To: Distribution

From: J. Falksen, E. J. Wallman

Date: 11/29/79

Subject: Compose Device Table Compiler (compdv)

This MTB proposes and discusses a reduction compiler translator for the creation of device tables for compose. See MTB 387 for an introduction to this subject and a list of definitions.

THE COMMAND

The compose device table compiler (compdv) allows a designer to create compose device tables for various devices. The input is a device description file in fairly free format that specifies the characteristics of the device, all of its fonts, and how to put them together.

Name: compdv

The compdv command translates a device description input file into an object segment containing a coded binary structure for use by compose in preparing formatted output for the described device. See ???? for a discussion of the device description file.

Usage: compdv path {-control_args}

where:

1. path

   is the path name of the input device description file. The entryname of the file must be of the form <device>.compdv, but the suffix need not be given in the command. The output segment is created (if it does not already exist) in the working directory with the entryname <device>.comp_dsm and may have various additional names attached by the translator. The star convention is not supported.

2. control_args may be chosen from:

-check, -ck
: process the input file, making all syntax checks and creating the ALM source intermediate file, but do not invoke the ALM assembler and do not delete the ALM source file. The default is to invoke the ALM assembler at the end of an error free translation and to delete the ALM source file.

-list, -ls
: create an ALM assembly output listing for the translation. The default is no listing.

THE DEVICE DESCRIPTION FILE

The input device description file consists of six (6) parts that must appear in the order shown:

Symbol Declarations (optional, distributed)

Global Font Values

Global Device Values (distributed)

Font Tables

Size Tables

Device Tables

GENERAL SYNTAX

LITERALS

Any place a quoted-string is mentioned, it means a string delimited by the double quote character ("). If a quote is needed within such a string, it must be doubled. For example:

```
"A quoted string"
"A ""quoted"" string"
```

COMMENTS

At any place in the source where the syntax allows white space to appear (except within a quoted string), a comment may be placed. A comment is any string beginning with /* and ending with */. For example:

```
/* This is a comment /
/* And this is a
   multiline comment /
```

NAMES

Anywhere <name> is mentioned it means a string of not more than 32 characters beginning with an alphabetic followed by an arbitrary [illegible] of alphanumerics and/or underscores. All <name>s in a device description file must be globally unique. For some usages, <name> is restricted to being shorter than 32 characters. The restrictions are given in the discussions of the various usages. For example:

```
A
name
here_is_1
```

INPUT

<input> is a single character given by either of:

```
ooo
     3 octal digits
"x"
     any single quoted character
```

RANGE

<range> is an inclusive ordered set of characters given as <input>-<input>. For example:

```
"A"-"Z"
     the uppercase alphabet
000-007
     the first eight ASCII control characters
```

OUTPUT

<output> is a blank separated list of elements selected from the following:

```
ooo
     3 octal digits
"xxx"
     any quoted string
XXX
     any declared symbol (See Symbol Declarations below)
nn(<output>)
     nn repetitions of an <output> string
SELF
     When used in font definitions, means the graphic being
     defined.  This is a reserved word; it may not be used as
     a <name>.
```

For example: (from dtc300s artwork)

```
PLOT PAD3 8("." RLF) 5(". ") " " UNPLOT LF
```

SYNTAX OF THE SECTIONS

SYMBOL DECLARATIONS

Symbols that represent output character strings may be defined for convenience in constructing font tables. All such symbols must be defined before their use. The Symbol Declaration section is not a formally delimited section, but consists of any number of the following declaration statement distributed randomly throughout the file.

```
dcl: <name>, <output>;
```

<name>
is the name of the symbol being defined and is restricted to a maximum length of 8 characters.

<output>
is the character string to replace a reference to the symbol.

For example: (from dtc300s)

```
dcl:   PLOT,   006;
dcl:   UNPLOT,033 PLOT;
dcl:   RLF,    032;
dcl:   LF,     012;
dcl:   PAD3,   3(177);
```

GLOBAL FONT VALUES

The Global Font Values section must appear in the file before any Font Table, Size Table, or Device Table sections and must contain the following items. The items may be given in any order.

EMunits: <integer>;
An arbitrary number of dimensionless units into which an EM space is divided for the definition of the widths of characters in the font tables in this file. This number essentially measures the resolution of the target device; the larger it is, the finer grain the output. For example: (from ascii, where the resolution is a single 10-pitch character)

```
EMunits: 1;
```

Spaceband: <min>,<avg>,<max>,<output>
The range of allowable interword space for the device defined in this file, given in EMunits.

<min>, <avg>, and <max> specify the minimum, average, and maximum values, respectively, are given in EMunits, must obey the relation:

```
0 <= min <= avg <= max
```

and are defined as:

<min> the least amount of interword space.
<avg> the average amount of interword space. This is the amount used for all unjustified lines and for any spaceband that is left in the output line by the internal justifying routine of compose.
<max> the maximum amount of interword space allowed before hyphenation is attempted. Note that justified lines may contain more than <max> space, but only in case hyphenation and letterspacing fail or are not allowed.

<output> is the character string to be emitted for spaceband replacement whenver one appears in the output stream.

The values defined here are used to generate a default <graphic-definition> for the spaceband of "040, <avg>, <output>;" in each defined font.

For example: (from dct300s where EMunits is 6 reflecting the plotting increment)

```
Spaceband: 3,6,18," "
```

GLOBAL DEVICE VALUES

The Global Device Values section is not a formally delimited section, but consists of any number of the following statements distributed randomly throughout the file after the Global Font Values section. The statements define values that apply to all device tables following their appearance. All have local counterparts to specify different values for a particular device.

Data dependencies affect the order in which certain statements may appear. Any such restrictions are given in the descriptions of the affected statements.

The statements also all have default values that describe the default (ascii or printer) device. Unless otherwise noted in the text, the default values are those shown in the individual examples.

Units: <keyword>;

the physical units in which space values are given. Space values are given as normal decimal numbers, e.g., 2, 14.7, 0.025, etc. The valid keywords are:

10	10 pitch monospace characters
12	12 pitch monospace characters
in	inches
mm	millimeters
pc	Typographic picas (6 picas = 1 inch)
pt	Typographic points (72 points = 1 inch)
pp	Picas and points as a decimal number

```
Units: pt;          /* points /
```

Artproc: <name>{$<name>};

the entryname and optional entrypoint of the procedure that supports special artwork features for the device. This entry is normally needed only for devices having graphic features beyond the scope of plotting and simple typographic rules. The default entryname is derived from the name of the device (see Outproc below). (See artproc in comp_dvt.incl.pl1 later in this document for calling sequence.) (Note: This interface is not yet active due to lack of a specific application. Its projected use is for the processing of half-tone raster files and generalized graphics files.)

```
Artproc: ascii_writer_$artproc;
```

Attach: <quoted-string>;

the attach description to use for the output switch when the -of argument has NOT been given. If not given, no on-line output is possible for the device.

```
Attach: "syn user_output";
```

Cleanup: <output>;

the control string that must be sent to the device to restore its normal mode of operation when interrupted in the middle of output. This string is required for plotting terminals to take them out of PLOT mode when interrupted.

```
Cleanup: "";        /* no cleanup needed /
```

Comment: <quoted-string>;

Comment is a string which is kept in the device table for the sole purpose of being passed on to the compout file as a part of its header. It is used by the process_compout command (See AZ98 ???) when transcribing the file onto the output medium.

```
Comment: "";        /* null comment /
```

DefaultMargs: <Units>, <Units>, <Units>, <Units>;

the default values for the top, header, footer, and bottom page margins, respectively. This feature allows for devices (such Braille embossers) that demand page margins other than those normally assumed for a printed document.

```
DefaultMargs; 48,24,24,48; /* 4,2,2,4 lines /
```

DevClass: <quoted-string>;

the class of the device. This string is placed in the output file header for use by the process_compout command (See AZ98 ???) and is used to set the DeviceClass builtin of compose.

```
DevClass: typewriter;
```

DevName: <quoted-string>;

the generic name of the machine within DevClass for which the Device Tables in this file provide support, e.g., V-I-P, Dymo, APS within "photocomp" or dtc300s, selecterm within "diablo". Note that within a generic device, such as dtc300s, there may be different specific devices (See Device below) for minor differences such as running in 12-pitch rather than 10-pitch. This string is also used to set the DeviceName builtin of compose. The default is the name of the first Device Table.

```
DevName: ascii;
```

Endpage: <input>;

the font character to select the page eject sequence for the device. A value of 000 means that there is no eject sequence.

```
Endpage: 000;       /* ascii /
```

Footproc: {<name>{$<name>}}{, <fontname>};

the optional entryname and entrypoint of the procedure to process footnote references and the optional font for them. The default entryname is derived from the name of the device (see Outproc below) and the default font is the default font for the device. (See footproc in comp_dvt.incl.pl1 for the calling sequence).

```
Footproc: ascii_writer_$footproc, ascii;
```

Interleave: <switch-value>;

the setting of the line sorting switch for compose. <switch-value> may be either of the keywords "on" or "off". If the switch is set on, the output in the page image structure is sorted by compose so as to appear in strictly increasing page depth order because the device does not support reverse leading to return to the top of the page for multi-column output. If the switch is set off, the output lines appear in the page image by page depth within the columns, each column being a sub-array in the structure. The default value for the switch is off; it must be set on for device with DevClass values of "typewriter", "diablo", or "printer."

```
Interleave: on;     /* sort output */
```

Letterspace: <integer>;

the maximum amount of interletter space allowed, given in EMunits.

```
Letterspace: 0;     /* not allowed /
```

MaxPages: <integer>;

the maximum number of pages to be contained in an output block for the device. The process compout command (See AZ98 ???) will produce physical output blocks containing no more than this number of output pages. Input blocks for some devices are limited by such factors as size of paper tape input reel, capacity of tape cassette or film magazine, etc. The default value means that the input capacity of the device is unlimited and all output will be a single physical block (or file).

```
MaxPages: -1;       /* unlimited /
```

MaxPageLength: <Units>;

the maximum length of a page. The default value means that the device will accept pages of unlimited length.

```
MaxPageLength: -1; /* unlimited /
```

MaxPageWidth: <Units>;

the maximum width of an output page.

```
MaxPageWidth: 979.2; /* 136 columns /
```

MinBotMarg: <Units>;

the minimum page bottom margin for the device.

```
MinBotMarg: 0;      /* ascii /
```

MinLead: <Units>;

the minimum amount of "lead" (vertical spacing) available in the device.

```
MinLead: 12;        /* 1 line /
```

MinSpace: <Units>;

the minimum value of horizontal space available in the device.

```
MinSpace: 7.2;      /* 1 column /
```

MinTopMarg: <Units>;

the minimum page top margin for the device.

```
MinTopMarg: 0;      /* ascii /
```

Outproc: <name>{$<name>};

the entryname and optional entrypoint of the procedure that converts the coded page image structure constructed by compose into a character stream acceptable to the device. This is the procedure that translates internal signal bytes into device control codes. The default entryname for the device described in <device>.compdv is <device>_writer_. (See outproc in comp_dvt.incl.pl1 for the calling sequence and page_image in comp_page.incl.pl1 for the page image structure, both later in this document.)

```
Outproc: ascii_writer_;       /* ascii device /
```

Sizes: <name>;

the name of the default Size Table. <name> must have already been defined as the name of a Size Table section.

```
Sizes: <first Size Table name>;
```

Stream: <switch-value>;

the setting of the compout file type switch for compose. <switch-value> may be either of the keywords "on" or "off". If the switch is set on, the compout file written when the -output_file control arg of compose is given will be an ASCII stream file suitable for processing with the print and dprint commands as well as the process_compout command. If the switch is set off, the compout file is a sequential file containing coded binary device information that must be processed with process_compout command (See AZ98 ???). Normally, this switch will be set on only for the ascii and printer devices, but may be used for any other device that has only those features commonly found in ASCII terminals or could be treated (by Multics) as a line printer. The default value for the switch is off; it must be set on for the ascii device.

```
Stream: on;
```

TapeRec: <integer>;
the length of records to be used when writing to a tape. The default value means that records may be of any arbitrary length.

```
TapeRec: -1;        /* unlimited /
```

FONT TABLE SECTION

A font table specifies the width and output string for each character contained in a font. In this context, a "character" is a 9-bit byte placed in the output page image by compose. This byte may be a normal ASCII graphic or a coded signal for some other output sequence.

A device description file may specify up to 100 fonts; each font table beginning with a "Font" statement and ending with the beginning of any Size Table, Device Table, or other Font Table. Global Device Values and Symbol Declarations may also appear within a Font Table section.

A Font Table section consists of exactly one "Font" statement followed by any number of <graphic-definition>s and <artwork-definitions>s.

```
Font: <name> {like <fontname> | use <fontname>};
```

The mutually exclusive optional clauses are:

like <fontname>
causes the entire contents of the previously defined referenced font to be copied into the new font as initial values. A new font normally contains only the default spaceband when initialized.

use <fontname>
the new font shares character storage with the previously defined referenced font and has no character storage of its own. This feature is used when two fonts are logically identical, but physically different in the target device (e.g., two different print wheels on the dtc300s such as PICA-10 and OCRB). When this form is used, no additional <graphic-definition>s or <artwork-definitions>s are allowed.

<graphic-definition>s

Each <graphic-definition> is a triplet of the form:

```
     <graphic>,<width>,<replacement>;
```

where:

<graphic>
is the graphic being defined and may be any of:

```
<input>
(<range>)
(<input> ¦ <range>, <input> ¦ <range>, ...)  ie,
 an arbitrary sequence of <input>s and/or <range>s.
```

or any of the following keywords:

EM	(EM space)
EN	(EN space)
thin	(thin space)
EM-	(EM dash)
EN-	(EN dash)
hyphen	(hyphen)
EM_	(EM align dash)
EN_	(EN align dash)
PS	punctuation space
``	opening double quote
''	closing double quote
^0	superior digits
^1	
^2	
^3	
^4	
^5	
^6	
^7	
^8	
^9	

(These keywords may be thought of as "builtin" symbols that must be assigned values if they are to be included in a font. Note that "hyphen" must be assigned a value in order that the hyphenation mechanism in compose may work.)

<width>
is the visual width of the graphic given in EMunits. Negative values are allowed.

<replacement>
is the output string to be transmitted to the device and may be any of:

<output>

a string constructed from octal values and graphics from this font.

<fontname> <graphic>

a copy of a <graphic> in the referenced <fontname> with an emitted signal to the device to change to the referenced <fontname>. <width> must omitted with this form.

<fontname> <output>

a string constructed from octal values and graphics FROM THE REFERENCED <fontname> with an emitted signal to the device to change to the referenced <fontname>. The string is constructed in this font.

If <replacement> is omitted, then the existing <replacement> for the <graphic> is used. If <replacement> AND ITS SEPARATING COMMA are both omitted, then SELF is assumed. Certain combinations of <graphic>, <width>, and <replacement> are invalid. These are diagnosed and reported by the translator.

For example:

```
Font: ascii;
004,0,"";               /* signals needed for APL manual */
005,0,"";
(016,017),0;            /* red/black ribbon shifts */
(041-"~"),1;            /* all the printing graphics */
(010,033),-1;           /* BSP & ESC have width -1 */
014,0;                  /* FF for line printer */
177,0;                  /* PAD character */
EM,2,040 040;           /* 2 spaces */
EN,1," ";               /* 1 space */
EM-,2,"--";             /* double dash */
EN-,1,"-";              /* single dash */
EM_,2,"__";             /* double underscore */
EM_,1,"_";              /* single underscore */
thin,1," ";             /* cant get less than 1 space */
303,3,"(c)";            /* copyright mark */
261,1,"†";              /* dagger */
301,1,"‡";              /* double dagger */

Font: underline_alpha like ascii;
("a"-"z"),1,SELF 010 "_";       /* underscore lowercase */
("A"-"Z"),1,"_" 010 SELF;       /* underscore uppercase */
222, 5, "M" 010 "N" 010 "W" 010 "Z"; /* a blacked-in box */
                                /* Note that canonical
                                   form is preserved */

Font: funny like ascii;         /* like ascii but */
("0"-"9"),,ascii_ SELF;         /* with underscored numerals */
```

```
222,, underlined_alpha 222;    /* no need to type it all again! */
221, 3, ascii, "...";          /* ellipsis */
```

<artwork-definition>s

<artwork-definition>s are similar to <graphic-definition>s except that they describe "extended" or manufactured graphics (at least on some devices) and are quadruplets of the form:

```
art <artname>,<width>,<replacement>;
```

where:

<artname>
the conventional name of the artwork construct or element selected from any of the following name groups.

This first group are graphics that are complete in themselves (the so-called "one-highs").

[	one-high [	]	one-high]
{	one-high {	}	one-high }
(	one-high (	)	one-high)
¦	one-high ¦	¦¦	one-high ¦¦
o	bullet	X	one-high X
cb	change bar	d	delete star
/	one-high /	\	one-high \
cm	copyright mark	tm	trademark
^	up arrow head	v	down arrow head
<-	left arrow head	->	right arrow head

This group contains graphics that are parts of larger artwork constructs, e.g., boxes, diamonds, and lozenges.

D^	diamond top	Dv	diamond bottom
D<	diamond left	D>	diamond right
Clf	left half-circle	Crt	right half-circle
-rul	horizontal rule	¦rul	vertical rule
/rul	right slant rule	\rul	left slant rule

This group contains the parts for the multi-line math symbols. The graphics for any symbol form a consistent set; if a math symbol is to be defined, all the parts must be given.

"["	"]"	"{"	"}"	"("	")"	"¦"	"¦¦"	
[tp	]tp	{tp	}tp	lptp	rptp	¦tp	¦¦tp	top parts
[ht	]ht	{ht	}ht	lpht	rpht	¦ht	¦¦ht	half top parts
[md	]md	{md	}md	lpmd	rpmd	¦md	¦¦md	middle parts
[hb	]hb	{hb	}hb	lphb	rphb	¦hb	¦¦hb	half bottom parts
[bt	]bt	{bt	}bt	lpbt	rpbt	¦bt	¦¦bt	bottom parts
[fl	]fl	{fl	}fl	lpfl	rpfl	¦fl	¦¦fl	filler parts

(Note: Because the left and right parentheses are used as part of the syntax of the device table compiler language they may not be used in forming tokens; hence the need to use the "lp" and "rp" constructs for their <artname> parts.)

<width>
is the visual width of the graphic given in EMunits.

<replacement>
is the output string to be transmitted to the device as for <graphic-definition>s above.

SIZE TABLE SECTION

A Size Table is a list of allowable pointsize values that may be used in conjunction with any number of Fonts in any number of Device Tables. That is to say, a Size Table may be referenced any number of times and may be used with one Font in some Device Table and a different Font in some other Device Table. A device description must contain at least one Size Table.

A Size Table section consists of exactly one Size statement of the form:

```
     Size: <name>, <Units>{, <Units>}...;
```

where:

<name>
is the internal reference name of the pointsize list being defined.

<Units>
is a value to be entered into the list. At least one <Units> value must be given. For example:

```
          Size: pitch10, 7.2;
```

DEVICE TABLE SECTION

A Device Table describes a specific device and provides the data needed by compose to format output for that device. The data in the table are gathered from default values, Global Device values, Font Table references, Size Table references, and local device values. There can be any number of Device Tables in a device description file, either describing different machines that are similar enough to share many attributes, or different configurations of the same machine.

Font Tables and Size Tables may be freely shared among Device Tables. However, if a font "borrows" from some other font, then both the "loaner" and "borrower" Font Tables must be included in the Device Table. For example the Mergenthaler V-I-P font "ascii" is based on the Clarinda font but it borrows a few characters from news/commercial Pi (NCPi). Hence, if the "ascii" font is to be included in a Device Table for the V-I-P (DevName value) machine, then "NCPi" must also be included. If it is not, then compose will complain if the borrowed characters are used.

In some machines, like the Mergenthaler V-I-P, that have limited font capacity, many Device Tables are likely to be needed to describe the many different configurations. Other machines, such as the Autologic APS-5, that have large font storage capacity, will usually need only one Device Table.

A Device Table section begins with a "Device" statement and ends with the beginning of another Device Table or the end of the device description input file. Global Device Values and Symbol Declarations may also appear within a Device Table section.

```
Device: <name>{, <name>}... init <initfont>, <initsize>{, <lead>}
        {<LNfont>{, <LNsize>}};
```

where:

<name>
is a name to be attached to the Device Table. This is the name referenced by the -device control argument of compose. <name>.comp_dsm is added to the output segment if it is not the primary entryname.

<initfont>
is the name of the initial Font Table to be used with the device. The named font must be valid for use with the device (see use: below).

<initsize>
is the initial pointsize to be used with the device. It must be a value in the initial Size Table for the device (see sizes: below).

<lead>
an optional value for initial leading (linespace) for the device. It must be a value in the initial Size Table for the device (see sizes: below). The default value is <initsize>.

<LNfont>
an optional font to be used for displaying line numbers when the -line_number control argument is given to compose. The named font must be valid for use with the device (see use: below). The default line number font is <initfont>.

<LNsize>
an optional pointsize to be used with <LNfont>. It must be a value in the initial Size Table for the device (see sizes: below). The default value is <initsize>.

Global/Local Device Values

All the items discussed in Global Device Values above have local counterparts. Local Device Values apply only to the Device Table in which they appear; any given are discarded when the Device Table is completed. The syntax of the Local Device Values is indentical to the corresponding Global Device Values except that the keyword tokens are spelled with all lowercase letters. These Device Values are all set to their current Global or default values when a Device Table is started.

The Global/Local Device Values statements are listed below.

```
units: <keyword>;
artproc: <name>{$<name>}>;
attach: <quoted-string>;
cleanup: <output>;
comment: <quoted-string>;
defaultmargs: <units>,<units>,<units>,<units>;
devclass: <quoted-string>;
devname: <quoted-string>;
endpage: <input>;
footproc: {<name>{$<name>}}{, <fontname>};
interleave: <switch-value>;
letterspace: <integer>;
maxpages: <integer>;
maxpagelength: <units>;
maxpagewidth: <units>;
minbotmarg: <units>;
minlead: <units>;
minspace: <units>;
mintopmarg: <units>;
outproc: <name>{$<name>};
sizes: <name>;
stream: <switch-value>;
taperec: <integer>;
```

Unique Local Device Values

The following Local Device Values have no Global Device Value counterparts.

The "dvc" (device-control) statement gives the name and specification of a device control command that may be invoked with the device-control (.dvc) control in compose. This is the mechanism used to transmit non-typographic commands to the device. The dvc statement may have either of these two forms:

```
dvc: <name> <name>{$<name>} {<quoted-string>;}
dvc: <name> <output>;
```

where:

<name>
is the name of the device control as it is to be referenced by the .dvc compose control.

<name>{$<name>}
is the entryname and optional entrypoint of a procedure that implements the device control if such a procedure is needed.

<quoted-string>
is an exec_com-like skeleton of a string to be passed to the device control procedure. Any replacable arguments ("&i") are replaced with corresponding parameters given with the .dvc compose control. Note that the full syntax of exec_com is NOT supported; only simple substitution of "&i" with "i" limited to a maximum value of 9. The double ampersand convention (&&) is used when a literal ampersand is wanted in the skeleton string.

<output>
is an <output> string to be transmitted to the device if such a string is sufficient to implement the device control.

Note: Either the procedure and optional skeleton argument or the output string must be given; both are not allowed.

(Note: This statement (and the associated .dvc compose control) will not be implemented for the first release (MR8.0) because of the lack of a specific application. The projected usage of the statement (and control) is to cause such events as film canister eject in the device and to sound alarms/flash lights to notify the operator that manual intervention is desired.)

The "use" statement includes a Font Table in the Device Table.

```
use: <fontname>{, <name>}... <output>;
```

where:

<fontname>
is the name of a previously defined Font Table to be included.

<name>
: is an additional name by which the font may be referenced for this device.

<output>
: is the character string to be transmitted to the device to cause it to switch to the font.

For example: (dtc300s)

```
Device: dtc300s init ascii, 7.2, 12;
attach: "syn_ user_output";
footproc: dtc300s_writer_$footproc;
use: ascii 033 " 0";
use: pica10, p10 033 " 6";
use: pica12, p12 033 " 5";
```

ARTWORK PART DESCRIPTIONS

The artwork parts for "diablo" terminals are plot strings made up of various spacing and the period (".") character. When contructing strings for such a terminal, they must conform to the following specifications.

> In these diagrams, the matrix of periods represents the 6x8 possible dot positions in a print position. The "+" represents the beginning and/or ending position of the print head. It is the position at which a single "." would print in normal typing mode. If there is only one "+" in a diagram, then the print head should return to the starting position. The ending position is consistently to the right of the starting position. The "o" represents a matrix position where a "." must be placed. Note that the vertical channel for lines is on the left edge of the matrix.

"One-High" Math Symbols

```
"["        "]"       "{"        "}"     "("        ")"     "¦"      "¦¦"

ooooo  ooooo           ooo  ooo           ooo  ooo         o       o  o
ʘ.....     ʘ.....    .ʘ....    o......  .ʘ....    o......  ʘ.....  o .ʘ....
ʘ.....     ʘ.....    ʘ.....     ʘ.....  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
ʘ.....     ʘ.....    ʘ.....     ʘ.....  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
ʘ.....     ʘ.....    ʘ.....     ʘ.....  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
ʘ.....     ʘ.....   o......     .ʘ....  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
ʘ.....     ʘ.....  o ......     ..ʘ...  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
ʘ.....     ʘ.....   o......     .ʘ....  ʘ.....     ʘ.....  ʘ.....  o .ʘ....
⊕.....+    ⊕.....+   ⊕.....+    ⊕.....+ ⊕.....+    ⊕.....+ ⊕.....+ o +o....+
o          o         o          o       o          o       o       o  o
o          o         o          o       o          o       o       o  o
o          o          o        o         o        o        o       o  o
ooooo  ooooo           ooo  ooo           ooo  ooo         o       o  o
```

```
  "o"          "X"        "cb"         "d"          "/"          "\"

                                                          o     o
  ......       ......     ......       ......       .....ʘ       ʘ.....
  ......     o ....ʘ.     ʘʘ....       ......       ....ʘ.       .ʘ....
  ......      o...ʘ..     ʘʘ....       ......       ...ʘ..       ..ʘ...
  ......       ʘ.ʘ...     ʘʘ....      o...ʘ..       ..ʘ...       ...ʘ..
 oʘʘ....       .ʘ....     ʘʘ....       ʘ.ʘ...       .ʘ....       ....ʘ.
ooʘʘʘ...       ʘ.ʘ...     ʘʘ....     ooʘʘʘʘʘ.       ʘ.....       .....ʘ
ooʘʘʘ...      o...ʘ..     ʘʘ....       ʘ.ʘ...      o......       ......o
 o⊕ʘ....+    o +...ʘ.+    ⊕ʘ....+     o+..ʘ..+    o +.....+      +.....+o
                          oo
                          oo
```

```
                          "cm"             "tm"

                          ............     ............
                          ..ʘʘʘ.......     ............
                          ʘʘ...ʘʘ.....   ooʘʘʘʘʘ.......
                         o.......ʘ....     .ʘ..ʘ.....ʘ.
                        o ........ʘ...     .ʘ..ʘʘ...ʘʘ.
                        o ........ʘ...     .ʘ..ʘ.ʘ.ʘ.ʘ.
                        o ...*....ʘ...     .ʘ..ʘ..ʘ..ʘ.
                         o+......ʘ....+    +ʘ..ʘ.....ʘ.+
                          oo   oo
                            ooo
```

NOTE: The "*" in the "cm" diagram represents the position of the "c" for "copyright". It may be changed to any other single letter of the users choice.

```
   "^"          "v"        "<-"         "->"

   ......       ......     ......       ......
   o.....       ......     ......       ......
  ooo....       ......     ......       ......
 ooooo...       ......     ......       ......
ooooooo..       ......     ......       ......
   ......       ......     ......       ......
   ......    ooooooo..     ...o..    o  ......
   +.....+    oo●oo...+    +.oo..+   oo X.....
               ooo          ooo      ooo
                o          oooo      oooo
                            ooo      ooo
                             oo      oo
                              o      o
```

Large Artwork Elements

```
"D^"         "Dv"          "D<"          "D>"

......       ......        ......        ......
......       o.....o       ......o       o.....
......       .o...o        .....o        .o....
......       ..o.o.        ....o.        ..o...
......       ...o..        ...o..        ...o..
......       ......        ..o...        ....o.
...o..       ......        ...o..        ...o..
+.o.o.+      +.....+       +...o.+       +.o...+
 o   o                          o         o
o     o                          o       o
```

```
    "Clf"            "Crt"

    ......           ......
    ...oooo     oooo ......
    ooo...          ooo....
   o......           ..o...
  o ......           ...o..
 o  ......           ....o.
 o  ......           ....o.
o   +.....+          +....o+
o                         o
o                         o
o                         o
o                         o
 o                       o
 o                       o
  o                     o
   o                   o
    ooo             ooo
       oooo     oooo
```

```
                  "-rul"       "¦rul"     "/rul"      "\rul"

                  ......       ......     ......      ......
                  ......       O.....     ......      ......
                  ......       O.....     ......O    O......
                  ......       O.....     .....O      O.....
                  ......       O.....     ....O.      .O....
                  ......       O.....     ...O..      ..O...
                  ......       O.....     ..O...      ...O..
                  ⊕OOOOO+      ⊕.....+    +O....+     +...O.+
                                          O                O
```

Math Symbol Top Parts

```
"[tp"      "]tp"     "{tp"      "}tp"   "lptp"     "rptp"  "¦tp"     "¦¦tp"

OOOOO  OOOOO           OOO  OOO           OOO  OOO         O       O  O
O.....     O.....    .O....    O......  .O....    O......  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
⊕.....+    ⊕.....+   ⊕.....+    ⊕.....+ ⊕.....+    ⊕.....+ ⊕.....+ O +O....+
```

Math Symbol Half-top Parts

```
"[ht"      "]ht"     "{ht"      "}ht"   "lpht"     "rpht"  "¦ht"     "¦¦ht"

......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
OOOOO. OOOOO.....    ..OOO. OOO ......  ..OOO. OOO ......  O.....  O .O....
O.....     O.....    .O....    O......  .O....    O......  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
⊕.....+    ⊕.....+   ⊕.....+    ⊕.....+ ⊕.....+    ⊕.....+ ⊕.....+ O +O....+
```

Math Symbol Middle Parts

```
"[md"      "]md"     "{md"      "}md"   "lpmd"     "rpmd"  "¦md"     "¦¦md"

O          O         O          O       O          O       O       O  O
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....   O......     .O....  O.....     O.....  O.....  O .O....
O.....     O.....  O ......     ..O...  O.....     O.....  O.....  O .O....
O.....     O.....   O......     .O....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
O.....     O.....    O.....     O.....  O.....     O.....  O.....  O .O....
⊕.....+    ⊕.....+   ⊕.....+    ⊕.....+ ⊕.....+    ⊕.....+ ⊕.....+ O +O....+
```

Math Symbol Half-bottom Parts

```
"[hb"     "]hb"     "{hb"      "}hb"   "lphb"     "rphb"  "¦hb"     "¦¦hb"

......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
......     ......    ......     ......  ......     ......  ......    ......
⊕.....     ⊕.....    ⊕.....     ⊕.....  ⊕.....     ⊕.....  ⊕.....  o +o....
o          o         o          o       o          o       o       o  o
o          o         o          o       o          o       o       o  o
o          o          o        o         o        o        o       o  o
ooooo  ooooo           ooo  ooo           ooo  ooo         o       o  o
```

Math Symbol Bottom Parts

```
"[bt"     "]bt"     "{bt"      "}bt"   "lpbt"     "rpbt"  "¦bt"     "¦¦bt"

o          o         o          o       o          o       o       o  o
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    .o....    o......  .o....    o......  o.....  o .o....
⊕oooo.+oooo⊕.....+   +.ooo.+ooo +.....+ +.ooo.+ooo +.....+ ⊕.....+ o +o....
```

Math Symbol Filler Parts

```
"[fl"     "]fl"     "{fl"      "}fl"   "lpfl"     "rpfl"  "¦fl"     "¦¦fl"

o          o         o          o       o          o       o       o  o
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
o.....     o.....    o.....     o.....  o.....     o.....  o.....  o .o....
⊕.....     ⊕.....    ⊕.....     ⊕.....  ⊕.....     ⊕.....  ⊕.....  o +o....
```

This is the include file describing the output pieces from compdv:

```
/*         BEGIN INCLUDE FILE - comp_dvt.incl.pl1 - 03/03/78 J Falksen        */

/*                                                            . */
/* .                                        |           |     . */
/* .   _      _             _             _ |          -|-    . */
/* .  / \    / \   |/|/|  |/ \           / \|  \   /    |     . */
/* . (      (   )  | | |  |   )         (   |   \ /     |     . */
/* .  \_/    \_/   | | |  |\_/           \_/|    V      \_    . */
/* .                      |      -----                        . */
/* .                      |                                   . */

dcl  comp_dvt_version fixed bin int static init (8); /* cur version     */

/* All names which end in "_r"are offset values within the device table  */
/*  segment.                                                              */

dcl 1 comp_dvt based (shared.devptr),
    2 version fixed bin,               /* version of this structure       */
    2 devname char (32),               /* name of device, unique as to    */
                                       /*  options, etc.                  */
    2 devclass char (32),              /* what general kind of device is  */
                                       /*  this, currently available:     */
                                       /* "typewriter" (line printer, too) */
                                       /* "braille"                       */
                                       /* "photocomp"                     */
    2 atd_r bit (18) aligned,          /* attach description for on-line  */
                                       /*  output                         */
    2 outproc entry                    /* page output processor           */
    (fixed bin,                        /*   function - 0=build            */
                                       /*              1=init page        */
                                       /*              2=init file        */
                                       /*              3=cleanup          */
     fixed bin (35)),                  /*   error code                    */

    2 footproc entry                   /* footnote procerence proc        */
    (char (*) varying,                 /*   reference string (IN)         */
     ptr,                              /*   comp_dvt_p (IN)               */
     fixed bin (31),                   /*   added footref width (OUT)     */
     fixed bin,                        /*   footref font (IN)             */
     fixed bin)                        /*   current font (IN)             */
     returns (char (*) varying),       /*   processed reference           */

    2 artproc entry (),                /* artwork proc                    */
                                       /* dont know how to describe yet   */

/* following values are in millipoints                                    */
    2 inl_min fixed bin (31),          /* min .in delta                   */
    2 min_lead fixed bin (31),         /* minimun lead                    */
    2 vmt_min fixed bin (31),          /* min usable .vmt                 */
    2 vmb_min fixed bin (31),          /* min usable .vmb                 */
    2 def_vmt fixed bin (31),          /* default .vmt                    */
    2 def_vmh fixed bin (31),          /* default .vmh                    */
    2 def_vmf fixed bin (31),          /* default .vmf                    */
    2 def_vmb fixed bin (31),          /* default .vmb                    */
    2 pdw_max fixed bin (31),          /* max page width available        */
    2 pdl_max fixed bin (31),          /* max page length available,      */
                                       /*   (-1 = unlimited)              */
    2 upshift fixed bin (31),           /* footnote reference shift         */
    2 init_ps fixed bin (31),           /* initial pointsize (millipoints)  */
    2 min_spb fixed bin,                /* min spaceband                    */
    2 avg_spb fixed bin,                /* nominal spaceband                */
    2 max_spb fixed bin,                /* max spaceband                    */
    2 lettersp fixed bin (31),          /* max letterspacing                */

    2 max_pages fixed bin,              /* max pages in output "unit"       */
                                        /*   (0 = unlimited)                */
    2 init_font fixed bin aligned,      /* initial font to use              */
    2 dvc_r bit (18) aligned,           /* device control table relptr      */
    2 fcs_r bit (18) aligned,           /* font change table relptr         */
    2 comment_r bit (18) aligned,       /* comment string relptr            */
    2 cleanup_r bit (18) aligned,       /* "cleanup" string relptr          */
    2 foot_font fixed bin aligned,      /* font index for footnote reference */
                                        /*   if one was specified           */
    2 rel_units fixed bin,              /* number of relative units an EM   */
                                        /*   is divided into                */
    2 sws unaligned,
      3 space_fill bit (1),             /* unused                           */
      3 space_fill_fast bit (1),        /* unused                           */
      3 interleave bit (1),             /* 0- page block has lines in column */
                                        /*   order left-to-right            */
                                        /* 1- page block has lines in line  */
                                        /*   order top-to-bottom            */
      3 drop_space bit (1),             /* unused                           */
      3 no_adjust bit (1),              /* 1- do not adjust lines           */
      3 mbz bit (22),
      3 endpage bit (9),                /* EOP char if not "0"b             */

    2 open_mode fixed bin (35),         /* when going to a file             */
    2 recleng fixed bin,                /* length of tape records           */
    2 font_ct fixed bin,                /* # fonts present                  */
    2 fonts (comp_dvt.font_ct),         /* fonts defined for this device    */
      3 font_r bit (18) unal,           /* font table relptr                */
      3 fcs_r bit (18) unal,            /* fcs string relptr                */
      3 size_r bit (18) aligned,        /* point size list                  */
      3 name char (32);                 /* font reference name              */

/* The usage formula for units:                                            */
/*                                                                         */
/*    rel_units * length_in_points                                         */
/*    ----------------------------  = length_in_units                      */
/*    points_per_EM                                                        */

/*        END INCLUDE FILE comp_dvt.incl.pl1                               */

/*        BEGIN INCLUDE FILE comp_dvc.incl.pl1                             */

/* .                            . */
/* .     |                      . */
/* .  __ |              __      . */
/* . /  \|  \   /      /  \     . */
/* .(    |   \ /      (         . */
/* . \__/|    V        \__/     . */
/* .                            . */
/* .                            . */
dcl  dvc_p ptr;
dcl 1 dvc based (dvc_p),                /* table relating dvc names         */
```

```
                                         /* with processing routines         */
    2 dvc_ct fixed bin,                  /* number of dvc's specified        */
    2 fill fixed bin,
    2 e (dvc.dvc_ct),
      3 id char (8),                     /* the name of the dvc              */
      3 str_r bit (18) aligned,          /* offset of macro string           */
      3 fill bit (36),
      3 rtn entry (                      /* dvc action routine               */
     char (*),                           /* call string                      */
     char (*),                           /* macro string (if any)            */
     fixed bin (35))                     /* error code                       */
     returns (char (1020) varying);      /* result string                    */
                                         /* .dvc request is replaced by      */
                                         /* result string                    */

/*       END INCLUDE FILE comp_dvc.incl.pl1                                  */

/*       BEGIN INCLUDE FILE comp_fcs.incl.pl1                                */

/* .                        . */
/* .   /\                   . */
/* .   |                    . */
/* .   |      /\     /---   . */
/* .  -|-    (       \_/\   . */
/* .   |      \_/     __/   . */
/* .                        . */
/* .                        . */
/* A font-change string is an output string used by the output module        */
/*  whenever it discovers that a font change is needed.                      */
dcl 1 fcs based (shared.fcs_table_ptr), /* font-change string table          */
    2 fcs_ct fixed bin,                 /* number of strings present         */
    2 e (fcs_ct),
      3 font_r bit (18) aligned,        /* font table                        */
      3 ref_r bit (18) aligned;         /* string to change to this font     */

/*       END INCLUDE FILE comp_fcs.incl.pl1                                  */

/*       BEGIN INCLUDE FILE comp_font.incl.pl1                               */


/* .                             . */
/* .   /\                 |      . */
/* .   |                 -|-     . */
/* .   |      /\    |/\   |      . */
/* .  -|-    (  )   |  |  |      . */
/* .   |      \_/   |  |  \_     . */
/* .                             . */

dcl 1 font based (shared.font_ptr),
    2 parts_r bit (18) aligned,         /* reserved for future use           */
    2 sws,
      3 punct_space bit (1) unal,       /* add EN after punctuation          */
      3 mbz bit (35) unal,
    2 data_ct fixed bin,                /* characters in the font            */
    2 data (0:font.data_ct),
      3 units fixed bin (31),           /* width in units                    */
      3 which fixed bin,                /* index into fcs array              */
      3 what_r bit (18) aligned;        /* output string                     */
/*        END INCLUDE FILE comp_font.incl.pl1                          */

/*        BEGIN INCLUDE FILE comp_other_ps.incl.pl1                    */


/* .                                      . */
/* .         o                   ‾|       . */
/* .                              |       . */
/* . /‾‾‾‾   ‾|     ‾‾‾7   /‾\    |       . */
/* . \ /‾\    |       /   ( /     |       . */
/* .  ‾  /    |      /     \‾/    |       . */
/* .  ‾‾‾    _|_    /___    ‾    _|_      . */
/* .                                      . */
dcl  sizel_p ptr;
dcl 1 sizel based (sizel_p),           /* list of allowed point sizes       */
    2 val_ct fixed bin,
    2 val (sizel.val_ct) fixed bin (31);

/* .                                      . */
/* .                          |           . */
/* .                         -|-          . */
/* . |/‾\   ‾‾‾\  |/‾\        |    /‾‾‾   . */
/* . |   )  /‾‾|  |           |    \ /‾\  . */
/* . |\ /   \ /|  |           \_    ‾  /  . */
/* . | ‾     ‾‾                    ‾‾‾    . */
/* . |                                    . */
dcl  parts_p ptr;                      /* parts with which to build         */
                                       /*  horizontal and vertical artwork  */
                                       /*  parentheses, brackets and braces */
dcl 1 parts based (parts_p),
    2 ranges fixed bin,                /* number of ranges                  */
    2 r (parts.ranges),
      3 min_hi fixed bin,              /* minimum height                    */
      3 max_hi fixed bin,              /* maximum height                    */

      3 e (0:11),
        4 x fixed bin,                 /* width                             */
        4 y fixed bin,                 /* height                            */
        4 c char (1),                  /* ASCII selector                    */
        4 fill char (3);


/* .                               . */
/* . |             |               . */
/* . |            -|-              . */
/* . |/‾\  /‾‾‾    |    |/‾\       . */
/* . |   ) \ /‾\   |    |          . */
/* . |\ /   ‾  /   \_   |          . */
/* .   ‾    ‾‾‾                    . */
/* .                               . */
dcl  str_p ptr;
dcl 1 bstr based (str_p),
    2 str_l fixed bin,
    2 str char (str_l);                /* string definition                 */
                                       /* this template is used to          */
                                       /*  reference fields which have      */
                                       /*  these offsets:                   */
                                       /* comp_dvt.atd_r    dvc.str_r       */
                                       /* fcs.ref_r         font.what_r     */
/* END INCLUDE FILE ..... comp_other_ps.incl.pl1 .....                      */
```