Standard Practice for
Data Exchange Format for Sewn Product Plotting Devices[1]

1. Scope

1.1 This practice describes a data format for transferring information from a sewn product computer aided design software program to a device that produces physical output, typically in the form of a printed or drawn image on paper.

1.2 This practice is based on a subset of the Hewlett Packard Graphics Language HPGL/2. Supported syntax and limitations are listed in 7.2. Unsupported syntax is listed in X1.1.

1.3 This practice only supports *X-Y* vector data and a limited set of additional functions. No provision is made to support bitmap/raster data used in applications like inkjet printing.

1.4 This practice supports a single system of units, an image fixed at 100 % scale and 1:1 aspect ratio. Scaling and custom unit systems are not supported.

1.5 This practice does not support curve interpolation or definitions. All curves are represented by discrete vectors (stroked) and are dependent on the resolution of the CAD software.

1.6 This practice requires that all coordinates are absolute, not relative, as defined in the HPGL/2 reference.

1.7 This practice only supports positive coordinates that are measured from a single *X-Y* origin point with coordinates 0,0.

1.8 This practice only supports fixed width fonts. Variable width fonts are not supported.

1.9 This practice intends to transfer a static image with no provision for editing.

1.10 This practice assumes monochromatic output. It does not support implied output colors.

1.11 This practice imposes no limits on the width or length of the plot data. Physical limitations imposed by the hardware and their effects on the output are the responsibility of the hardware manufacturer.

1.12 This practice does not support frame advance commands or any methods that insert multiple origin points or floating coordinate systems.

1.13 This practice limits the plot file to contain a single block of data demarked by a compatible header and terminator. Multiple blocks of data in a single file are not allowed.

1.14 The intended application of this practice is limited to the class of output devices found in the sewn product industries that produce apparel, textiles, upholstery, and others that use soft or semi-rigid materials.

1.15 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

2. Referenced Documents

2.1 *ASTM Standards:*[2]
D6963 Terminology Relating to Sewn Products Automation

3. Terminology

3.1 *Definitions:*For definitions used in this practice, see Terminology D6963.

4. Summary of Practice

4.1 Plot data are organized to allow a parsing algorithm to recognize adherence to the ASTM format then are written as a data block into a computer file.

4.1.1 A specific group of commands are required at the beginning of the data block, called a header.

4.2 Plot data follow a subset of the HPGL/2 ASCII language.

4.2.1 The practice specifies syntactical and parametric constraints to limit ambiguity.

4.2.2 Basic HPGL formatting is supported. Advanced HPGL functions are beyond the scope of the practice.

5. Significance and Use

5.1 This practice provides a common format that allows a computer design system to generate data that an output device can accurately reproduce independent of the hardware manufacturer.

---

6. Plot File

6.1 Plot data are written in data blocks of ASCII characters, one data block per computer file.

6.2 Data blocks are made up of commands.

6.2.1 Each command is a two-letter sequence (see 7) always written in CAPITAL letters.

6.2.2 An instruction and its associated ed parameters (if required) are terminated by the semi-colon (;) terminator. Other terminators are not supported that is mnemonic, <CR>, or <LF>.

6.3 Some commands are followed by parameters.

6.3.1 Parameters immediately follow the command with no separating space and come before the command terminator.

6.3.2 Multiple parameters are separated by a comma (,). Other separators are not supported that is blank space, etc. Strings of coordinates are not allowed; that is, PDX,Y,X,Y,X,Y.

6.4 Blocks of data that adhere to this practice begin and end in a specific way.

6.4.1 Blocks begin with specific commands in a defined order, commonly called a header, as follows (information in brackets, [ ], are variable):

IN; CO "ASTMXXXXX-XX"; CO "Author: [Author Name]"; CO "Creation Date: [DD-MM-YYYY]"; CO "Creation Time: [HH-MM on a 24 hour time scale]"; PA; DTETX,1; LM0; [All subsequent plot file data follow this header]…

6.4.2 Blocks are terminated with the ASCII file separator character, FS (decimal 28).

6.5 All coordinate data use the HPGL/2 default plotter-unit system.

6.5.1 1 plotter-unit = 0.025 millimeter (mm)

6.5.2 40 plotter-units = 1 mm

6.5.3 1016 plotter-units = 1 inch (in.)

6.5.4 3.39 plotter-units = 1 dot at 300 dots per in. (dpi)

7. Syntax

7.1 Syntax commands are used to govern interpretation of formatted data. The following is a list of the supported syntax and a brief description. Detailed summaries of the commands are outlined in 7.2. HPGL/2 commands not listed below are not supported by the practice and are documented in Appendix X1.

7.1.1 *CO*—Comments,

7.1.2 *DI*—Absolute Direction,

7.1.3 *DT*—Define Label Terminator,

7.1.4 *IN*—Initialize,

7.1.5 *LB*—Label,

7.1.6 *LM*—Label Mode,

7.1.7 *LT*—Line Type,

7.1.8 *PA*—Plot Absolute,

7.1.9 *PD*—Pen Down,

7.1.10 *PU*—Pen Up,

7.1.11 *SI*—Absolute character Size, and

7.1.12 *SP*—Select Pen.

7.2 The following descriptions outline the implementation of HPGL/2 syntax commands allowed by this practice. Unless otherwise noted, usage exactly follows the specification described in the agreed HPGL/2 reference.

7.2.1 *CO*—Comments. Parameters between quotes are ignored by the output device. This practice requires four sequential CO commands containing specific text as header data. These must immediately follow the IN; and precede the PA; syntax at the beginning of the file and use the format that follows. Information in brackets, [ ], are variable. Example:

CO "ASTMXXXXX-XX"; CO "Author: [Author Name]"; CO "Creation Date: [DD-MM-YYYY]"; CO "Creation Time: [HH-MM on a 24 hour time scale]";

7.2.2 *DI*—Absolute Direction. Specifies the direction characters in a label are drawn. In this practice, DI syntax only affects usage of LB and SI. Default values are DI1,0. Example:

…DI1,0;...

Result: defines label direction as horizontal, the default for this practice.

7.2.3 *DT*—Define Label Terminator (header only). Indicates the end of a text string associated with an LB command. This practice requires a single DT command near the end of the header immediately following the PA syntax. The required arguments for DT are ETX (decimal 03) and mode 1. These are the only allowed parameters. No other instances of DT are allowed. Example:

… PA;DTETX,1;…

Result: defines the label terminator as ETX (decimal 03). Mode "1" indicates that this character is not printed.

7.2.4 *IN*—Initialize (header only). Resets the configuration of the output device to defaults and raises the logical pen, equivalent to a PU; syntax command. Also indicates the beginning of a plot file conforming to this practice and is always the first syntax command in the header. No other instances of IN are allowed.

7.2.5 *LB*—Label. Prints the subsequent text string using the currently defined font. Requires the label terminator, ETX, at the end of the text string, as defined by the DT command included in the header. In this practice, LB syntax is only affected by the DT, LM and SI syntax. Example:

…LBSM-FrontETX;

Result: prints "SM-Front" at the current pen position.

7.2.6 *LM*—Label Mode (header only). Determines whether the LB command interprets characters in 8 or 16 bit segments. This practice requires a single LM command at the end of the header immediately following the DT syntax. The required argument for LM is 0 (8-bit mode). This is the only allowed parameter. No other instances of LM are allowed. Example:

… DTETX,1;LM0;…

Result: 8-bit mode, each byte is a character.

7.2.7 *LT*—Line Type. Determines the pattern of the drawn line. LT is allowed in two forms. LT; creates a solid line (the default line type). Non-solid line types are available only by using LT with three required parameters; line_type, pattern_length (always in millimeters), and mode (always 1).This practice only supports line types +2 through -2 (including 0) as specified in the HPGL/2 reference. Unsupported line types require stroked equivalents. The line type is used until a subsequent LT is encountered. In this practice, LT only affects the PD command. Examples: