

[Install/Demo]

ShadeParagraphPCL

Introduction

This macro will goes a long way to correcting one of the major oversights in Word for Windows 1.x: the shading of paragraphs.

It is far from perfect. But it seems to work for the more common sorts of paragraph shading.

This version adds a manual mode. which will allow you to print a test page, measure in inches where you want a shaded box, and enter the measurements directly. This may be necessary if you are using soft fonts or one of the font scaling programs (Adobe Type Manger or FaceLift).

Given the way this macro addresses the printer, it may not work at all with SuperPrint. I have not tested it with that program.

What it does

ShadeParagraphPCL works by inserting a {print } field at the beginning of the paragraph to be shaded. This print field is extremely complex, and uses PCL commands to 1) save the current cursor postion, 2) tell the printer where to begin shading, 3) tell the printer the dimensions to be shaded, 4) tell the printer what density of shading or style of hatching, and finally 5) return the cursor to the saved cursor position.

How it does what it does

This is the trickiest part. ShadeParagraphPCL repaginates your document so that it can determine *the current top of the current page*. It then, literally, steps from the top of the page to the current paragraph and calculates where on the page the paragraph begins. It calculates the paragraph width. All of these calculations are completely dependent on the current margin settings as they are found in FormatDocument

What what it does looks like

~~§This is a paragraph that is shaded. To see what the print code looks like (so you know what to remove if you don't want the shaded paragraph, or if you add text before this page, you can find it and remove it), turn on ViewFieldCodes (or ShowAll). You'll see the print field at the beginning~~

of the paragraph.

Requirements:

ShadeParagraphPCL has been tested on a Hewlett-Packard Laserjet III. Since it does not use the extended commands of the PCL/2 language, it *should* work with earlier models.

Note: this documentation file is formatted with the internal CG Times (WN) font found on the HP Laserjet III. If you are printing on an HP Laserjet II (or are using a cartridge), please reformat Normal style to use the internal font. If you are using Adobe Type Manager and have Tms Rmn or CG Times aliased to call the ATM Times font, there will be problems. More on ATM later in the documentation. ShadeParagraphPCL works with *some* soft fonts... but I cannot test will all possible variations. How to tweak the print field will also be covered later in the documentation.

Further note: this macro was tested using Word 1.1a and the HP Laserjet driver numbered 3.77. In the mysterious world of Word for Windows printing there's no telling what variable might not bring things to a screeching halt. If this doesn't correctly print on your system you might want to try the latest HP driver before resorting to manual mode.

Usage

The dialog box

When you run ShadeParagraphPCL you will be presented with the following dialog box

Shading Type

The ComboBox visible in this example allows to select the Percent Shading for the rectangle.

You can type in a value manuall (for instance, 15%)

Display Toggle

Display Patterns

Cross-Hatched

If you check on **Display Patterns** and then click on **OK**, the dialog box will redraw, displaying a ListBox of Hatch patterns (and **Display Patterns** will change to **Display Shades**:

Area

Paragraph

This option will create a print field to shade the current paragraph

If you have more than one paragraph selected when you run ShadeParagraphPCL, only the first paragraph will be shaded.

Tip: If you want the *appearance* of multiple paragraphs, you can achieve this by using NewLine charactrs (instead of carriage returns).

Page

This option will shade the entire area of the page as defined by the margins found in Format Document.

There is currently no option to place a border around the entire page (I'm working on it).

Page Tip

You could place the Shade.Page print field in the header on the first page of your document if you wanted every page of a multi-page document to be shaded.

Format Border

Checking this option (and then OK) will run FormatParagraph so that you can change the border type for the current paragraph.

Options

Best Guess

This is the automatic mode for ShadeParagraphPCL. It calculates the print field measurements by summing the lines, line spacing, space before and after, etc of all the paragraphs preceeding the paragraph to be shaded.

Enter measurements

This option, new in version 1.3, allows you to enter the starting points (top

and left), and dimensions (height and width) of the shading rectangle. Make all measurements from the physical edge of the paper (top and left). ShadeParagraphPCL will automatically compensate for the unprintable area of the Laserjet printers.

Note: if you are measuring a boxed paragraph, know that ShadeParagraphPCL automatically increases the width of the box slightly (so that the shading does not begin and end **on** the character, but 2.1 points to the left and right). So measure from slightly within the left of the border.

The PCL language ignores decimal places beyond hundredths.

If you have Enter measurements selected, and have not entered any values, you will be returned to the dialog box with an error.

If you return to the dialog box after printing a test page, the current values will be preserved.

Remove / Confirm

This option will remove all {print } fields from the current document. If you have Confirm checked then you will be prompted before each removal.

Print test page and return

This option is useful if you wish to print a page, measure it, then enter the values manually. and wish to print the result.

Returning to the dialog box

Note: by default each of the functions listed on the dialog box execute and then return to the dialog box for further instructions.

To change this, edit the macro and replace the appropriate occurrences of Goto Again with Goto Bye. (The one you will not wish to change is the one associated with dlg.Toggle).

Cautions, limitations

Fonts

ShadeParagraphPCL has been tested on a Laserjet III, with ATM installed and NO soft-fonts.

I have found, for reasons yet determined, that ATM fonts can cause lots of problems.

I would recommend that you use the internal fonts, **CG Times** and **Univers** and **Courier** for documents containing shaded paragraphs.

LineSpacing

ShadeParagraphPCL works best with linespacing set to Auto. It will, however, do a pretty fair job with both 1li and -1li and -1.5" type measurements.

But if the paragraph to be shaded can be printed in a single page document, simply leave the line spacing set to Auto and do everything in single spacing...

Pictures

Make sure that there are no pictures on the current page, otherwise the calculations will be substantially off.

Borders

The fewer bordered paragraphs on the current page, before the paragraph to be shaded, the better.

Adobe Type Manager

I have tested ShadeParagraphPCL with several ATM fonts and it works with most. However, there are two problems.

Firstly, there will be some deviation from the result achieved with the internal fonts. This is because ATM (and other soft fonts) have different internal "leading". The leading assumed by ShadeParagraphPCL is 20% of the pointsize. (You will find this value listed at the top of the source code as Percent=.2). It seems that ATM does better if this is increased to .21. You can experiment to see what works best with your configuration.

The second problem is a bug in Word for Windows. If you have a document that contains both soft fonts, or ATM fonts, and hard fonts, and {print } fields, weird things begin to happen.

The clearest way to illustrate this is as follows: if you have a document that is entirely ATM fonts, and a boxed paragraph, and a ShadeParagraphPCL print field for that boxed paragraph, what you will get is: 1) a page printed that contains the text and the shading, and another page that contains the box...

So, the warning is: if you do use ATM, don't use borders.

The {Print } field, a dissection

This is somewhat technical, and, I hope, can be skipped. But if you are curious to know what's going on here, here's the explanation:

Below is an example print field. What follows is an explication of the

individual "escape sequences" that make the command work.

```
{print @&f0S@&a1420H@&a5284.53V@*c3280H@*c510V@*c20G@*c2P@&f1S}
```

First of all, you have to imagine that anywhere you see @ in this explication you substitute the Escape character (which doesn't print in Word), this is decimal 1B, Ascii 27 (and has nothing whatever to do with the Escape key at the top left of your keyboard). It is a special character that signals the Laserjet that what follows is a command sequence and not something to be printed.

@&f0S -- this Pushes the current cursor position into memory.

@&a1420H -- this is the horizontal position (from the left edge of the page) to begin printing the shaded area. It is measured in Decipoints. There are 720 decipoints to an inch.

@&a5284.53V - This is the vertical position (from the top edge of the page) to begin printing the shaded area, measured in decipoints..

@*c3280H -- This is the height of the shaded area, measured in decipoints.

@*c510V - This is the width of the shaded area, measured in decipoints.

@*c20G - this is the shading for the area (in this example 20 represents 20%).

@*c2P -- this prints the fill as either shaded or cross hatched. (In this example the 2 represents shaded; 3 would create a cross-hatched pattern)

@&f1S -- and finally this Pops the saved cursor position.

If you wish to tweak a field created by the Best Guess mode, simply bear in mind that all of the measurements are in Decipoints -- 720 decipoints to an inch -- so, for example, increasing the vertical position by 720 decipoints will move the box down on the page by one inch.

Revision History

Version 1.0, released April 26, 1991

First public release.

Version 1.1, released April 30, 1991

Fixed several problems with calculating placement and width.

Fixed handling of hanging indent paragraphs.

Removed Cell, Row, and Picture options from Area group. These will be implemented, if possible, in a later release.

Added an option button, **Remove**, to remove {print } fields from the current document. Added a checkbox to confirm this procedure.

Version 1.3, release May 6, 1991

Added manual mode.

Revised the LineSpace() function to accomodate for fixed fonts and LinePrinter.

Replaced the fixed Magic number Leading with a percentage (.2) of the current font size.

Registration

Click

This macro is distributed as ShareWare. If you use this macro you must register and send \$25.00 to the address below.

Registration will place your name on a mailing list for future Gadfly macro products. And entitles you to use any future version of ShadeParagraphPCL without further payment.

Corporate site licenses are available. Please contact the author at the address below and a fee will be negotiated for the specific macro and use involved.

© Guy J. Gallo

219 East 69th Street, NYC 10021

~~And a legal note: This macro is provided with no warranties. It may not be published without express permission from the author.~~

Gadfly Macros Registration -

Guy Gallo 219 East 69th New York, NY 10021

Registration fee:

\$25.00

NYS Sales Tax (if appropriate)

Total enclosed:

NAME:

COMPANY:

STREET:

CITY:

STATE, ZIP:

Electronic address:

Where did you get ?

COMMENTS:

is copyright 1991, by Guy J. Gallo. No portion of this document or the macros it contains may be modified, copied, distributed or otherwise altered without the express written permission of the author. This includes, but is not limited to, distributing the package for a fee, or distributing personal modifications to the included macros.