

a filesafe Word for Windows screenplay formatter

Version 1.1a

Introduction

The Gadfly Script Formatter (**GScript**) is a suite of styles and macros for *Word for Windows* to aid in the editing and formatting of a screenplay.

In the olden days, writers hired script services to prepare their manuscripts for submission. With the introduction of the Personal Computer, screenwriters either settled for the best their word processor could do, or purchased a second application to create a properly formatted print version of the script.

Files and Installation

If you've gotten this far, then chances are you've already installed properly (since I've built into this document a series of checks to see that everything is where it should be).

Files

You must have the following four files:

GSCRIPT.DOC

This documentation. It is, in fact, a *Word for Windows* template containing several macros to run the demo and set defaults.

GSDEMO.DOC

A sample screenplay used by the GScript Demo.

GSDRAFT.DOT

The draft template. It contains all of the style definitions and macros necessary for the entering and editing of a screenplay.

GSCRIPT.DOT

The main application template. It contains all of the style definitions of GSDRAFT.DOT, plus several macros for the final formatting of the screenplay.

Installation

All four of the above files should be copied to the directory that contains your template (DOT) files. Usually this is the same directory that contains WINWORD.EXE.

What does GScript do?

GScript formats a screenplay with proper indents, font selection and size, and strings inserted at the top and bottom of the page (when the script is formatted for printing). The resulting layout conforms to the standard established ages ago

(before the dawn of the Personal Computer), by the Writer's Guild of America and the Studio System. This format is based on the formula you've heard a zillion times:

One page equals one minute of screen time.

That may seem like an arbitrary producerish concern. In fact it possesses a general truth that can be extremely helpful during the writing of the script.

The font used in this template is Courier 12 pt. If you are accustomed to measuring indentations in terms of characters (rather than in terms of inches), Courier 12 pt is a 10 "pitch" font: there are ten characters per inch.

Here are the specifications:

Font:

Courier 12 point, 10 pitch (that is, 12 points high, 10 characters per inch). This is the same font you might remember as standard on the antique IBM Selectric typewriters.

Margins:

Left - 1.25"; Right - 1.0"; Top - 1"; Bottom 1"

Character Name:

2.5" (26 Characters) in from left margin

Dialogue:

1.2" (13 Characters) in from left margin

Parenthetical

1.9" (20 Characters) in from left margin (the first line has a hanging indent of one character).

Transition

4.5" (46 Characters) in from left margin

The fact is, you can cheat and make your screenplays *appear* shorter by increasing the margins for action and dialogue; you can use proportionally spaced fonts (like the Times Roman 12pt used in this document). The reasons against doing so are two fold:

- 1) No one is fooled. A script that comes in at 120 pages of 10pt Helvetica is, we all know, even producers know, actually 132 pages of script.
- 2) It helps the actual composition of the screenplay if it is formatted according to these styles and indents. The flow of the script, the tempo and timing is simply easier to see, to feel, if the script is formatted in Courier 10 pitch with proper margins. You can sense how much description is on the page, how much dialogue.

What more does it do?

GScript has defined several "short cut" keys for applying styles:

Style

Function

Key

Char

Character Name

Alt-C

Dialogue

Dialogue

Alt-D

DialogueItalic

Emphasized Dialogue

(no key)

Paren

Parenthetical

Alt-P

Transition

Transition Break

Alt-B

heading 3

Slugline

Alt-S

Normal

Action

Alt-A

Note: These styles are also listed on the Format menu

These are the paragraph styles specific to screenplay format. In addition **GScript** contains specifically defined styles that allow functional use of the outliner, footnotes, and annotations.

These additional paragraph styles are all defined as "hidden" so that they will not print unless you specifically tell *Word* to print them.

What GScript does that other formatters don't

Michael Gross, a Beta Tester, described GScript as "Filesafe". What he meant was: GScript makes no changes to your document that cannot be *easily* reversed. It does not tamper with the disk file. In fact, it doesn't even save its additions to your file unless you specify that it should or do so manually.

All other script formatters that I know of require some kind of conversion from your word processor to a format they can read and understand. Even if they can convert in the other direction, back to your word processor, something is inevitably lost in the translation. Not so with GScript.

More specifically, since GScript is integrated with *Word for Windows* it possesses several advantages:

- 1) You don't have to create an ASCII file for formatting. **GScript** is a unified system capable of creating and printing both "draft" scripts, and fully formatted "presentation" scripts.
- 2) Other formatters either ignore, strip, or choke on some of the most useful tools found in powerful word processors. With **GScript**, since you have complete control over what styles are hidden, you can fill you screenplay with notes, outline levels, footnotes, annotations and the like, and still create a formatted print file without changing a thing.
- 3) There is no need for two copies of your screenplay (i.e. one draft, one presentation), since **GScript** is capable of stripping out all of the added strings and all of the altered styles.

Note: even in "draft" mode the styles in GScript are such that you will never get "stranded" character names at the bottom of a page, and dialogue will never be broken.

Notices

GScript is copyright 1990, by Guy J. Gallo. No portion of the macros contained in this template, or this documentation, may be modified, copied, distributed or otherwise altered without the express permission of the author. This includes, but is not limited to, basing a similar application upon the macros contained in **GScript**, distributing the package for a fee via any method, disk or electronic.

GScript is released as ShareWare. You may evaluate it for a period of 30 days. At that point a registration fee of \$60.00 must be paid if you continued to use this software. Consider, please, that the commercially available alternatives cost approximately four times as much.

If you desire the style definitions and macros and key assignments without the formatting ability of **GScript**, you may use the included GSDRAFT.DOT, a sub

set of GSCRIPT.DOT, which is distributed as FreeWare. That template contains all the aids for entry and editing of a draft screenplay.

GScript is provided as is. I make no warranties and take no responsibility for possible data loss or corruption. With a system this complex there is always the possibility of error. Make backups first...

FormatScript

Introduction

While the paragraph styles defined in **GScript** are useful in and of themselves, and would produce a printable copy of your screenplay, the meat of the matter lies in a macro called **FormatScript**.

FormatScript examines your screenplay and determines where each page break falls. It then performs one of several actions, dependent upon where the break occurs.

If it falls in the middle of a scene, either between paragraphs of action or within a paragraph of action, then it inserts the bottom and top Continued Strings.

If it falls at a slugline it does nothing (except remove the "space before" amount from the slugline).

If it falls within a paragraph of dialogue, it inserts the Dialogue More String, inserts the bottom and top Continued Strings, and inserts the character's name at the top of the continued speech along with the Character More String.

Here is a graphic of the main **FormatScript** dialog box:

And here follows a description of each of the functions:

Mode

The Mode options are grouped as OptionButtons. You will notice that you can only have one button active at a time.

Automatic

If you select this mode, **GScript** operates without user intervention, making intelligent decisions about page breaks when they fall within paragraphs of dialogue or action.

By default the break will be placed one line up, unless doing so would strand single lines at either the bottom of this or the top of the next page.

Interactive

In this mode **GScript** will stop and prompt you for confirmation when it

encounters a page break that falls within a paragraph of dialogue or action. It will present you with the following menu:

By default **GScript** recommends one line up from the current cursor position. However, you can break the paragraph at the insertion point, or two lines above the insertion point, or at the top of the current paragraph.

Note: selecting Paragraph when you are within a paragraph of dialogue actually places the break before the Character Name of the current paragraph of dialogue (so technically, in the case of dialogue, it is "two paragraphs up").

Remove All

This is one of the great virtues of **GScript**. It is a simple matter to remove all formatting instructions, hard page breaks, and additional strings inserted by running FormatScript in Automatic or Interactive mode.

After running FormatScript in this mode (or running the macro StripFormatting directly), you are left with a version of the document that is suitable for draft printing. There will be no inserted "Continueds" or "Mores", etc.

Options

The option choices are all checkboxes. You can have more than one checkbox active at a time.

Scene Page Count On

Default is Off.

If this option is checked, then after FormatScript has gone through the document determining page breaks and inserting Continued and More strings, it will go back through the document and count the length of each scene and place the number to the right of the top continued string.

For example, if scene 111 is four pages long and begins on page 75, the following would be printed at the top of the following pages:

Page 76 (Nothing. This is the first "over-run" and is not noted):

111

Page 77 (This is the second full page of over-run):

111

Page 78:

111

Page 79:

The process of determining each scene's length is rather complicated (and terribly clever), and so it adds approximately ten minutes to the time needed to format a full length script.

~~Scene Numbers On~~

Default is on.

If this option is checked, then current scene numbers will be printed to the left of the top continued.

You can turn this off if:

You do not have your sluglines numbered, or

You have your sluglines numbered and simply don't want to insert them on continued pages.

~~Renumber All SlugLines~~

If this option is checked, **GScript** will renumber all sluglines before proceeding with the format. (Like StripFormatting, you can run this macro independently of **FormatScript**. It is called **NumberAllSlugs**).

If you entered slugline numbering as you wrote the script, then this is unnecessary. If however, the script is not numbered, or not completely numbered (you just added a scene), then you should check this box.

You will be presented with a dialog box that looks like:

Left:

Right:

Both:

Note that if you select either Left or Both then the indent of heading 3 (Slugline) will automatically be changed to a "hanging indent" of -.5" so that the numbers will print in the "gutter" region of the page.

Remove Outline numbering: This checkbox is only needed if you have previously numbered your screenplay using the built in outline numbering function of *Word for Windows*, found on the **Utilities Renumber** menu. It will first remove all the outline numbering before proceeding. (For a note on why this is necessary, and why **GScript** does not use the built in outline numbering, see the Technical Notes on page Error: Reference source not

found.)

~~Resize if Automatic~~

If this option is checked, and the Mode selected is Automatic, then **GScript** moves and resizes *Word for Windows* into the upper left of your screen. This allows you to easily access another *Windows* application (although, be forewarned that things will slow down).

~~Print File When Done~~

If this option is checked, then **GScript** will automatically begin printing the screenplay when it is finished formatting. This is useful if you are running in Automatic Mode and want a finished print out when you return from lunch with Sam.

~~Save To New File Name~~

If this option is checked then before anything else, **GScript** will prompt you for a new file name and make a copy of your script. This is useful if you don't trust **GScript** (which you shouldn't at first, I suppose). It isn't, as I've said before, necessary, since **Remove All** will strip all the formatting from the document.

~~Save Settings As Default~~

If this option is checked then before anything else, **GScript** will write the current settings -- the Mode, the Options checked, and the Strings -- to your WIN.INI file. The next time you run **GScript** it will use those settings.

Strings

These are the specific strings that will be placed in your script at page breaks.

~~Dialogue More String~~

What will be inserted below a broken paragraph of dialogue:

and so I said to Gertrude, I can't

--more--

~~Character Continued~~
~~String~~

What will be inserted next to the Character Name on the top of the following page, when a page break is inserted into a paragraph of dialogue, e.g.

FRANK (cont'd)

take this anymore. Then she hauled off and hit me with a skillet.

~~Bottom Continued String~~

What will be inserted at the bottom of the page when a scene continues:

(CONTINUED)

~~Top Continued String~~

What will be inserted at the top of the page when a scene continues:

CONTINUED:

Limitations

At this time, FormatScript is an all or nothing proposition. That is, you cannot run it on a half formatted screenplay. If you have created a formatted screenplay, added some scenes, deleted others, and then want to format the script again, you must first StripFormatting and start over.

Whether or not you have to RenumberAllSlugs will depend on whether or not you numbered the new slugs as you entered them.

Click here -> [☒](#)

Macros

The macros found in GSCRIPT.DOT template cannot be viewed or edited. This may sound, at first, like a limitation, but consider:

The main macro, FormatScript, is incredibly complex. Changing anything in it, deliberately or inadvertently, could result in its not functioning at all.

This template is intended specifically for the final phase of preparing a script. There should be no need whatever to alter the macros.

You could, if you wish, still record a macro and store it in GSCRIPT.DOT. You can add glossaries and you can (though it is not recommended) alter or add styles.

GScript includes a second template, GSDRAFT.DOT, which contains all of the macros necessary for entering and editing a screenplay. If you wish to modify any of the editing macros, or to add macros of your own, do so to GSDRAFT.DOT. The style definitions in GSDRAFT.DOT are identical to the style definitions in GSCRIPT.DOT. There is a menu item (under the Utilities menu) to toggle between the two templates.

GSCRIPT.DOT is fairly large. Saving it take time. If you are going to add macros, glossaries, key assignments or menu assignment, it is best to do so withing GSDRAFT.DOT. I don't know about you, but life's too short to spend it watching that damn hour glass.

Macros Reference

These are the macros included in GSCRIPT.DOT:

Macro Function

ChangeThisPara

Converts a paragraph style to the style expected by the FormatScript. Does the same as Convert Styles, but on a paragraph by paragraph basis. This is useful if you have directly formatted paragraphs rather than styles.

ConvertStyles

Converts stylenames to those expected by the FormatScript.

FormatScript

Formats the screenplay. This is the big cheese.-

NumberAllSlugs

Inserts a {seq slug} field to number all slug lines. Takes some time.

NumberAlphas

Allows you to number a scene with an alphabetic modifier: 1a, 1b, 11a.

NumberOmitteds

Allows you to specify that a scene or range of scenes has been omitted.

NumberThisSlug

Numbers the current slug line. If you enter this as you work on the script (and then update fields), it is unnecessary to NumberAllSlugs.

NumTopContinueds

A macro called by FormatScript to place scene length numbers at the top Continued. Adds several minutes to the time needed to format a script. -

SideBySideDialogue

Inserts a table for side by side dialogue

StripFormatting

Removes all of the strings and styles (and paragraph changes) made by FormatScript. This is, I think, very useful, since you may wish to print "draft" copies of the screenplay as you work, and only FormatScript when a draft is completed.-

StripNumbers

Removes all of the {seq slug} fields used to number slug lines

ToggleSlugIndent

Toggles the left indent of slug lines so that if there are numbers they will print in the gutter. If you have sluglines numbered, then the indent should be -.5; if you have no numbers, it should be 0". This macro toggles between the two.

Menu Geography

Documentation Menu Choices

There are only three macros associated with this documentation file.

SetDefaults

Located on the Macro menu.

GScriptDemo

Located on the Macro menu.

PrintRegistrationForm

Located on the File menu.

GSCRIPT.DOT Menu Choices:

~~Added to Insert~~

The command **Side By SideDialogue** inserts a two column table and the necessary styles to begin entering simultaneous dialogue.

~~Added to Format~~

The commands added to the Format menu are the most frequently used; they are the commands to format paragraphs for the various parts of a screenplay.

Slugline

applies the style "heading 3"

Action

applies the style "Normal"

Character Name

applies the style "Char"

Dialogue

applies the style "Dialogue"

Parenthetical

applies the style "Paren"

Break

applies the style "Transition"

The last two items on this menu alter the format of the sluglines:

Toggle Slug Indent

This command toggles between a left indent of 0" (appropriate if the slugline is not numbered), and -.5" (appropriate if the slugline is numbered).

Number This Slug

This command places a sequential number on the current slugline. It places the number left of the slugline by default. If you have run NumberAllSlugs, it determines whether to place it Left, Right, or Both according to the choice made during the last usage of NumberAllSlugs.

~~Added to Utilities~~

FormatScript...

This is the main formatting macro.

Number Sluglines...

This will number all sluglines sequentially

Strip Numbers

This will remove all scene numbers

Strip Formatting

This will remove all formatting

Reset Small Kaps

The command **Reset Small Kaps** is necessary only if you have one of the font managers that can produce "scaleable" fonts, and you have entered Sluglines,

Character Names, and/or Transitions in mixed case. This is explained below (page Error: Reference source not found).

The next two commands are useful if you are converting a document from another word processor or from a different *Word for Windows* template.

Convert Styles

Allows you to interactively map current stylenames GScript stylenames

Convert Paragraph

Allows you to map the current paragraph to a GScript style name.

Attach GSDraft

This command attaches the draft template, which allows for editing of macros (and which save much quicker if you have a mind to add your own macros, glossaries, styles, key definitions, etc).

Miscellaneous notes about the template

GScript's Use of Outline Levels

The outline definitions in **GScript** are solely based on my own preferences. In writing screenplays I have found it useful to work from a very skeletal outline that consists of general act divisions, beneath which are sequence divisions, beneath which are the actual scene sluglines. For example:

The above graphic is the included example document with only outline levels showing: Heading 1 for Act Divisions; and Heading 2 for "Sequence" descriptions such as "Titles", "Girl meets Boy", and the like. Sluglines are always (and must be) formatted as Heading 3. This organization allows for simple navigation through a long script.

Note: even if you don't use heading levels 1 and 2, sluglines must be formatted as heading 3. Otherwise FormatScript will not function properly.

The Styles

One of the neatest features of *Word for Windows* is the ability to define what the next paragraph style will be. As a result, for instance, if you press Alt-C for a Character Name, and then press enter, you don't have to press Alt-D to format the paragraph for Dialogue. It already is. Similarly, pressing enter in a line of dialogue automatically styles the next paragraph as Character Name. This makes for incredibly simple text entry.

The "follow" specifications are as follows.

Current Style

Next Style

heading 1

heading 2

heading 2

HiddenNote

(page - 18)

heading 3 (Slugline)

Normal (Action)

Normal (Action)

Normal (Action)

Char (Character Name)

Dialogue

Dialogue

Char

Paren

Dialogue

Transition

heading 3 (Slugline)

Headers and Footers

The template has pre-defined headers and footers.

The header is {Author} - {Title} flush left and (page - {page}) flush right.

The footer is {printdate} flush left

The predefined font for Headers/Footers is Courier 10pt. If you don't have 10pt Courier, it will print as 12pt Courier. You can change this to suite your preferences.

The fact that the header uses "fields" means that you must edit SummaryInfo (EditSummaryInfo), and place the Screenplay title in the Title edit field. Author should automatically be filled in.

If you want a different layout, or different text, or a different font for your header or footer you can either:

- 1) edit GSCRIPT.DOT and GSDRAFT.DOT directly, and change the default header/footer text, or
- 2) change the header and footer within the screenplay itself.

SideBySide Paragraph Styles

GScript contains macros and styles for easily inserting side by side dialogue. The command is located on the **Insert** menu; the styles associated with this command are:

SbSChar

SideBySide Character
(no key)

SbSDialogue

SideBySide Dialogue
(no key)

The command that inserts a table of two columns is located on the Insert menu.

Colors

GScript has predefined color attributes. For instance, the hidden heading levels 1 and 2 are red; heading 3 (Slugline) is blue; Annotations are red, and footnotes are blue; headers and footers are green.

You can change any of these particular paragraph styles to other colors if you wish (since they are hidden to FormatScript).

However, do not define any of the "visible" paragraph styles (Char, Dialogue, DialogueItalic, etc) as red. The reason for this limitation is that 1) FormatScript uses red to mark paragraphs of action and dialogue that have been broken by formatting and 2) StripFormatting uses the color red as a search criterion when it is trying to find and restore those paragraphs that were broken by FormatScript.

Conversion of other documents

It is likely that the first time you use **GScript** you will be interested in formatting a document that was either created in another word processor and imported into *Word for Windows*, or created using your own template (with your own style names, indents, font and font sizes).

In order for FormatScript to work properly these documents must be converted into a recognizable form. I have included two macros to help with the conversion.

ConvertStyles

If the screenplay is based on a template, with styles defined for the six screenplay paragraphs Slugline, Dialogue, Parenthetical, Action, Character Name, and Transition, then simply run the macro named ConvertStyles. You will be presented with a dialog box that looks like:

The left hand listbox contains all of the styles currently defined in the document. The right hand listbox contains eight functions -- the six basic screenplay paragraph styles, plus Italic Dialogue and Notes.

You simply select a style in the left box and the function it corresponds to in the right box. In the above example, "C" would map to "Character", "D" would map to "Dialogue".

What the macro actually does is a search and replace for all paragraphs styled as "C" and changes them to "Char" style (Char is the name for the Character Name style).

The checkbox titled "Remove inactive styles?" performs a very interesting and useful function. Let's say you've imported a screenplay from PCWord. The resulting document contains all of the styles defined in the PCWord style sheet. Yet the document only uses a portion of those styles. Clicking on this option and then selecting OK searches through the list of available styles, checks to see if a style is actually used in the document, and deletes those that are not active.

This considerably shortens the list of styles displayed in FormatDefineStyles. And if it is run first, it speeds up the operation of ConvertStyles.

Note: This macro is specific to **GScript**. The styles needed for **GScript** operation are omitted from the test/deletion. However, if you want a version of this macro that will work on other imported documents, look for GTOOLS.ZIP on CIS or a BBS near you. There is a macro called **EditClearIdleStyles** which removes all unused styles from a WfW document.

ChangeThisPara

What happens if you have directly formatted your screenplay using the ribbon and ruler? The result would be that **all** paragraphs would be variations on the Normal style, just with different indents.

In this case, you can use the macro named **ChangeThisPara**. What this macro does is look at the current paragraph (the one containing the insertion point), determines all of it's specific characteristics (indents etc), then presents a dialog box that looks like:

You simply select the function that all instances of this paragraph should map to, and the macro search through the document for paragraphs with the same characteristics and formats them with the appropriate **GScript** style.

A Note on Carriage Returns

If you are coming to *Word for Windows* from one of the less powerful word processors, then you are probably accustomed to entering double spacing by entering two carriage returns between those paragraphs that require extra spacing, as, for instance, between Action and Character Name, or between Dialogue and the next Character Name.

One of the most powerful features of *Word for Windows* is the fact that you can specify, when defining a paragraph format, how much space to skip before the paragraph, and how much to skip after the paragraph. As a result, there is no need for extra carriage returns.

Here are the Space Before specifications used in **GScript**:

Style
Space Before

heading 3 (Slugline)
1.5 lines (18pts)

Char
1 line

Dialogue
(no space before)

Action
1 line

Transition
1 line

If you are importing a document from another word processor, or from a text file, it is possible that you will have double spacing doubled by the existence of multiple carriage returns. (The best indication that this is a problem is if your action paragraphs are separated by much more than one line).

If this is the case then you must strip out the extra carriage return. You could do so manually (how tedious), or you can use *Word for Windows* to Search and Replace pairs of carriage return with single carriage returns.

1. Select Replace from the Edit menu

2. In the Search Field enter

^p^p

3. In the Replace Field enter

^p

Whether you leave Confirm checked or not depends upon your patience and your trust. (This should work on *most* files, but cannot be guaranteed for all.)

~~Second Draft Tools~~

The following tools are helpful if you are preparing a production draft or a revision of a formatted screenplay.

Note: If you have formatted a screenplay, and numbered your scenes, and wish to freeze the scene numbering, you should select the entire document (Ctrl-Keypad5) and press Ctrl-Shift-F9 to turn all the {seq slug} fields into their most recent result.

Revision Marks

Screenplay formatters sometimes allow you to mark "X-Changes", that is, lines that have changed from draft to draft.

Since GScript is integrated with *Word for Windows* you have only to turn on Revision Marks for all changes to be displayed between drafts.

abc Scene Numbering

If you wish to number a scene with an a, b, c mark (rather than renumber the entire screenplay, simply run the macro NumberAlphas (found on the Utilities Menu).

You will be presented with the following menu.

Enter the scene number and which alpha increment to append.

Know that you must currently be located on a heading 3 paragraph (a Slugline). If you aren't, an error message is printed.

This command is intended for use *after* you have formatted a presentation script and do not wish to renumber the sluglines. For that reason you might wish to freeze all fields by selecting the entire document and pressing Ctrl-Shift-F9.

This macro is not included on any of the menus. You can either run it directly by selecting Macro.Run and entering NumberAlphas, or you can assign it to a key combination or menu of your choice.

Omitted scenes

If you are preparing a shooting script and wish to denote omitted scenes, run the macro NumberOmitteds (found on the Utilities menu).

You will be presented with the following menu:

Know that you must currently be located on a heading 3 paragraph (a Slugline). If you aren't, an error message is printed.

This command is intended for use *after* you have formatted a presentation script and do not wish to renumber the sluglines. For that reason you might wish to freeze all fields by selecting the entire document and pressing Ctrl-Shift-F9.

This macro is not included on any of the menus. You can either run it directly by selecting Macro.Run and entering NumberOmitteds, or you can assign it to a key combination or menu of your choice.

Some Technical Notes

Note on Small Kaps and UpperCase

Word for Windows does not have a "Character Style" for all UpperCase. As a result, those paragraphs which are always formatted with upper case letters are formatted as Small Kaps, using a point size of 12.5 (instead of 12).

In this template, the paragraph formats affected are **Heading 1** (Slugline), **Char** (Character name), and **Transition**.

The result, on most printers that support Courier 12pt, is the equivalent of all upper case.

For instance, on the Laserjet I, II, IID, IIP, and III without soft fonts, Small Kaps 12.5 is equivalent to UpperCase 12pt. This is because when the Laserjet (and Windows) encounters a font that doesn't actually exist it tries for the closest match possible.

In fact, even if the line in question is mixed case (both capital and lower case), it will print as all capitals.

The problem arises when you do have a 12.5pt font. There are two ways: either with a soft font (which is unlikely), or if you have Adobe Type Manager, Bitstream FaceLift, Zenographics SuperPrint, or (someday) Microsoft's TrueType. These programs allow for "scaleable" fonts in increments of .5 points. As a result, if you ask for 12.5pt Courier, that's what you'll get.

So what's the result? If you type a line as all lowercase, there's no problem. Even with these "scaleable" fonts, 12.5pt lowercase will print as 12pt uppercase. The problem lies when a line, for whatever reason, contains both upper and lowercase (remember, these are paragraphs that are formatted as Small Kaps, 12.5pt). In the case of a word that is both upper and lower case, formatted as 12.5 Small Kaps, on a system that has scaleable fonts, what will happen is: the capitals will print as upper case 12.5; the lower case will print as upper case 12pt.

The solution: I have included a macro called **ResetSmallKaps** which will search for every Slugline, Character Name, and Transition and toggle it to all LowerCase.

Because there can be very many paragraphs to find and toggle, this macro can take a far amount of time. Be forewarned.

Why {Seq Slug} instead of {Autonum}?

The reason I used "sequence" fields to number the sluglines, rather than the more automatic (and faster) "outline" numbering is simple: it is an easy matter to

"refer" to the most recent {seq} field, while there is no way, that I have discovered, to refer to the most recent {autonum}. This is how I can easily place the current scene number at the right of the slugline, and how I can easily place it at the top continued of the next page.

Styles used by FormatScript

When FormatScript inserts strings into your document, it uses specific styles. The style names are:

tContinued

bContinued

More

You should not delete these styles from GSCRIPT. They must exist for both FormatScript and StripFormatting to function properly.

WIN.INI Settings:

The following keywords are written to WIN.INI when **GScript** is installed.

Keyword

Meaning

Mode

0 = Automatic, 1 = Interactive

TCont

Top Continued string

BCont

Bottom Continued string

More

Dialogue More String

DCont

Character Continued String

NumTop

0 = Don't count scene length; 1 = Put Scene Page Count after Top Continued (this takes some time).

SlugNum

0 = insert scene numbers; 1 = Don't insert scene numbers (useful if you don't have sluglines numbered).

(page -27)

NewFile

0 = don't create new file; 1 = save formatted file to a new name

PrintFile

0 = off; 1 = Print the file after formatting (useful for automatic mode)

Renumber

0 = off; 1 = Renumber all slug lines before formatting (takes time)

Iconize

0 = off; 1 = resize and move Word for Windows if we are in Automatic mode. This is useful when you want to work with another Windows application while **GScript** is busy... Note, things will still be slower.

NoWarning

This setting more or less follows Mode. If it is 1 then the time warning message box displayed when running NumberAllSlugs and NumTopContinueds will not display.

Installed

Simply tells some of the internal macros if **GScript** has been installed.

Side

Used by NumberAllSlugs, NumberThisSlug, NumberOmitteds, and NumberAlphas to determine where (0=left, 1=right, 2=both) to place slugline numbers.

Support, Updates and Disks

~~GScript is being distributed as ShareWare via electronic bulletin boards and on line services.~~

Support

~~There is no direct telephone support.~~

~~I will answer all online correspondence at the electronic addresses given below. If you do not have a modem or access to these electronic addresses, then write to the postal address given below, but be forewarned: you will get a much faster response over the wires than over land.~~

Updates

~~Updates will be handled in the same manner: that is, they will be distributed via modem to these electronic services. Once you are registered for GScript, you are registered. Simply acquire the most current version from wherever you acquired the first version.~~

Disks

~~It is possible to order disks directly from the below address, however 1) the disk will contain nothing additional to the distributed ZIP; and 2) there will be a shipping charge of \$10.00 (domestic), \$15.00 (international).~~

Reaching the Author

Electronic Mail Addresses:

CompuServe ID# - 71171,3555

Internet gg2@columbia.cu

SmartNet Network, Windows Conference

InterLink Network, Windows Conference

WGA BBS #938

Mail Address

G. Gallo
219 East 69th Street
New York, NY 10021

Future versions of GScript

A last word: this is the initial release of **GScript**. I have tried, with the help of several beta testers, to see that it is bug free and up to all the possible variations that a manuscript can put in its way. I'm sure that will prove to be fatuous.

Please report any bugs you might find to the above addresses.

Also, I am interested in hearing user suggestions for features in future versions of **GScript**.

Possible additions:

- Greater "shooting script" formatting capabilities, such as increased automation of Alpha and Omitted scenes, and a-b page numbering

- A report of scene lengths

- Better handling of two column dialogue

- Possible version for three camera television script format

Please let me hear from you. And, if you find **GScript** useful, please register. The \$60.00 requested is not much for a working screenwriter. If you think it's too little, I'll gladly accept additional contributions.

-

~~Gadfly Script Formatter, version 1.1~~

Registration

~~GScript Registration fee:~~ _____ ~~\$60.00~~

~~NYS Sales Tax (if appropriate)~~ _____

Total enclosed:

GScript is copyright 1990, by Guy J. Gallo. No portion of this document or the macros contained in the associated files, GSCRIPT.DOT and GSDRAFT.DOT, may be modified, copied, distributed or otherwise altered without the express written permission of the author. This includes, but is not limited to, basing a similar application upon the macros contained in **GScript**, distributing the package for a fee, or dis-tributing personal modifications to the

NAME:

COMPANY:

STREET:

CITY:

STATE, ZIP:
included macros.

Where did you get GSCRIPT?

COMMENTS: