

File Converter

| | |
|--|----|
| File Converter | 3 |
| Preliminary Recommendations | 4 |
| Converting a Label Format | 4 |
| Retrieving Variable Data | 7 |
| Special Features of the Converter | 8 |
| Source Printer | 8 |
| Object Overlay | 8 |
| Font Width | 8 |
| Formula Variables | 9 |
| Type of Variables | 10 |
| Name of Variable Objects | 10 |
| 2D Bar Codes | 10 |
| Bar Codes | 11 |
| Fonts | 11 |
| Windows™ Fonts | 12 |
| Figures | 12 |
| Printer Settings | 13 |
| Label Format | 13 |
| Dimensions and Positioning of Objects | 13 |
| Images | 13 |
| Appendices | 14 |
| Import of Monarch MPCLII Command Files | 14 |
| Import of UBI LabelShop 2.0 Description Files | 15 |
| Import of Zebra ZPLII Command Files | 15 |
| Import of Datamax Command Files | 15 |

Appendix

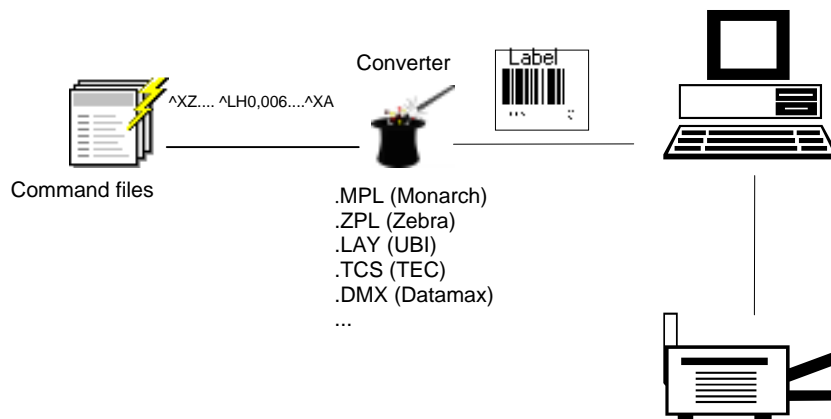
| | |
|--|----|
| Import of TEC TECSCRIPT Command Files | 16 |
| Import of Eltron EPLII Command Files | 16 |
| Import of INTERMEC IPL command files | 16 |
| Import of SATO SPL Command Files | 17 |
| Import of Avery-Novexx Easy Plug Command Files | 17 |
| Import of Printronix IGP/PGL Command Files | 17 |
| Importing Description Files from Datamax's PcBatch Program | 18 |
| Importing Printronix IGP/VGL Command Files | 18 |
| Error Messages | 19 |
| "Memory Problem" | 19 |
| "Empty File" | 19 |
| "DII Problem" | 19 |

File Converter

The file converter translates labels which are described in printer language or generated by other applications into labels formatted in accordance with the label program.

One very attractive feature of the converter is its ability to recreate all of the objects that make up the label background (text, bar codes, figures and images).

However, because of the diversity of printer languages, it is not always possible to accurately recreate certain details such as font style, printer settings, and so on.



Warning: If you have a LITE version of the software, your distributor can supply you with the converter. However, your program will be subject to certain restrictions that do not apply to the full version. Restrictions will be pointed out throughout the user documentation.

Appendix

Preliminary Recommendations

Printer commands must be in text files whose extensions match the types of printers for which they were created (.zpl, .mpl, etc.).

There are no limits on the size of printer command files.


Generally speaking, it is strongly recommended that the original label be recognized by the printer model for which it was created. If it is not, we cannot fully guarantee the quality of the label recreation.

Moreover, users are advised to avoid inserting comments into label files, even if the printer language allows them to.

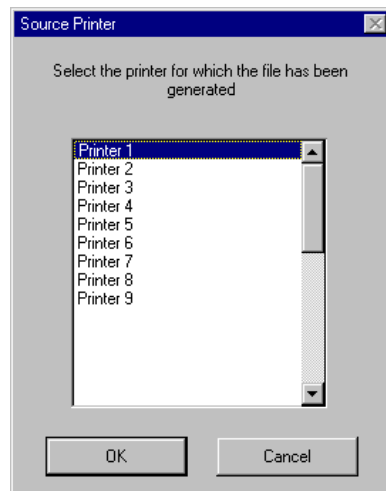
Converting a Label Format

Before Conversion

- Start up the label application.
- Using the **Printer** dialogue box, select the printer model closest to the one for which the label to be converted was created.
- Select the **Convert** command from the **File** menu.
- Select the required file format from the scrolling **File List**.
- Select the command file to be converted, then confirm your choice.

 The **Source Printer** dialogue box appears on the screen.

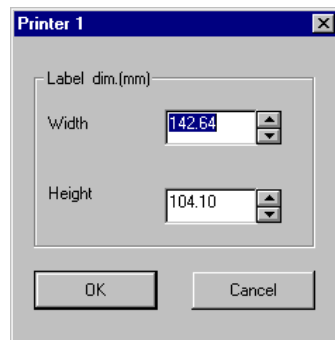
Converter



- Select the printer model closest to the one for which the label was created from the above list, then confirm your choice.

↪ The label appears on the screen.

Some printer languages do not supply the dimensions of the label format. In this case, a dialogue box for input of the label dimensions appears on the screen.



- Input the missing dimensions of the label format, then confirm.

↪ The label appears on the screen.

Appendix

After Conversion

- Select the **Save Under** command to store the label in the label program's format.
- Using the **Printer** dialogue box, select the destination printer.

Printer changes can create error messages, such as problems with margins, cropping, and so forth. If this happens, you will have to adjust the values from the **Format** dialogue box of the **File** menu.

Retrieving Variable Data

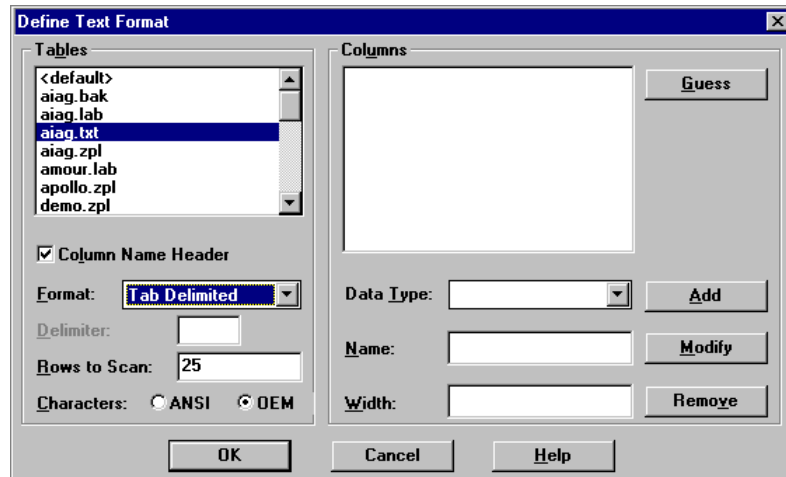
If your command file contains variable data and if the converter allows (see **Appendices** p. 14), the variable data will be retrieved in a text file.

The file will share the name of the command file associated with it. Only the extension, i.e., “.txt”, will be different.

The file will be located in the same directory as the command file. To retrieve the data extracted from the command file into the label, install the ODBC text driver (see **Chapter Installing ODBC Data Sources** of the reference manual).

The ODBC driver will have to be set as follows:

- ♦ extension “.txt”;
- ♦ the TAB character as the delimiter;
- ♦ OEM format characters;
- ♦ file set to feature a heading line for column names.



Special Features of the Converter

Source Printer

When you request retrieval of a label, the converter offers you a list of source printers. Choosing one is mandatory and determines how faithfully the label will be recreated. In addition, choosing a source printer retrieves the printer's parameters.

If you are not familiar with the printer model or if you selected the wrong one, the converted label may not recreate the original.

It is therefore recommended that users make a careful inventory of the labels they have in each printer before beginning a conversion process.

Object Overlay

It is not always possible to preserve the order in which objects were overlaid in the original label.

For example, if the label contains a reverse video text simulated by a white text against a black background, the black area may end up above the text.

This can be solved by moving the black rectangle below the text after conversion. The **Above/Below** commands can be accessed from the **Object** menu.

Font Width

The width of the vector fonts may need adjusting. Indeed, setting font widths requires doing calculations that employ different units, which may result in different interpretations.

Font width expressed as a percentage will have to be verified from the **Text** dialogue box of the **Object** menu.

Formula Variables

Not applicable to the LITE version.

Formula variables do not exist in the LITE version. If your command file contains a formula variable, the converter creates a non-valid object on which the word «#ERROR» is displayed.

Printer languages can sometimes be used to create **Formula** variables.

However, users will have to check the validity of the formulas after conversion, because most languages use proprietary techniques to establish formula variables.



...for experienced users

There are two ways **Formula** variables can be retrieved in printer language:

- One is for the converter to translate several written instructions into printer language in a single formula incorporated into the label program. For example, several printer language instructions needed to calculate a modulo 10 are translated into just one function.
- Another is for the printer language instruction to be translated into several formulas in the label program.

It is up to the user to optimize the formulas, if desired, using functions incorporated into the label program.

Appendix

Type of Variables

The variable type of the printer languages is generally left unchanged.

For instance, a **Counter** printer variable will be translated by an actual counter.

However, users must still check to see whether they have the type of variable they need (for instance, a field from an ODBC base translated by an **Import** field, and so on).

Indeed, converters do not necessarily know the source variable.

Name of Variable Objects

Variable objects are assigned names which are selected arbitrarily. Unless the language specifies a name, the converter generates a numbered, incremental name field.

For example:

- ♦ **Formula** variables are renamed "Formula1," "Formula2," etc.;
- ♦ **Import** variables are renamed "Import1," "Import2," etc.;
- ♦ **Counter** variables are renamed "Counter1," "Counter2," etc.;
- ♦ **Date** variables are named "Date1," "Date2," etc.

Reminder: **Formula** variables are not recognized by the LITE version.

2D Bar Codes

If your label contains 2D bar codes (see **2D Bar Codes** appendix), you must verify their parameters.

Indeed, most languages have special 2D bar code parameters, which are not always standard.

Bar Codes

Certain bar codes, which are processed by the printer language, are not yet recognized by the software. As far as possible, the converter replaces them with graphic image bar codes, or, if there is no matching graphic image, by a 39 code.

Users can change the bar code type if they do not like the code substituted.

Fonts

The converter needs certain minimal information to recreate the source font as closely as possible. Fonts are identified by triplets containing the height, width, and name (visibility code) of the font.

The label program may fail to recognize a font described in the command file.

There are two possible reasons for this:

- ◆ the converter has encountered an unknown font without descriptive information. It arbitrarily replaces the unknown font with any font it recognizes. When this happens, it is very likely that the recreated text will differ from the original, and it is up to users to change the font style if they want to reproduce the original label accurately.
- ◆ the converter has encountered an unknown font with descriptive information. The font is replaced by a known font that shares the features of the font to be recreated. This greatly reduces the risk of differences between the original and the recreation.

Appendix



...for experienced users

This tip will be helpful to users who are extremely familiar with their printer and its language.

To accurately recreate the font of the original label, define the original label in the label program.

To do this, users must have the triplet identifying the font (not all printer languages provide you with this information), and must define the font from the **Fonts** section of the printer's **Settings** box (refer to the appendix for your printer).

Windows[™] Fonts

Most software programs that create labels in printer language replace **Windows[™]** fonts with logos (bitmap images).

After conversion, the logos will be recreated on the label, but will not be able to be manipulated like text.

Using the original label, change the **Windows[™]** fonts into printer fonts before converting them. After conversion, users may, if desired, change the recreated fonts back into **Windows[™]** fonts.

Figures

Some printer languages support drawing commands (circles, polygons, and so forth) for which the software program has no equivalent. In that case, the converter simulates them using images.

Printer Settings

If you want to retrieve the setting commands contained in a label file, first select the **Load Printer Settings** and **Load Printer and Port** options, by initiating the **Option** command (**Document** section) from the **Tools** menu.

All printer settings except printer speed are retrieved. However, if the label to be recreated is missing parameters, the current or default parameters of the installed printer will be used. It is therefore recommended that you check them.

Reminder: the LITE version automatically sets the printer settings.

Label Format

The software program needs the label format. If the printer language does not define the two dimensions (width and height) of the label to be recreated, the program displays a dialogue box for input of the missing value. The value begins with the data inputted previously for another label. Once the format has been retrieved, the converter generates a special format whose name and type are the same as the language mark. It is up to users to save their format if they want to keep a record of it.

Save the format from the **Format** dialogue box (see Chapter **Format**).

Dimensions and Positioning of Objects

It is not always possible to reproduce the dimensions and positioning of objects. Indeed, objects must be calculated using different units, which, depending on the source printer language, can produce different interpretations.

It is up to users to verify the dimensions and positioning of objects, using the dialogue box for each one.

Images

Images are usually retrieved. However, certain types are not (see **Appendices** p. 14).

Appendices

Import of Monarch MPCLII Command Files

The converter imports command files in Monarch MPCLII format (Monarch Printer Language II).

The usual extension of the files is “.MPL.”

They can be generated using the Monarch 2095, Monarch PLATFORM, and Monarch Quick-Set programs, or manually by users who know MPCLII language.

- a) You can insert the following into your command files:
- a single control packet,;
 - several graphic image packets;
 - a single format packet—the only one really necessary;
 - several check digit packets;
 - several batch packets.

But the font download packets and the optional data inputted are lost.

- b) Make sure that the sequencing of the packets and fields matches the sequencing of MPCLII language. Only the batch packet(s) may follow the format packet; all others must precede it.
- c) Try to keep the number of options per field to a minimum, and even to one if possible.
Most available MPCLII options (R1, R2...R51, etc.) are replaced with formulas. It is up to users to make sure that the formulas are coherent.
- d) The product does not incorporate the ”Code 16K” and “Maxicode” codes, and replaces them with a “PDF417” and “Code 39” respectively.

Import of UBI LabelShop 2.0 Description Files

The converter imports description files in UBI's UBI LabelShop 2.0 format.

Their usual extension is ".LAY."

- a) If the converter detects a conversion error, it isolates the error in a file. The file is stored in the same place as the original file, and is assigned the same name, but ends with the extension ".ERR."
- b) the batch files ".TBL" are translated into text format batch files (see the users' manual).

Import of Zebra ZPLII Command Files

The converter imports Zebra ZPLII format command files (Zebra Programming Language II).

Their usual extension is ".ZPL."

- a) The "Code 49" code is replaced by a "PDF417" code.
- b) The converter recognizes both methods for defining variables (use of "FN" and "FV" instructions).
- c) The variable data are not retrieved.

Import of Datamax Command Files

The converter imports command files in Datamax format.

Their usual extension is ".DPL."

- a) The "TELEPEN" code is replaced by a Code 128; the "FIM" code is replaced by a Code 93.
- b) The 8-bit IMG and 8-bit BMP image formats are not recognized.
- c) The variable data are not retrieved.
- d) The font download commands are not incorporated.

Appendix

Import of TEC TECSCRIPT Command Files

The converter imports TEC TECSCRIPT command files.

Their usual extension is “.TCS.”

- a) All **Text** and **Bar Code** objects are translated into **Import** variables.
- b) Since there are many versions of the TECSCRIPT language, the converter may not recognize some instructions. When the converter encounters an unknown instruction, the program displays a dialogue box indicating the nature and location of the error (see **Error Messages** p.17).

Import of Eltron EPLII Command Files

The converter imports command files in Eltron EPLII format (Eltron Printer Language II).

The usual extension of the files is “.EPL.”

Import of INTERMEC IPL command files

The converter imports file commands in IPL (Intermec Programming Language) format.

The usual extension of these files is «.IPL.»

- a) The bar codes «Code 49,» «Code One,» and «Code 16K» are replaced by a PDF417 code.
- b) Resolution modification commands are not yet included.
- c) The bar code extensions +2 and +5 are not included.
- d) The printer counter fields are retrieved as simple variables, not as counters.

Import of SATO SPL Command Files

The converter imports command files in SATO SPL (Sato Programming Language) format.

The usual extension of the files is “.SPL.”

Import of Avery-Novexx Easy Plug Command Files

The converter imports command files in the Easy Plug format of Avery-Novexx.

The usual extension of the files is “.PLG.”

Import of Printronix IGP/PGL Command Files

The converter imports command files in Printronix IGP/PGL (Printronix Graphics Language) format.

The usual extension of the files is “.PGL.”

- a) The various work units (SCALE DOT and SCALE CHAR) are preserved.
- b) All graphics are retrieved, regardless of format (PGL or PCX), except for those in TIFF format.

Comments & Advice: If the label to be converted comes from a matrix printer, retrieving certain objects can create slightly different interpretations, notably for:

- the retrieval of ratios and fine bars in bar codes when the command file refers to predefined values such as “X1,” ”X2,” etc.); and
- the ranking of certain fonts.

The changes are due to the lower resolution of matrix printers.

To optimize retrieval, we advise you to test the command files on different types of printers (thermal, matrix and laser) and save the best output.

Appendix

Importing Description Files from Datamax's PcBatch Program

The converter imports description files saved in Datamax's PcBatch format.

The files' usual extension is ".FDM".

Unlike other converters:

- ♦ the source printer is associated with the description file and users are therefore not asked to supply it.
- ♦ because the unit of measurement is not associated with the description file, users are asked to provide it.

Importing Printronix IGP/VGL Command Files

The converter imports command files saved in the IGP/VGL format (Code V Graphics Language) of Printronix.

The files' usual extension is ".VGL".

Note:

If the label to be converted comes from a matrix printer, the recovery of certain objects can generate slightly different interpretations, notably with respect to:

- the positioning of fonts; and
- the positioning of bar codes.

The problem is caused by the low resolution of this type of printer.

Error Messages

Error messages may appear while command files are being converted.

The messages tell users what and where the errors are so that they can be corrected.

Error messages consist of one of the messages discussed below, followed by a line number.

“Memory Problem”

The application has insufficient memory to complete the conversion.

Users will have to close other applications they may have open.

“Empty File”

The converter has detected an empty command file.

“Dll Problem”

The converter cannot find one of its components.

This is an installation problem—contact your distributor.