4	Counter Fields	A	Specialist Text Lines
5	Day types	W	Image Memory Partitions

2.9.1 Variable Insertion field (Fixed text) type 0

Esc0E {nnnnnnnnnn} {0} {tttttttt...}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{0} = 0 = unchecked variable.

{tttttttt} = Text of Variable Insertion Field. (60 characters max)

This type provides simple text that can be easily edited or changed in the printer.

2.9.2 Variable Insertion field (using limits) type 1

Esc0E {nnnnnnnnnn} {1} {tttttttt...} {,} {LLLLL} {,} {HHHHH}

{nnnnnnnnnn}	=	Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character.
{1}	=	Variable set up with high and low limits. (numerical text only)
{tttttttt}	=	Text of variable insertion field
{,}	=	Separator character (coma)
{LLLLL}	=	Lower limit of variable text.
{,}	=	Separator character (coma)
{HHHHH}	=	Higher limit of variable field.

The text must be number between the lower limit and the upper limit or an error will be generated. The number range is -2147483646 to 2147483647.

2.9.3 Variable Insertion field (24 Hour Time types) type 3

Esc0E {nnnnnnnnnn} {3} {T} {oooo}

{nnnnnnnnnn}	=	Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)
{3}	=	Variable Time indication.
		[Optional parameters from here]
{T}	=	Type of Time insertion Field (see below listing)
{oooo}	=	Amount of minutes to offset. (valid range -1439 to +1439)

Types of 24 hour Time Insertion fields:-

1 = hh:mm:ss (24 hour)	2 = hh:mm (24 hour)
3 = hhmm (24 hour)	4 = hh (24 hour)
5 = mm	6 = ss

Each field is always 2 digits with '0' padding on the left.

The times and dates are all taken from the real time clock which should be set correctly. The offset field allows the time to be shifted backwards and forwards by up to 1 day. Each variable field has its own offset and will not affect other time fields.

All 24 hour time types update automatically.

2.9.4 Variable Insertion fields (Counter Fields) type 4

This function is a special Variable Insertion counter field that allows the user to specify:-

Start Number
Justification
Increment
Pallet Count
Counter rollover or Batch Quantity

(Maximum 20 fields per format design)

Esc0E {nnnnnnnnnn} {4} {sssssss.....} {,} {J} {,} {iiiiiii...} {,} {pppp..} {,} {rrrrrrr...}

where:-

Esc E	=	Escape Code to tell printer Variable Insertion Field.
{nnnnnnnnnn}	=	Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)
{4}	=	Specifies Counter type Variable Insertion field.
{sssssss..}	=	Counter start Number, leading zero's specify the maximum counter capacity and printability. Leading spaces before the start number indicate the number will be printed with leading spaces. (see justification parameter)

[Optional parameters from here]

{,}	=	Separator character (coma)
{J}	=	Justification Character, (0 = left), (1 = right)
{,}	=	Separator character (coma)
{iiiiiii...}	=	Increment amount, specify the amount each print has to advance.
{,}	=	Separator character (coma)
{pppp..}	=	Pallet Count/Repeating sequence Number, specifies the number of the same prints required before incrementing to the next count.
{,}	=	Separator character (coma)
{rrrrrr}	=	Roll - over number, specifies the last printed number before the counter resets to the original start number.

Specifying Stop Number

To specify the Number of Prints required, or stop number simply "edit" the Quantity line of the Format.

2.9.5 Variable Insertion field (Day types) type 5

Esc0E {nnnnnnnnnn} {5} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{5} = Variable Day indication.

{T} = Type of Time insertion Field (see below listing)

[Optional parameters from here]

{oooo} = Number of Days to offset. (Valid range -9,999 to 9.999)

{,} = Separator character. (coma)

{rrrr} = Offset from Midnight in minutes to rollover date. (valid range -1439 to +1439)

{,} = Separator character. (coma)

{mmmm} = Number of Months to offset, from printer date. (Valid range -9,999 to 9.999)

All Day fields update automatically. You can't have Month and Day offset at the same time, one at least must be zero.

Types of Day Insertion fields:-

1 = dd (day of month, 2 numerical digits).

2 = d (day of week as a numerical number, uses **SYSDAY1** Global variable)

3 = ddd (day of week , 3 alphabetical, uses **SYSDAY3** Global variable)

4 = ddddd... (day of week, alphabetical spelling, uses **SYSDAY** Global variable)

5 = JJJ (Julian Day of Year, Day 366 = 29 Feb)

6 = JJJ (Julian Day of Year, Day 366 = 31 December)

2.9.6 Variable Insertion field (Year types) type 6

Esc0E {nnnnnnnnnn} {6} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{6} = Variable Year indication.

{T} = Type of Time insertion Field (see below listing)

[Optional parameters from here]

{oooo} = Number of Days to offset. (Valid range -9,999 to 9.999)

{,}	=	Separator character. (coma)
{rrrr}	=	Offset from Midnight in minutes to rollover date. (valid range -1439 to +1439)
{,}	=	Separator character. (coma)
{mmmm}	=	Number of Months to offset, from printer date. (Valid range -9,999 to 9.999)

All Year fields update automatically. You can't have Month and Day offset at the same time, one at least must be zero.

Types of Year Insertion fields:-

1 = Y	(Year, 1 numerical digits. 1998 = 8)
2 = YY	(Year, 2 numerical digits. 1998 = 98)
3 = YYYY	(Year, 4 numerical digits. 1998 = 1998)
4 = WW	(Week number of year, 2 numerical digits)

2.9.7 Variable Insertion field (Month types) type 7

Esc0E {nnnnnnnnnn} {7} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnnn}	=	Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)
{7}	=	Variable Month indication.
{T}	=	Type of Time insertion Field (see below listing)

[Optional parameters from here]

{oooo}	=	Number of Days to offset. (Valid range -9,999 to 9.999)
{,}	=	Separator character. (coma)
{rrrr}	=	Offset from Midnight in minutes to rollover date. (valid range -1439 to +1439)
{,}	=	Separator character. (coma)
{mmmm}	=	Number of Months to offset, from printer date. (Valid range -9,999 to 9.999)

All Month fields update automatically. You can't have Month and Day offset at the same time, one at least must be zero.

Types of Month Insertion fields:-

1 = MM	(Month, 2 numerical digits. JAN = 01)
2 = MMM	(Month, 3 Alphabetical characters. JAN = JAN) uses SYSMON3 Global variable.
3 = MMMMM...	(Month, Alphabetical characters APRIL = APRIL) uses SYSMONTH Global variable.
4 = M	(Month single alphabetical character, JAN = A) uses SYSMON1 Global variable.

2.9.8 Variable Insertion field (12 Hour time types) type 8

Esc0E {nnnnnnnnnn} {8} {T} {offset} {,} {am} {,} {pm}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{8} = Variable Time indication.

{T} = Type of Time insertion Field (see below listing)

[Optional parameters from here]

{offset} = Amount of minutes to offset. (valid range -1439 to +1439)

{,} = Separator character, coma.

{am} = Optional first 12 hour indication, change as required.

{,} = Separator character, coma.

{pm} = Optional second 12 hour indication, change as required.

Types of 12 hour Time Insertion fields:-

1 = hh:mm:ss (12 hour) 2 = hh:mm (12 hour)

3 = hhmm (12 hour) 4 = hh (12 hour)

5 = mm 6 = ss

Each field is always 2 digits with '0' padding on the left. The optional am and pm fields are added to the end of the numeric output.

The times and dates are all taken from the real time clock which should be set correctly. The offset field allows the time to be shifted backwards and forwards by up to 1 day. Each variable field has its own offset and will not affect other time fields.

All 12 hour time types update automatically.

2.9.9 Shift Codes (hours of shift) Type 9

(Max 24 shift periods in one day, max 4 shift variables)

Inserts the "SYSSHIFT" text into the relevant fields. **See section ????**

Esc0E {nnnnnnnnnn} {9} {TTT1} {,} {TTT2} {,} {TTT3} {,} {TTT4} {,} {TTT5} {,} {TTT6}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{TTT1} = Start time of 1st Shift period. (24 hour clock)

{,} = Separator character (coma)

{TTT2}	=	Start time of 2 nd Shift period. (24 hour clock)
		[Optional parameters from here]
{,}	=	Separator character (coma)
{TTT3}	=	Start time of 3 rd Shift period. (24 hour clock)
{,}	=	Separator character (coma)
{TTT4}	=	Start time of 4 th Shift period. (24 hour clock)
{,}	=	Separator character (coma)
{TTT5}	=	Start time of 5 th Shift period. (24 hour clock)
{,}	=	Separator character (coma)
{TTT5}	=	Start time of 6 th Shift period. (24 hour clock)

If only 3 shifts are required per day, then only specify that values {TTT1} through to {TTT3}.

2.9.10 Variable type W

This variable allows multiple separate images to be created within a single format. Using SYSILEN the image memory that is normally used as a single image is split into a number of smaller sections. Each section can then be written with a different image all from the same format. When printing, a different image can be selected instantaneously for each print.

This currently only works on continuous printers.

Make a Multi-Image printer by splitting up the image memory into several sections and then store each image into a section.

This variable allows multi images in a format to be loaded direct to the image sections, which have previously been created by using the variable SYSILEN, and seeing the result in SYSNUM.

Esc0E {nnnnnnnnnn} {W} {0} {1}

Where:-

{nnnnnnnnnn}	=	Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name). This field is a dummy and any name can be used.
{W}	=	Image Memory Store type variable.
{0}	=	Image plane. (0 = Write to both image planes) (1 = Write to image plane 1) (2 = Write to image plane 2)
{1}	=	Number of the image section being stored.

Note! The SYSNUM retrieves the number of images allowed in a single image plane.

If you require any dates or counters you must use both image planes. (Allowing for updates etc)

If the images are just graphics or text that does not need update, you can specify individual planes for storing the images. This allows you to use both image planes individually, allowing double the SYSNUM number of images.

To use this variable, place it before one or more image generating lines in the format. All those images will appear in that section. To put data into another section have another variable, ie:

```
Esc0Edummy1 W11
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
Esc0Edummy1 W12
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
Esc0W {nnnnnnnnn} {xxxx} {yyyy} {s} {o}
Esc0Edummy1 W21
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
```

This puts:

```
barcode and text into plane1/section1
barcode and text and graphics into plane1/section2
barcode and text into plane2/section1
```

2.10 Updating Existing Variable fields

(within currently selected Format)

It is possible to update an existing variable field within a format already loaded to the Printer (local variable). This allows a much higher speed updating of a format, as you do not have to send again any fixed text fields, lines or graphics. Updated fields should be made the same length and type as the original loaded within the format, but they can be shorter. (longer fields may overlap other printing fields)

Example1 (a typical format including a variable Lot No.)

```
Esc0FTEST2
Esc0PSPEED 0100
Esc0PBURN1 0420
Esc0PBURN2 0420
Esc0PPRESSURE 0018
Esc0ELot Number 0123456
Esc0B010050001015004011v789034\ØLot Number\Ø
Esc0TSwiss 005002501200vLot:-\ØLot Number\Ø
Esc0Q000000
Esc0K
```

To update the Lot Number only you send:-

```
Esc0UELot Number 0334477
Esc0Q000000
```

This would then upgrade the Lot number variable to 334477 instead of 123456

This example shows the quantity being sent as well. This is an option if you need to set a new batch number quantity.

All types of variable insertion fields can be updated this way.

2.11 Specifying Rollover times of Automatic Date Fields

Normally automatic date fields always rollover (increment to next date) at midnight, it is possible to change the actual time the specified format that is loaded to the printer will automatically update using offsets.

Rollover times are always set from midnight, either plus or minus in minutes. (-1439 to + 1439) range.

This does not change the actual time of the printer.

Example, you wants to print a normal day field type 5 with no day offsets but rollover the day at 10pm (2200) hours instead of midnight (2400).

```
Esc0Edate      51000,-120
Esc0TARIALBD  00500050120000\0Date\0
```

Where:-	Date	=	Name of Variable
	2	=	Day type Variable
	1	=	Style of Day type
	000	=	Offset in days (not used)
	,	=	Field separator (coma)
	-120	=	Offset time in minutes from midnight.

2.12 Specifying Daylight Saving Dates

You can program in the exact dates the Printer needs to adjust itself to allow for daylight savings. The Customer must specify both dates at which the time clock should adjust automatically, and whether the first date specified has the clock going forward or backwards by 1 hour.

It is assumed that any clock going forward will happen at 1am (0100 hours).

It is assumed that any clock going backward will happen at 2am (0200 hours).

The settings are made in the form of a variable as shown below. This can be set at any time but will only be relevant for the particular year that you set the date at.

Esc0 {XSYSDSAVE} {ddd1} {,} {ddd2} {,} {h}

{XSYSDSAVE}	=	System Variable name.
{ddd1}	=	The first date when a change is to take place. (4 digits MMDD)
{,}	=	field separator. (coma)
{ddd2}	=	The second date when a change is to take place. (4 digits MMDD)
{,}	=	field separator. (coma)
{h}	=	Hour advance/retard flag. 0 = Forwards 1 hour at date, 1 = Backwards 1 hour at first date.

Note ! You must reset the time in the Printer after sending the daysaving Variable.

2.13 Requesting Daylight Saving

Esc0 {XSYSDSAVE}

The printer responds with:-

- {XSYSDSAVE}** = System Variable name.
- {0416}** = The first date when a change is to take place. (4 digits MMDD)
- {,}** = field separator. (coma)
- {1121}** = The second date when a change is to take place. (4 digits MMDD)
- {,}** = field separator. (coma)
- {h}** = Hour advance/retard flag. 0 = Forwards 1 hour, 1 = Backwards 1 hour.

2.14 Special control variables used within Automatic Date Codes.

Within the printer are some dedicated system variables that control the way that the date variable fields work. When a date variable is used the text created comes from these system variables. By changing the system variable you can change the text output. The system variables involved are:-

- Esc0SYSDAY** =Monday etc (Alphabetical characters) see example
- Esc0SYSDAY3** =MON (3 Alphanumeric characters) see example.
- Esc0SYSDAY1** =1 (1 Alphanumeric Character) see example.
- Esc0SYSMONTH** =JANUARY (Alphabetical characters) see examples.
- Esc0SYSMON3** =JAN (3 Alphabetical characters) see examples.
- Esc0SYSMON1** =1 (1 Alphanumeric Character) see example.
- Esc0SYSSHIFT** =S1 (Max 5 Characters) see example

The above variables default to the English language. If customisation is required, please see the examples below that show how each type is created and sent to the Printer.

Change the required text in the text string. Do not forget to separate each field with a coma.

Escape Code	Global Text Name	Used with variable type	Text string information
2.14.1.1.	SYSDAY	5	Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday
Esc0X	SYSDAY3	5	MON,TUE,WED,THU,FRI,SAT,SUN
Esc0X	SYSDAY1	5	1,2,3,4,5,6,7
Esc0X	SYSMONTH	7	January,February,March,April,May,June,July,August Etc.....
Esc0X	SYSMON3	7	JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
Esc0X	SYSMON1	7	A,B,C,D,E,F,G,H,J,K,L,M
Esc0X	SYSSHIFT	9	S1,S2,S3,S4,S5,S6

Note !

The Printer always assumes that Monday is the start of the week, and January is the first month of any year.

The above fields cannot be deleted from the Printer, only overwritten. An initialisation (Esc0INIT) of the Printer will change the variables back to the default English settings.

2.15 Auto Print Command (from remote computer or similar)

Esc0GP

This command allows a user to command a print signal from a computer or similar.

2.16 Quantity Command

Esc0GQ {NNNNNN}

Where:-

GQ = Escape codes for sending new quantity command.

{NNNNNN} = Actual new quantity value. (up to six digits)

Allows the operator to reset the quantity counter to zero and to specify a new quantity. The printer will then print up to that number of labels and then stop.

3 Commands to Control and Monitor the Printer.

3.1 Delete all user specified Global Variable Text Fields

Esc0DE

This command deletes all Global Variable Text fields from the printer File store. The current print format will be de-selected, and the printer will be set "Off Line".

3.2 Delete Individual Global Variable Text Field

Esc0DE {name}

Where:- {name} = Actual name of Global Text field in printer, this name must be correct and is case sensitive.

This command deletes a specific Global Variable Text fields from the printer File store. The current print format will be de-selected, and the printer will be "Off Line".

3.3 Delete all Global Graphics

Esc0DV

This command deletes all Global Graphics from the printer File store. The current print format will be de-selected, and the printer will go to "Off Line".

3.4 Delete Individual Global Graphic

This command deletes a specific graphic from printer memory.

Esc0D V {name}

Where:-

{name} = Actual name of graphic in printer, this name must be correct and is case sensitive.

This command deletes a specific Global Graphic from the printer File store. The current print format will be de-selected, and the printer will go to "Off Line".

3.5 Change System Variables

Esc0X {name} {data}

{name} = The actual specific system variable you wish to change.

{data} = Content of Data field, relevant to field description.

Please see separate manuals for Mechanical Variables & System Variables, which have a full description of the function and ranges available.

Note! At anytime you may read back an individual System Variable with the escape sequence:-

Esc0ZX {Name} Where {name} is the Actual system variable code name as shown above, this will then return the actual value of the specified field.

System variables can be changed as **globals** or as **locals**. If they are sent outside a format description then they are global and the change will affect all the formats in the printer. If they are sent as part of a format (between the Esc0Fname and the Esc0K) then they are local. A local system variable will override a global system variable when a format is selected for printing. Different formats can have different local system variable settings.

3.6 Select Format

This command permits the selection of a format from the file store, which is then loaded to the printer memory ready for printing.

Esc0S {Name}

where:- {name} = Actual name of Format in file store, this name must be correct and is case sensitive.

3.7 De-Select Format

This command de-selects the currently selected format being printed, and parks the printhead.

Esc0S

3.8 Request Acknowledge

Esc0Z

Printer replies with:- Esc Z (mmmm,nnnn)
OK

Where (mmmm,nnnn) = Model no of printer & Software Version number.

3.9 Request Format Data

Esc0ZF

Printer replies with:- All format names stored within the Printer.

```
F      {name 1}
F      {name 2}
F      {name 3}
Etc.....
OK
```

You can also upload an actual format file contents by specifying:-

Esc0ZF {name 1}

Printer replies with:-

```
Esc0FName
Esc0PSPEED      0050
Esc0PBURN1      0450
Esc0PBURN2      0450
Esc0PPRESSURE   17
Esc0PROTATE     0
Esc0OFFSET      050.0
Esc0TARIALBD    00500120120000pen Date Equipment
Esc0Q000000
Esc0K
OK
```

Note! If the format name is not recognised the printer will just return OK and the error "List file unknown" will be displayed on the terminal.

3.10 Delete all Formats

This command deletes all formats from file store & both printer memories.

Esc0DF

This command deletes all fonts from the printer File store. The current print format will be de-selected, and the printer will be "Off Line".

3.11 Delete Individual Format

This command deletes a specific format from the printer file store. If this is the currently selected/printing format then it will be de-selected, and the printer will be "Off Line".

Esc0DF {name}

where:- {name} = Actual name of Format in printer, this name must be correct and is

case sensitive.

Returns OK.

3.12 Request System Variables

Esc0ZX

Printer replies with:- a list of the System variable names and there current values.

3.13 Request Individual System Variable

Esc 0ZX {Name}

where {name} is the Actual system variable name. This will then return the value variable name and the current value. If the name is not recognised the printer will return:-

```
name ???  
OK
```

and the error "List parameter unknown" will be displayed on the terminal

3.14 Delete System Variables Changes

Esc0DX

This command restores all system variables back to the original default settings.

3.15 Delete Individual System Variable Changes

Esc0DX {name}

This command restores a system specific variable back to the original setting.

Where {name} is the Actual system variable name.

The printer replies with:-

```
OK
```

If the name is not recognised the error "Sys param unknown" will be displayed on the terminal.

3.16 Request Global Text Variable Names

Esc0ZE

Printer replies with:-

```
{nnnnnnnnn1}  
{nnnnnnnnn2}  
{nnnnnnnnn3}  
Etc.....  
OK
```

Where :- nnnnnnnn1 = First name of Global Text in file store as specified.

Where:- nnnnnnnn2 = Second name of Global Text in file store as specified.

Names of texts will always be padded to 10 characters long, spaces after name. Next text will automatically follow on next line. After last text name OK will indicate end of list.

3.17 Request Global Text Variable Information

Esc0ZE {name}

Where:-

{name} = Actual name of text in file store.

Printer replies with:-

EscGE {name} {T} {S} {O} {aa...aa}
OK

Where:-

{name} = Actual name of Global Text Variable in filestore.

{T} = Type of Global Variable Field.

{S} = style of the Type of Global Variable Field.

{O} = offset characters.

{aa..aa}= user defined variable used text fields.

If the name is not recognised the printer returns OK and the error "Global variable unknown" will be displayed on the terminal.

3.18 Request Printer Font Names

Esc0ZA

Printer replies with:-

{nnnnnnnnn1}
{nnnnnnnnn2}
{nnnnnnnnn3}
Etc.....
OK

Where :- nnnnnnnn1 = First name of Font in Printer as specified.

Where:- nnnnnnnn2 = Second name of Font in Printer as specified.

Names of fonts will always be padded to 10 characters long, spaces after name. Next font will automatically follow on next line. After last Font name OK will indicate end of list.

3.19 Request Printer Font Information

Esc0ZA {name}

Where:-

{name} = Actual name of Font in printer, this name must be correct and is case sensitive.

Printer replies with:-

Esc0A {name} {FFFFFFF} Esc {data}
OK

Where:-

{name} = Actual name of Font in Printer.

{FFFFFFF} = File size in bytes, (six numerical characters)

{data} = Actual file contents of the font.

Note: this exactly the same format in which the data was sent.

3.20 Delete all Printer Fonts

Esc0DA

The printer replies with:-

OK

This command deletes all fonts from the printer File store. The current print format will be de-selected, and the printer will be "Off Line".

3.21 Delete Individual Font

Esc0DA {name}

Where:-

{name} = Actual name of Font in printer, this name must be correct and is case sensitive.

The printer replies with:-

OK

This command deletes a specific font from the printer File store. If the name is not recognised the printer returns OK and the error "Delete file unknown" will be displayed on the terminal.

3.22 Request Global Graphic Names

Esc0ZV

Printer replies with:-

{nnnnnnnnn1}
{nnnnnnnnn2}

{nnnnnnnnn3)
Etc.....
OK

Where :- nnnnnnnn1 = First name of Graphic in file store as specified.

Where:- nnnnnnnn2 = Second name of Graphic in file store as specified.

Names of graphics will always be padded to 10 characters long, spaces after name. Next graphics will automatically follow on next line. After the last graphic name OK indicate end of list.

3.23 Request Global Graphic Information

Esc0ZV {name}

Where:-

{name} = Actual name of graphic in file store.

Printer replies with:-

Esc0V {name} {FFFFFF} Esc {data}
OK

Where:-

{name} = Actual name of Graphic in file store

{FFFFFF} = File size in bytes, (six numerical characters)

{data} = Actual graphic file data.

Note: This exactly the same format in which the data was sent.

3.24 Print Quantity (Used within a format only)

Esc0Q {qqqqqq}

Where:-

{qqqqqq} = Quantity of prints required, 000001 - 999999 sets a batch of that quantity. (000000 = continuously printing until stopped by operator)

3.25 Request Print Quantity

Esc0ZQ

Printer replies with:-

Esc0Q {000000} {,} {PPPPPP}
OK

Where:- {000000} = Quantity Value set in format design.

{,} =

{PPPPPP} = Quantity of prints done, within selected format.

3.26 Request Total Prints

Esc0ZT

Printer replies with:-

Esc0T {0000000000}
OK

Where:- {0000000000} = Total number of Print Cycles of Printer

3.27 Printer output to communications line.

After a print cycle, the printer replies on the communications port with data to help synchronise downloads or to check that printing has happened correctly. The reply data can be:-

**EscDONE or
EscREADY or
Specialist Text lines (Created with 'Y')**

This is very useful for automatically downloading the next print in a queue etc.

Because the printer has dual image memories, you can decide how to utilise the memories for the best configuration for your application. Using the Esc0SYSUPMODE feature you can determine the kind of output you require:-

Esc0SYSUPMOD = 0 (no output to communications port)
Esc0SYSUPMOD = 1 (Output to communications port, sends "DONE" after finished print)
Esc0SYSUPMOD = 2 (Output to communications port, sends "READY" as printing starts The quicker option).
Esc0SYSUPMOD = 3 (Output to communications line, sends "READY" whilst printing but gives error "Print before image update" if an update has not happened since the last print).
Esc0SYSUPMOD = 4 (Output 'Y' line to communications port after finished printing)
Esc0SYSUPMOD = 5 (Output 'Y' line to communications port, as printing starts)

Please see separate section describing in detail the printer functions etc. (see page ???)

3.28 Pause Printing

Esc0CC

This command requires no additional parameters. It causes the printer to pause printing after the operation on the current format is complete.

3.29 Resume Print

Esc0CR

This command requires no additional parameters. It causes the printer to resume printing from the point at which it was stopped by the pause print command.

3.30 Request Memory

Esc0ZM

Printer replies with:-

EscM {aa....aa} { , } {bb....bb}
OK

Where:- {aa....aa} = The Maximum file store memory. (Bytes)

 { , } = coma, field separator only.

 {bb....bb} = The amount of available memory in file store. (Bytes)

3.31 Request Format Name**Esc0ZN**

Printer replies with:-

Esc0N {nnnnnnnnnnn}
OK

Where:- {nnnnnnnnn} = The Name of the Format being printed.

3.32 Request Printer Status**Esc0ZS**

Printer replies with:-

Esc0S {nnnn} { , } {ttttttttt...t}

Where:- {nnnn} = Status Text Identifier.

 { , } = Coma (field separator)

 {ttttttttt...t} = The Status of the Printer. (Cassette off etc.)

3.33 Request Errors**Esc0ZZ**

Printer replies with:-

EscZ {nnnn1} { , } {nnnn2}
OK

Where:- {nnnn1} = Error Text Identifier.

 { , } = Coma (field separator)

3.34 Request a range of variables and parameters.**Esc0Z9**

Printer replies with:-

```
EscZ{n1} \01 {Id} \02\01{Vn}\01{R1}\02\01{R2}\02\01{R3}\02\01{Pm}\02
\01{M1}\02\01{{M2}\02\01{Pc}\02\01{S1}\02\01{S2}\02
\01\03{E1}\04\03{E2}\04\02
\01{Fn}\02
\01\03{L1}\04\03{I1}\04\03{L2}\04\03{I2}\04\02
```

Where {n1} = network number

{Id} = Printer Id

{Vn} = Version number

{R1} = Ribbon left

{R2} = Total ribbon

{R3} = Ribbon warning length

{Pm} = Prints per minute

{M1} = Total memory

{M2} = Memory available

{Pc} = Prints of current label

{S1} = Status message number

{S2} = Status message text

{E1} = First error message

{E2} = Second error message.

All error messages are output

{Fn} = Format name

{L1} = Name of first format variable

{I1} = Value of first format variable

{L2} = Name of second format variable

{I2} = Value of second format variable

All variables used in a format , Global or Local will be listed.

3.35 Clear Errors

Esc0DZ

Allows user to clear all errors from a computer.

The printer replies with:-

OK

3.36 Initialise Part Printer

Esc0CINIT

This command erases all stored global graphics, global variables, formats and fonts.

Only the System Variables & Mechanical Variables are left in the printer with their previously programmed values. (Must be in capital letters.)

3.37 Initialise All Printer

Esc0CINEW

This command erases all programmed system variables, global graphics, global variables, formats and fonts.

The software default settings (System Variables & Mechanical Variables) will then be active.

Basically any programmed settings you have made will be totally cleared. (Must be in capital letters.)

3.38 End of Format Command

Esc0K

This Command identifies the end of a Format file.

3.39 Insert and overwrite a local Variable Text field.

Command superseded by UE ! Still works for compatibility.

This command allows you to insert and overwrite an existing local variable field within the current format being printed.

Esc0I {name} {aa...aa}

Where:-

{name} = Actual Name of Variable field you are replacing.

{aa...aa} = Information to overwrite exiting Text field, the string length must be exactly as the original variable.

3.40 Update and overwrite a local graphic field.

This command allows a local graphic embedded within a format to be updated without reloading the whole format. The format will be edited in the file store and the image will be updated.

Esc0UV {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {c {d...d}..c}

See section 1.41 for details of all the parameters

3.41 Program Printer

Esc0CIPROG

This command generates a printer reset. A new control program can then be downloaded from the computer as the printer restarts. (Must be in capital letters)

3.42 Loading a file to the Printer

**Esc0H {name} {n} {XXXXXX} {NNNNNN}
Esc {file of data}**

Where:-

{name} = Name of file being sent to printer (15 characters including spaces)

{n} = Application Identifier. (see list below)

{XXXXXX} = Size of file being loaded to the printer
 {NNNNNN} = Length of file transferred

Identifier types:-

Terminal files	=	1
Printer files	=	2
Font files	=	3 (Truetype or SFP fonts)
Format Files	=	4
System files	=	5 (Internal generated files)
Graphic files	=	6
Burn Files	=	7
Configuration files	=	8
Variable files	=	9 (used with the "Y" command)

This command is used for restoring printer status and downloading burn and configuration files. The identifier type is the equivalent of the file extension for files in windows.

3.43 Request All Files

Esc0ZH{T}

This command allows you to view all files in the printer file store, with their type identification numbers.

Example of files received:-

```

ARIAL _____ 3
TETLEY ONE _____ 4
TETLEY TWO _____ 4
LOT _____ 8
BEST BEFORE _____ 8
OK

```

The {T} type parameter is optional. If it is included only files of that type will be listed.

3.44 Request individual file information

Esc0ZH {name}{T}

This requests that a particular named file is uploaded from the printer. The name must be made up to 15 character with additional spaces. The type number must be included

The printer replies with:

```

{All file data}
OK

```

3.45 Delete a file from file store

Esc0DH {name} {n}

Where:-

{name} = Name of file being sent to printer (15 characters including spaces)

{n} = Application Identifier. (see list section 2.85)

Printer replies with:- OK

3.46 Selecting a new Burn File

Esc0X {SYSBFILE} {name of file}

Where:-

{SYSBFILE} = Name of the Variable being used

{name of file} = Name of the file being loaded to the file store.

Note!

The printer should then be switched off and then re-started to ensure the format has been updated.

3.47 Text Line output to Communications Port

This feature allows a user defined text line to be generated and sent via a communications port to an attached computer. The text is regenerated and sent after every print. The text can be fixed or contain a mixture of fixed and variable data exactly like a text field on a label. A standard printer status text field can also be included after the user-defined data.

This command only works if the variable SYSUPMOD is activated. See below for the options:-

Esc0SYSUPMOD = 0 (no output to communications port)

Esc0SYSUPMOD = 1 (Output to communications port, sends "DONE" after finished print)

Esc0SYSUPMOD = 2 (Output to communications port, sends "READY" as printing starts The quicker option).

Esc0SYSUPMOD = 3 (Output to communications line, sends "READY" whilst printing but gives error "Print before image update" if an update has not happened since the last print).

Esc0SYSUPMOD = 4 (Output 'Y' line to communications port after finished printing)

Esc0SYSUPMOD = 5 (Output 'Y' line to communications port as printing starts)

The text output line is defined in a format as follows:

Esc0Y { P } { S } {Fixed Text} {Variable Text} {Fixed Text} ...

Where:-

{P} = Communication port number.

{S} = Status of Printer. (0 = not included, 1 = included)

{Fixed Text} = Fixed information.

{Variable Text} = Variable Name, actual value of the variable information output to communications port.

{Fixed Text} = Fixed information.
 ... = More fixed and variable sections

You would normally want the text output to the communications port that from which you send the format (set value of 0.) However, you may wish to send the data to a different computer via a different port. This is controlled with the communication port number:-

0 = Active communications port
 1 = Serial Port
 2 = USB Port
 3 = Ethernet (TCP/IP) Port
 4 = Secondary Ethernet (TCP/IP) Port, we might add this later.

Example Format:-

```
0FCounter
0PSPEED 0100
0PBURN1 0240
0PBURN2 0240
0PPRESSURE 20
0PHOMOFF 03.0
0PROTATION 0
0ECounter1 40001,1,1,1,1,9999
0ESeconds 36
0TArial 0010010016001Box No.\0Counter1\0
0Y01COUNTER1, \0Counter1\0;TIMES,\0Seconds\0:
0Q00000
OK
```

Example of Output from Printer:-

```
<ESC>Y01COUNTER1,0001;TIMES,33:
6108:Print on Line
OK
```

3.48 Request Mechanical Variables

Esc0ZY

Printer replies:-

```
Esc0X {M1}
Esc0X {M2}
Esc0X {M3}
.....
OK
```

Where {M1} {M2} {M3} are list of the mechanical variable names and there current values.

Please see separate manuals for Mechanical Variables & System Variables, that have a full description of the function and ranges available.

3.49 Request Individual Mechanical variable

Esc 0ZY {Name}

Where {name} is the Actual system variable name. This will return the variable name and the current value.

Please see separate manuals for [Mechanical Variables & System Variables](#), that have a full description of the function and ranges available.

3.50 Request all Print Parameters

(Within the format being printed)

Esc0ZP

Printer replies with all format parameters, and the associated values.

```

EscXSPEED      0050
EscXBURN1     0450
EscXBURN2     0450
EscXPPRESSURE 20
EscXPEEL      07.0 (Not normally used)
EscXR0TATE    0
EscXOFFSET    050.0 (Continuous Machines only)
EscXHOMOFF   26.0 (Intermittent machines only)
OK

```

3.51 Request an individual Print Parameter

(Within the format being printed)

Esc0P {name}

Where:-

P	=	Escape code for Print parameter Variables.
{name}	=	Description of Parameter Field to change. This field must be ten characters long with printable ASCII characters followed by trailing spaces. (Names of fields are pre-defined as list below)

Printer replies with:-

```

EscX{NNNN}{VVVV}
OK

```

Where:-

{NNNN}	=	Name of parameter requested
{VVVV}	=	Value of parameter requested

3.52 Request format variables

Esc0ZI

Printer replies with:-

```

{nameL_1} {GenTextL_1}\09{nameL_2} {GenTextL_2}
{nameG_1} {GenTextG_1}\09{nameG_2} {GenTextG_2}

```


OK

Where:-

{nameL_1}	=	Name of first local variable used
{GenTextL_1}	=	Current generated text from variable
...		
{nameG_1}	=	Name of first global variable used
{GenTextG_1}	=	Current generated text from variable
...		

4 How to use the printer in special situations.

4.1 How to make a Thermocode 2 printer faster.

Open Date Thermocode 2 printers can be used straight out of the box in most cases. They simply need suitable formats and fonts to be loaded and you can start to print.

However, to make the printers universally usable like this we have to make assumptions about how they are used. These assumptions can reduce the printers speed. If some of the assumptions do not apply in a particular case then it is possible to change the printer settings and the way it is used and it will then be possible to run faster.

What do we mean by speed? There are a number of things that could be called the printer speed:

- The number of prints per minute.
- The time to download new information.
- The time to update the format information and to generate the initial image.
- The cycle time for a single print.

Each of these aspects of speed can be adjusted for a particular application. In some cases the adjustment can be used on all kinds of printer, others are particular to Intermittent, Continuous or Economy.

4.1.1 The number of prints per minute.

4.1.2 The time to download new information.

Download speed is normally limited by the serial port baud rate setting which is 115200 by default. The following list runs from fastest to slowest:

- Use the built-in USB port.
- Use TCP/IP.
- Change the serial port baud rate to 230400.

You can also reduce the amount of information sent to the printer:

- When downloading fonts, use smaller fonts, possibly .sfp.
- When downloading graphics, use a compressed .pcx or .lz graphics format.

4.1.3 The time to update the format information and to generate the initial image.

This can be split into three sections, initial generation, internal updates such as in-built counter fields and dates and download updates where new image data is set to the printer from a computer.

To speed up initial image generation:

- Reduce download time, see above.
- Use a .sfp font. This is a fixed size and is quicker to generate than a true type font.
- If using multiple fonts set SYFASTFON to the number of fonts in use.
- Use compressed graphics, .pcx or .lz.
- Reduce the text size. Bigger text, slower updates.

To speed up internal updates:

- Use a .sfp font. This is a fixed size and is quicker to generate than a true type font.
- If using multiple fonts set SYFASTFON to the number of fonts in use.
- Reduce the text size. Bigger text, slower updates.

To speed up download updates:

- Reduce download time, see above.
- Use a .sfp font. This is a fixed size and is quicker to generate than a true type font.
- If using multiple fonts set SYSFASTFON = the number of fonts in use.
- Use compressed graphics, .pcx or .lz.
- Reduce the text size. Bigger text, slower updates.
- Start download as soon as the current label starts to print. Set SYSUPMOD = 2 and the printer will send READY as it starts to print.
- If every label is different from the last set SYSUPFAST = 2. This only updates one memory plane and does not write to file, so if the printer is powered OFF and ON the last update will not be remembered. You can also set SYSUPMOD = 3. This will cause a printer error if there has been no update since the last print.
- Use a graphic (compressed) instead of text. Although graphics take longer to download they are quicker to draw in the printer. Experiment to see which is best.

4.1.4 The cycle time for a single print.

Reduce the height of the head above the rubber when waiting to print. Set SYSHEIGHT to a smaller value.

4.2 Using the printer on a cross-web mounting.

When using a continuous printer on a cross-web mounting there are a number of system variables that have to be set correctly:

SYSCORNUM Sets the number of prints across the web.
SYSCORLEN Sets the distance between each print on the web.

With these two variables set a single trigger pulse will cause the printer to print SYSCORNUM images SYSCORLEN apart.

Additional control is then provided by:-

SYSCORNOU Prevents updates during a web sweep.

If there are time or date updates on the label there can be problems updating the image between the prints across the web. Setting SYSCORNOU prevents these updates and will allow a faster cross-web and closer repeats.

If SYSCORLEN is very small there may not be time to raise and lower the head and return the ribbon between prints. This can be overcome with:-

SYSCORFAST Keeps head down and runs ribbon for full cross-web sweep.

This may however use too much ribbon.

Also look at:-

SYSMCSPEED
SYSCMCLR

Which control operation when the cross-web speed drops too low.

4.3 How to use multiple image planes to print multiple images

5 System variable listing

6 Mechanical variable listing