Contents

Introduction

What does GScript do?
What GScript does that other formatters don't
Files and Installation
Getting Started
Formatting Methods
Notices

GScript Styles

Definitions and Usage
Styles and "follow" specifications
Styles and Line Spacing
A Note on Double Carriage Returns
A Note on Removing Extra Carriage Returns

FormatScript - The Main Formatting Macro

FormatScript - Introduction
Limitations

GScript Macros Reference

Overview
Alphabetical Listing of Macros

Advanced Topics

Macros
Menu Item Additions
The ToolBar in GScript
GScript's Use of Outline Levels
Headers and Footers
Colors
Revision Marks
Why {Seq Slug} instead of {Autonum}?
Advanced Margin Tweaking
WIN.INI Settings

Price, Support, Updates and Disks

Price
Support
Updates
Disks
History
Possible additions to future versions
Reaching the Author
About the Author

What does GScript do?

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.

GScript greatly reduces the effort required to enter, edit, and format a screenplay.

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). It is used in two stages:

During the creation of the screenplay, **GScript** simplifies editing with specially tailored styles, key assignments, toolbar assignments, and macros.

After a draft of a screenplay is complete and you wish to *format* the screenplay for delivery, with page breaks inserted, and continuation marks inserted, **GScript** does all the hard calculation and work automatically when you run a single macro: **FormatScript**.

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.

Feature	**Specification:**
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:

- 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.
- 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 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:

- You don't have to create an ASCII file for formatting. **GScript** is a unified system capable of creating and printing both "<u>draft</u>" scripts, and fully formatted "presentation" scripts.
- 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 your screenplay with notes, outline levels, footnotes, annotations and the like, and still create a formatted print file without changing a thing.
- 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.

Files and Installation

Files
Installation

Files

You must have the following four files:

File	**Location**	**Purpose**
GSCRIPT.DOC	In your Dot-Path	This documentation. It is, in fact, a *Word for Windows* template containing several macros to run the demo and set defaults.
GSDEMO.DOC	Dot-Path	A sample screenplay used by the GScript Demo.
GSCRIPT.DOT	Dot-Path	The main template.
GSCRIPT.HLP	WfW directory	A Windows Help file.

Installation

To install **GScript** simply copy the enclosed files to the directories specified above.   All of the files, except the help file, should be placed in the directory containing your other templates (DOT files...)

The HLP file should be stored either in your *Word for Windows* directory (the same place as WINWORD.EXE and WINWORD.HLP.

Getting Started

If you are new to *Word for Windows* or are unfamiliar with the use of styles and templates, please review Chapter 8 "Formatting with Styles" (page 189) and Chapter 37 "Document Templates" (page 699) in the *Word for Windows User's Guide*.

To get up and running with **GScript** create a new document based on the GSCRIPT.DOT template:

- Select **New** from the **File** menu.
- Highlight **GScript**
- Press **Enter**

You will be presented with the **FileSummaryInfo** dialog box in which you can enter the title of the screenplay and confirm the author information.

A lot of what follows will be clearer if you access the various macros and style definitions while reading the help file.

Formatting Methods

Gscript is both a collection of style definitions and a collection of macros to make using those styles simple and to format the resulting document in various ways.

There are basically three ways in which you can format your document with **Gscript**: DRAFT, MASTER, and SHOOTING. The differences among these formatting methods, and the way to access them, are discussed below:

DRAFT
MASTER
SHOOTING

DRAFT

This mode is the simplest.   It requires no extra steps.   If you follow the style guidelines, you will have a script that is perfectly presentable.   And certainly all you need while you are writing the screenplay.

You create such a draft screenplay simply by entering text and formatting with the speed keys listed in the section on style Definitions and Usage.

Characteristics:

By default the magins are 1.25" left, and 1" top, bottom, and right.

Dialogue (Alt-D) will never be separated from the Character Name (Alt-C) because the Dialog style is defined as "Keep Together" and the Char style is defined as Keep with Next.. However,   because of this it is possible that   long speeches will be moved to the following page, leaving large white space on a page.   And therefore this is not the most economical format.

Sluglines (Alt-S) will never be "stranded" as the last line on a page.

Transition Breaks (Alt-B), however, *can* become stranded at the top of a page.   To prevent this you must manually format the immediately preceeding paragraph of Action or Dialog as "Keep with Next" in the **FormatStyle** dialog box.

MASTER

Master refers to what is known as a "master shot script" or "master scene script". The scenes in this type of screenplay are not, usually, broken down by precise camera setups or camera movement, or shot length. This is intended for first drafts of a screenplay. It is the most readable form for those who aren't technically involved with film-making.

This is also the most economical way to format a screenplay. You do so by running **FormatScript** either from the **GSMainMenu**, or the **Tools** menu, and selectng "Master - dialogue only" in the Break Where option box.

Characteristics:

The effect of this is that when **FormatScript** encounters a page break that would fall in the middle of a paragraph of dialogue, it will insert a hard page break and place the "more" string (-more- by default)at the bottom of the page, and then begin the next page with the Character Name and the "dialogue continued" string ((cont'd) by default).

It does not insert any "continued" strings either at the top or bottom of a page.

For this reason it is the most economical way to print a screenplay. The approximate page savings is 15%. For instance, a script that paginates at 136 pages in Draft mode, will paginate at around 128 or 129 (depending, of course, on lots of small variables).

In this mode it is usually not necessary to have scene numbers for each slugline, but that's a matter of personal preference. I like to use scene numbers because it makes reference to a scene easier (even if your working copy has added pages between the time you submitted the screenplay and Sam calls you in for the meeting...)

SHOOTING

This format is the most thorough and most accurately mimics the format of a screenplay as it would be produced by a script service (in the olden days).

Usually this format is used when submitting a second draft or a director's draft, prior to going into production.

Characteristics:

Scenes are numbered, significant camera angles and setups are included.

You format a script in this format by running the macro **FormatScript** and selecting "Shooting - scenes and dialogue" in the Break Where option box.

There are two additional options (presented as checkboxes beneath "Break scenes and dialogue") which will affect how the screenplay is formatted.

Number Top Continueds

Checking this option before beginning formatting will cause **FormatScript** to insert the current scene number before the Top Continued inserted on the continuation pages.

You should uncheck this option in either of two cases: 1) your scenes are not numbered at all, or 2) they are numbered, but you don't want them inserted on the continuation pages.

This option is only relevant when you select Shooting as the Format type. With Master it has no effect.

Count Pages to each scene

Checking this option before beginning formatting will cause **FormatScript** to calculate the number of pages in a scene and insert the page count to the right of the Top Continued inserted on the continuation pages.

This option adds some time to the formatting process. It is appropriate for final drafts submitted to directors and production staff.

For more explanation of these options Format.

Notices

GScript is copyright 1991, 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 $75.00 must be paid if you continue to use this software. Consider, please, that the commercially available alternatives cost approximately four times as much.

Nothing in this version of **GScript** is disabled.   There is a single reminder screen, which appears when you run the main formatting macro.   When you register, I will send you a key to disable this message.

Please, if you use the program, register it.   It is approximately one-fourth the price of commercial programs that provide similar functionality.

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...

Definitions and Usage

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

StyleName	Function	Key
Char	Character Name	Alt-C
Dialogue	Dialogue	Alt-D
DialogueItalic	Italic Dialogue	(no key)
Paren	Parenthetical	Alt-P
Transition	Transition Break	Alt-B
heading 3	Slugline	Alt-S
Normal	Action	Alt-A

These are the paragraph styles specific to screenplay format.

These key assignments, with the exception of **Alt-A** do not conflict with any default key combination used by *Word for Windows.* It is also unlikely that you have personally customized these key combinations (since they are unavailable for customization in the **ToolsOptionsKeyboard** dialog box.) **Alt-A,** however, is used to access the **Table** menu. This was not an option in *Word for Windows* 1.x; and since I'd already determined to use this key combination for the Action style, and since I can't see how you will be inserting tables into a screenplay very often, I have decided, until convinced otherwise, to retain **Alt-A** for **GScript's** use. This will not, of course, effect **Alt-A** when other templates are in control of your document.

In addition **GScript** contains specifically defined styles that allow functional use of the outliner, footnotes, and annotations.

Style	Function
Annotation text	Notes to yourself
Annotation reference	
Footnote reference	
Footnote text	Another way to keep notes
Heading 1	Act divisions
Heading 2	Sequence divisions

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

In addition, there are styles used by **GScript** for various ancillary functions, and for marking the text inserted during the formatting:

Style	Function
bContinued	For the bottom Continued placed between broken scenes.
ByLine	For the by line on the title page
Heading	Centered, bold, spacing expanded by 3pts. Used for the title of the screenplay on the first page
HiddenNote	Hidden and boxed. For notes
More	For the More screen placed after dialog
SbSChar	Side by Side Character name

SbSDialogue	Side by Side Dialogue
tContinued	For the top Continued placed between broken scenes.
Title	For the title on the title page
TitleAuthor	For the author on the title page

You should not delete any of these styles from **GScript**.   They must exist for both **FormatScript** and **StripFormatting** to function properly.

Styles and "follow" specifications

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
heading 3 (Slugline)	Normal (Action)
Normal (Action)	Normal (Action)
Char (Character Name)	Dialogue
Dialogue	Char
Paren	Dialogue
Transition	heading 3 (Slugline)

Styles and Line Spacing

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

A Note on Double Carriage Returns

One of the most common errors in entering a screenplay with **GScript**, and one that often results in blank pages and incorrect formatting, is the use of extra carriage returns to insert more blank space between scenes.

It is very important that you do not use extra carriage returns to create "*white space*" *between scenes or paragraphs.*

In order to change the line spacing between scenes or between sluglines you **must** change the definition of the style.

GScript includes a macro, **ChangeLineSpacing** to facilitate changing default line spacing. It can be accessed from the **GScript Main Menu.**

A Note on Removing Extra Carriage Returns

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 Replace one at a time or choose Replace All depends upon your patience and your trust. (This should work on *most* files, but cannot be guaranteed for all.)

Also, if you have a document that contains carriage return "triplets" or more, you may have to run the above EditReplace more than once.

However, if you fail to check for carriage return pairs the main formatting macro, **FormatScript will do it for you.**

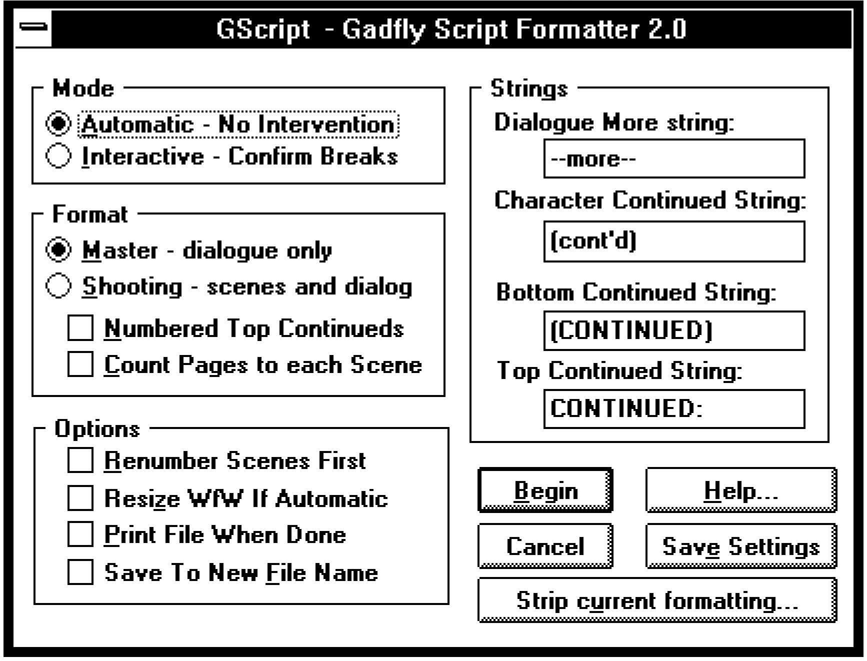
FormatScript - Introduction

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:



GScript - Gadfly Script Formatter 2.0

Mode
● Automatic - No Intervention
○ Interactive - Confirm Breaks

Format
● Master - dialogue only
○ Shooting - scenes and dialog
□ Numbered Top Continueds
□ Count Pages to each Scene

Options
□ Renumber Scenes First
□ Resize WfW If Automatic
□ Print File When Done
□ Save To New File Name

Strings
Dialogue More string: --more--
Character Continued String: (cont'd)
Bottom Continued String: (CONTINUED)
Top Continued String: CONTINUED:

And here follows a description of each of the functions:

Mode
Format
Options
Strings
PushButtons

Mode

There are two **Break** options.   They are grouped as OptionButtons.   You will notice that you can only have one button active at a time.

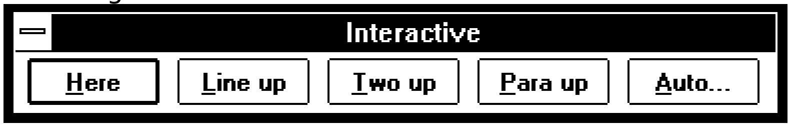
Automatic - No Intervention

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 - Confirm Breaks

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.

Format

Master - dialogue only

This option is useful if you are printing a draft of a screenplay and do not want to break scenes.

In this mode **FormatScript** will check to see if a page break falls in the middle of a paragraph of dialogue, and if so, place the "more" string at the point where the dialogue is interrupted, and place the character name and the "dialogue continued" string before the continuation of the dialogue.

This mode does not insert top and bottom "continued" strings.

Shooting - scene and dialogue

If you select this option button, **FormatScript** will paginate your document and place breaks on each full page, determining if the page break falls in the middle of a scene, and if so, will place a "continued"   string at the bottom of the current page, and at the top of the next page.

It will also check to see if the page break falls in the middle of a paragraph of dialogue, and if so, place the "more" string at the point where the dialogue is interrupted, and place the character name and the "dialog continued" string before the continuation of the dialogue. (This is in addition to the "continued" strings.)

The following check boxes are only relevant if you are breaking both scene and dialogue paragraphs.

Number Top Continueds

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.

Count pages to each scene

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  CONTINUED:
```

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

```
111  CONTINUED: (2)
```

Page 78:

```
111  CONTINUED: (3)
```

Page 79:

```
111  CONTINUED: (4)
```

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.

Options

The following CheckBoxes tell **FormatScript** how to behave while formatting.

Renumber Scenes First

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**). See NumberAllSlugs

Resize if Automatic

If this option is checked, and the Mode selected is Automatic, then **GScript** moves and re-sizes *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

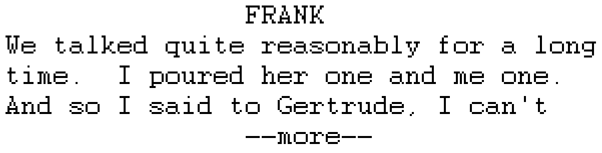
If this option is checked then before anything else, **GScript** will prompt you for a new file name and make a copy of your script. This is useful if you don't trust **GScript** (which you shouldn't at first, I suppose). It isn't, as I've said before, necessary, since **Remove All** will strip all the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

Strings

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

Dialogue More String

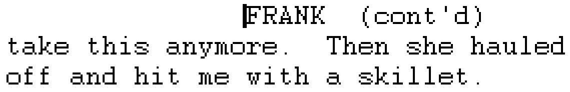
What will be inserted below a broken paragraph of dialogue:



```
                FRANK
 We talked quite reasonably for a long
 time.  I poured her one and me one.
 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:

PushButtons

Here follows a list of the various buttons found on the **FormatScript** dialog box.

Begin

Begin formatting using all the options selected

Cancel

Cancel formatting.

Help

Display the GSCRIPT.HLP file. This file must be located in the Word for Windows directory.

Strip current formatting

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 **StripFormatting** you are left with a version of the document that is suitable for draft printing. There will be no inserted "Continueds" or "Mores", etc.

Save Settings

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 begin with the saved settings as the new defaults.

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.

If you attempt to format a script that has already been formatted, you will be warned and asked to start over.

Overview

Most of the macros found in GSCRIPT.DOT are encrypted. 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.

You can still add custom macros to the template, or modify the speed key and menu and toolbar assignments. You can, of course, add glossaries as well.

Alphabetical Listing of Macros

AutoNew
AutoOpen
ChangeLineSpacing
ChangeThisPara
ConvertStyles
CreateTitlePage
FormatScript
GScriptMainMenu
NumberAllSlugs
NumberAlphas
NumberOmitteds
NumberThisSlug
NumTopContinueds
RevisionAccept
RevisionSearch
RevisionUndo
SideBySideDialogue
StripFormatting
StripNumbers
ToggleHidden
ToggleOutline
ToggleSlugIndent

AutoNew

This is the first of two Auto Macros in GSCRIPT.DOT.

This macro will run whenever you create a new document based on GSCRIPT.DOT.   It runs the FileSummaryInfo command so that you can supply a title and confirm author information for the title page.

AutoOpen

This is the second of two Auto Macros in GSCRIPT.DOT.

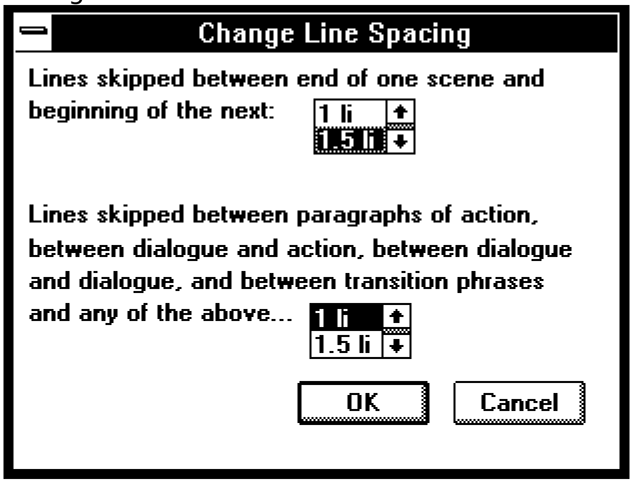
This macro will run whenever you open a document based on GSCRIPT.DOT. It simply returns to the last place you made any kind of edit on the document. You can add to this macro if there are any kind of view preferences you wish to force into effect when you open a screenplay document.

ChangeLineSpacing

Speed Key: None.

Purpose: This macro allows you to change the number of lines skipped between scenes and between paragraphs of action or dialogue.

When you run this macro (available from the **GScriptMainMenu**), you will see the following dialog box.



Change Line Spacing

Lines skipped between end of one scene and beginning of the next: 1 li / 1.5 li

Lines skipped between paragraphs of action, between dialogue and action, between dialogue and dialogue, and between transition phrases and any of the above... 1 li / 1.5 li

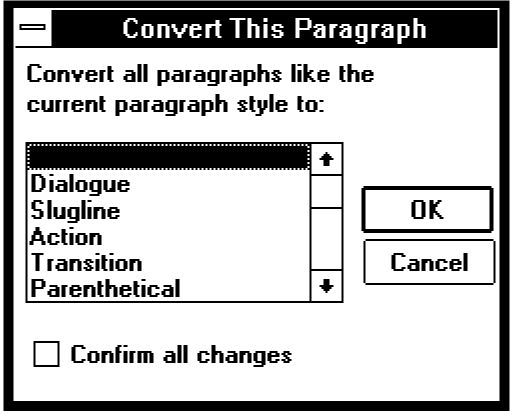
ChangeThisPara

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.

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 its specific characteristics (indents etc.), then presents a dialog box that looks like:



Convert all paragraphs like the current paragraph style to:

□ Confirm all changes

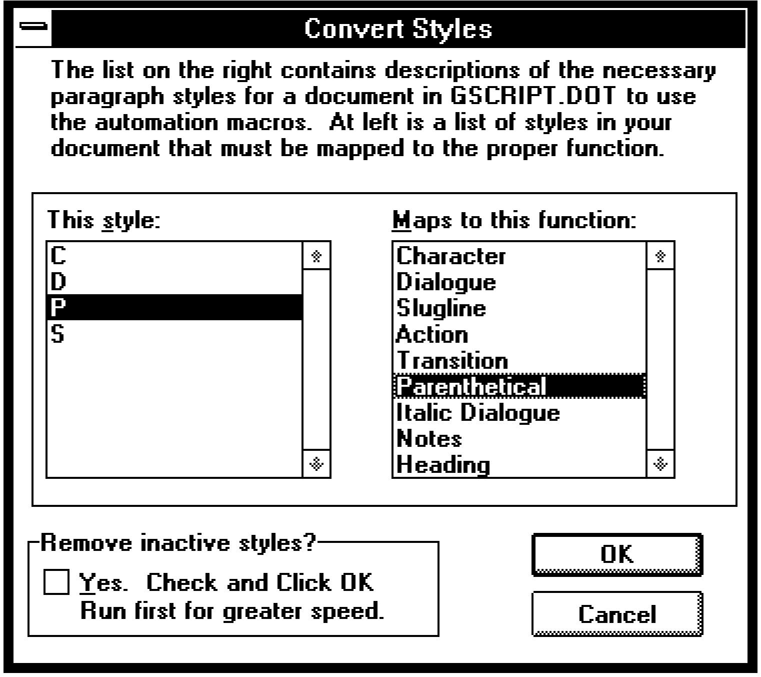
You simply select the function that all instances of this paragraph should map to, and the macro search through the document for paragraphs with the same characteristics and formats them with the appropriate **GScript** style.

ConvertStyles

Converts style names to those expected by the **FormatScript**.

This is particularly useful when you are converting a script already written in *Word for Windows* or Word for DOS.

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



Convert Styles

The list on the right contains descriptions of the necessary paragraph styles for a document in GSCRIPT.DOT to use the automation macros. At left is a list of styles in your document that must be mapped to the proper function.

This style: P

Maps to this function: Parenthetical

Remove inactive styles?

□ Yes. Check and Click OK
Run first for greater speed.

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 *Word for DOS*.   The resulting document contains all of the styles defined in the *Word for DOS* 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**.

Hint for *Word for DOS* users: in earlier versions of *Word for Windows* the conversion of a *Word for DOS* style sheet mapped styles using the mnemonic key combination for the name in *Word for Windows.*   For instance, a style accessed with Alt-D was copied into *Word for Windows* as a style *named* **D.**   This always struck me as odd, and, thankfully, has been changed.   Now the name of the new style is not based on the mnemonic, but on the style description.

So, if you want a *Word for DOS* screenplay style sheet to map accurately to *Word for Windows,* go into the **Gallery** and edit the styles so that the descriptions match the specifications found in the table Definitions and Usage.

CreateTitlePage

This macro will check to see if a title page already exists and ask if you want to replace it or insert a new one.

Speed Key: None.

Purpose: to insert a title page at the top of the screenplay.

How it works: This macro will insert a new section at the top of the screenplay.

It then inserts the predefined glossary entry named TitlePage, and places a bookmark over the inserted text, also named TitlePage.

This glossary entry, stored in GSCRIPT.DOT consists of the following parts:

- A paragraph formatted as Title style, which contains the Field {Title} .This field is replaced with the Title information from **FileSummaryInfo.**
- A paragraph formatted as ByLine style, which is the phrase "a screenplay by". This phrase can be edited to suite your preference.
- A paragraph formatted as TitleAuthor style, which consists of the Field {Author}. This field is replaced by the Author information from **FileSummaryInfo.**
- A two column table which is absolutely positioned, using Frames, at the bottom of the page. The left hand cell contains, by default, the phrase: "Registered with the Writer's Guild of America", and formatted with style TableLeft. The right hand cell contains the Fields {username} and {useraddress} and formatted as style TableRight. These fields are replaced by the information contained in the **ToolsOptionsUserInfo** dialog box.

It is important to understand that this title page uses fields. If you are completely unfamiliar with fields, please review the User Reference Chapter 41.

The most important consequence of using fields is the effect of moving a document from your personal machine to another machine. It is possible, if you update a field before printing, that the information stored on the second machien will be placed in your document. For instance, the UserName and UserAddress is specific to each machine. The fields Author and Title will travel with the document.

There are two ways to avoid this problem.

UnLinkFields - you can select the title page, by pressing F5 twice and selecting the bookmark "TitlePage", and then pressing Ctrl-Shift-F9.

Replacing the fields with custom strings - This may be something you will want to do in any case, since you may not want your own address placed at the lower right. You may want your agent or producer's name there. To do so, simply select that cell and change it to whatever text you desire..

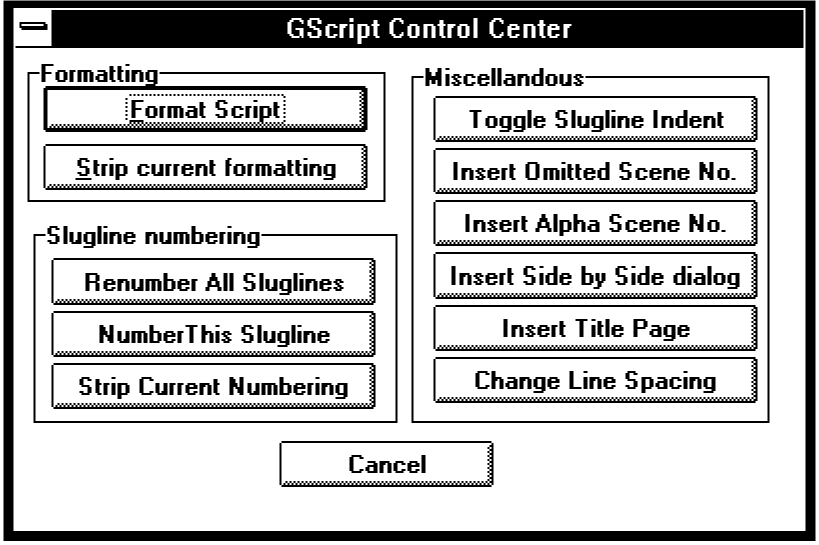
FormatScript

Formats the screenplay.   This is the big cheese.

See FormatScript - Introduction

GScriptMainMenu

A dialog box that displays buttons for several **GScript** functions.



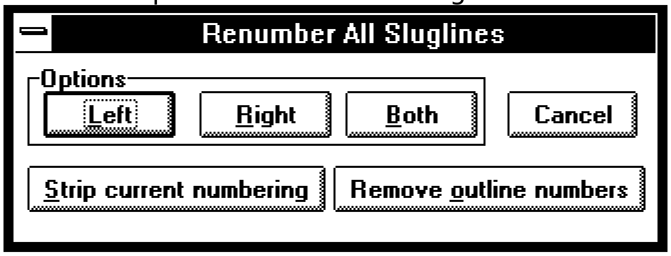
This is an unencrypted macro so that you can customized it -- adding or removing or changing the speed keys to suite your preferences.

NumberAllSlugs

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

Speed Key: None.

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



Left: Places the scene number left of the slugline

Right: Places the scene number right of the slugline

Both: Places the scene number both right and left

Strip current numbering:
Remove Outline numbering:

Strip current numbering:

This button is necessary only if you wish to remove all numbers before proceeding... It runs an independent macro named **StripNumbers.**

Remove Outline numbering:

This button 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,   Styles and Line Spacing.)

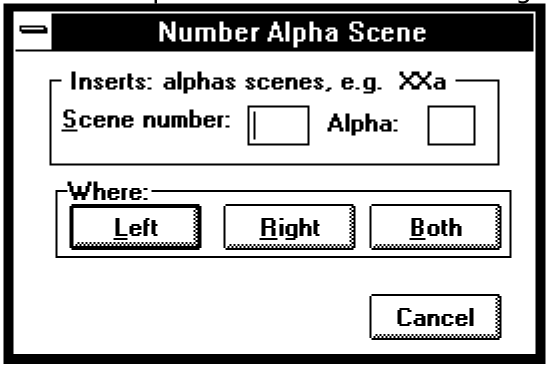
See NumberThisSlug

NumberAlphas

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

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.

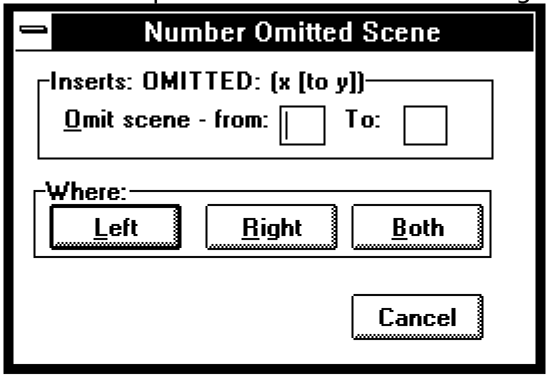
This command is intended for use *after* you have formatted a presentation script and do not wish to renumber the sluglines.   For that reason you might wish to freeze all fields by selecting the entire document and pressing Ctrl-Shift-F9.

NumberOmitteds

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

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.

NumberThisSlug

Speed Key: Alt-N.

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.

This macro should not be called directly.

RevisionAccept

This macro marks the currently selected revision as accepted.

There is no default speedkey (however you could assign one).

It is located on the ToolBar as Icon:

RevisionSearch

This macro searches for the next revision marked text.

There is no default speedkey (however you could assign one).

It is located on the ToolBar as Icon:

RevisionUndo

This macro undoes the currently selected revision.

There is no default speedkey (however you could assign one).

It is located on the ToolBar as Icon:

SideBySideDialogue

Inserts a two column table for side by side dialogue.

Speed Key: none.

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:

StyleName	Function	SpeedKey
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.

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

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.

ToggleHidden

Purpose: to easily toggle between displaying and hiding text formatted as Hidden.

Speed Key: Ctrl-Shift-H

Since, by default, **GScript** uses hidden to format Outline Levels 1 and 2, this is a useful toggle to have installed.

ToggleOutline

Purpose: A small utility macro that will toggle in and out of Outline View.

Speed Key: Ctrl-Shift-O

ToggleSlugIndent

Purpose: 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.

Speed Key: Alt-Ctrl-I

Macros

A Table of GScript's Macros

These are the macros included in GSCRIPT.DOT:

Macro	Function
AutoNew	This macro will run whenever you create a new document based on GSCRIPT.DOT. It runs the FileSummaryInfo command so that you can supply a title and confirm author information for the title page.
AutoOpen	This macro will run whenever you open a document based on GSCRIPT.DOT. It simply returns to the last place you made any kind of edit on the document. You can add to this macro if there are any kind of view preferences you wish to force into effect when you open a screenplay document.
ChangeLineSpacing	This macro allows you to change the number of lines skipped between scenes and between paragraphs of action or dialogue.
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 style names to those expected by the FormatScript. This is particularly useful when you are converting a script already written in *Word for Windows* or *Word for DOS*.
CreateTitlePage	This macro will check to see if a title page already exists and ask if you want to replace it or insert a new one.
FormatScript	Formats the screenplay. This is the big cheese.
GScriptMainMenu	A dialog box that displays buttons for several GScript functions. This is an unencrypted macro so that you can customize it -- adding or removing or changing the speed keys to suit your preferences.
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, it is unnecessary to run NumberAllSlugs. Speed Key: Alt-N
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.
RevisionAccept	Accepts the currently selected revised text.
RevisionSearch	Searches for the next instance of revised text.
RevisionUndo	Undoes the currently selected revised text.
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.

ToggleHidden — Toggles the display of hidden text. Speed Key: Ctrl-Shift-H

ToggleOutline — Toggle in and out of outline view. Speed Key: Ctrl-Shift-O

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. Speed Key: Alt-Ctrl-I

Menu Item Additions

Added to Insert
Added to Format
Added to Tools

Added to Insert

Side By SideDialogue - This macro   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.

For a description of the style definitions of the various styles, please refer to Definitions and Usage

Apply Slugline   applies the style "heading 3"

Apply Action applies the style "Normal"

Apply Character applies the style "Char"

Apply Dialogue applies the style "Dialogue"

Apply Parenthetical applies the style "Paren"

Apply Break applies the style "Transition"

Note that the speed key combinations are listed to the right of the command description

Added to Tools

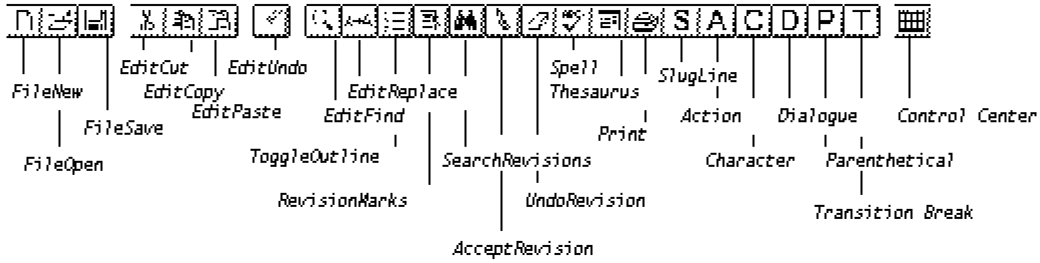
GScript Main Menu - Displays the **GScript** Control Center.   See GScriptMainMenu

FormatScript... - This is the main formatting macro.   See FormatScript - Introduction

The ToolBar in GScript

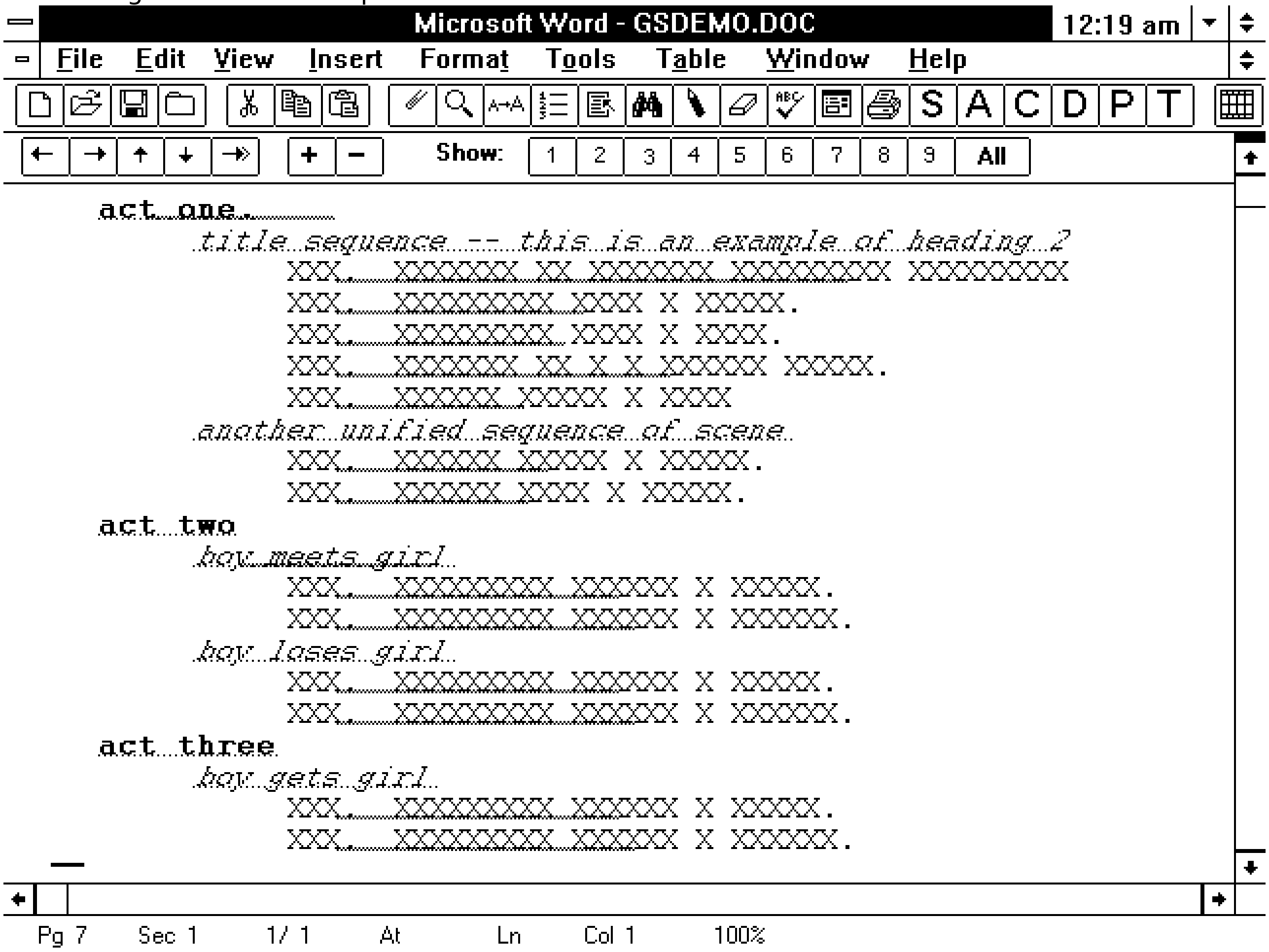
Below is a legend for the ToolBar visible in any document based on GSCRIPT.DOT.

This can be modified if you so desire.



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:



```
act one.
        title sequence -- this is an example of heading 2
                XXX.   XXXXXXX XX XXXXXXX XXXXXXXXX XXXXXXXXX
                XXX.   XXXXXXXXX XXXX X XXXXX.
                XXX.   XXXXXXXXX XXXX X XXXX.
                XXX.   XXXXXXX XX X X XXXXXX XXXXX.
                XXX.   XXXXXX XXXXX X XXXX
        another unified sequence of scene
                XXX.   XXXXXX XXXXX X XXXXX.
                XXX.   XXXXXX XXXX X XXXXX.
act two
        boy meets girl
                XXX.   XXXXXXXXX XXXXXX X XXXXX.
                XXX.   XXXXXXXXX XXXXXX X XXXXXX.
        boy loses girl
                XXX.   XXXXXXXXX XXXXXX X XXXXX.
                XXX.   XXXXXXXXX XXXXXX X XXXXXX.
act three
        boy gets girl
                XXX.   XXXXXXXXX XXXXXX X XXXXX.
                XXX.   XXXXXXXXX XXXXXX X XXXXXX.
```

Pg 7   Sec 1   1/ 1   At   Ln   Col 1   100%

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.

To make using the outline simpler, **GScript** includes a small macro named **ToggleOutline** which is present on the ToolBar as ☰

.   It can also be accessed with the speed key Ctrl-Shift-O.

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 blank by default.   If you add a footer, be careful not to add a multiline footer, or to format the footer with a large font, as this may affect the pagination of the formatted script.

The fact that the header uses "fields" means that you must edit the **FileSummaryInfo** information 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 directly, and change the default header/footer text.   This will affect all subsequent documents based on **GSCRIPT**

or

2) change the header and footer within the active screenplay itself.   This will not affect subsequent documents based on **GSCRIPT.**

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 or cyan.   The reason for this limitation is that 1) **FormatScript** uses these colors to mark paragraphs of action and dialogue that have been broken by formatting and 2) **StripFormatting** uses the colors as a search criterion when it is trying to find and restore those paragraphs that were broken by **FormatScript**.

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.

There are three macros in **GScript** that will aid in using Revision Marks.

RevisionSearch -- looks for the next instance of revised text

RevisionAccept -- accepts the currently selected revised text

RevisionUndo -- undoes the currently selected revised text

These three macros have been installed on the toolbar (See The ToolBar in GScript), along with a button to easily access the main ToolsRevisionMarks macro (that is, the macro that would be accessed if you selected RevisionMarks from the Tools menu).

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.

If you select the entire document and press Ctrl-Shift-F9, you will "unlink fields", replacing the field codes with their result.

You should only unlink {seq slug} fields when the screenplay is complete because once unlinked, the slugline numbers cannot be automatically changed (added to or subtracted from) nor can they be removed automatically.

An unintended, but useful consequence of using fields to number sluglines is that pressing F11 moves from the current point to the next field.   So, if you have only Left numbers inserted (which is the default when you press Alt-N), you will move to the slugline for the next scene.

Advanced Margin Tweaking

This is a dangerous section to read. It tells you more than you may want to know, and should be used with care. You only need to know about this if you have formatted a script and found single line pages and want to try fiddling some numbers to fix the situation rather than fiddling your words.

When **Gscript** formats your screenplay it does so by 1) turning Auto Pagination on, 2) by changing the default margins (which are top one inch, bottom one inch) to something slightly larger (thereby reducing the text area), 3) inserting the appropriate strings where necessary, and 4) resetting the top and bottom margins to something much smaller (thereby increasing the text area to accomodate the added lines).

There is a macro named **Margins** which contains a series of functions (you don't need to understand WordBASIC to follow this, but it may help...) which are set to a series of eight values, representing the start and ending values for Top and Bottom margins for two modes, Master and Shooting.

Here is a listing of the default settings found in **Margins**:

```
Sub MAIN
REM GScript Margin settings
Print "This macro is called by FormatScript to set
margins before and after formatting..."
End Sub
'Start for Master - dialog only
Function tdStart
    tdStart =      0.6
End Function
Function bdStart
    bdStart = 0.7
End Function
'End for Master - dialog only
Function tdEnd
    tdEnd = 0.66
End Function
Function bdEnd
    bdEnd = 0.33
End Function
'Start for Shooting
Function tbStart
    tbStart = 1
End Function
'End for both
Function bbStart
    bbStart = 0.9
End Function
'End for Shooting
Function tbEnd
    tbEnd = 0.75
End Function
Function bbEnd
    bbEnd = 0.3
End Function
```

These numbers are used by **Gscript** to change the margins. For instance, if you are formatting in Master mode, **GScript** will change the 1" top and 1" bottom to the values contained in the functions tdStart and bdStart, that is to .6" top margin and .7 inch bottom margin when it starts, and then change margins again, at the end, to the values contaned in tdEnd and bdEnd, to .66" top maring and .33" inch bottom margin.

You should never have reason to change any of the End settings. If you do, do not change the bottom margin to less than .30".

If, however, you are getting stranded lines, you can try increasing one of the Start settings in increments of .1" or .15" and reformatting.

To to this you would edit the macro named **Margins** by selecting it in the list displayed unter **Macros** on the **Tools** menu.   After changing one of the decimal values, press Ctrl-F4 to close and answer yes when prompted to save changes, and then select **SaveAll** from the **File** menu, answering yes when asked to save GSCRIPT.DOT.

WIN.INI Settings

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

Keyword	Meaning
BCont	Bottom Continued string
DCont	Character Continued String
Iconize	0 = off; 1 = re size 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.
Installed	Simply tells some of the internal macros if **GScript** has been installed.
Mode	0 = Automatic, 1 = Interactive
More	Dialogue More String
NewFile	0 = don't create new file; 1 = save formatted file to a new name
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.
NumTop	0 = Don't count scene length; 1 = Put Scene Page Count after Top Continued (this takes some time).
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)
Side	Used by NumberAllSlugs, NumberThisSlug, NumberOmitteds, and NumberAlphas to determine where (1=left, 2=right, 3=both) to place slugline numbers.
SlugNum	0 = insert scene numbers; 1 = Don't insert scene numbers (useful if you don't have sluglines numbered).
TCont	Top Continued string

Price

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

GScript costs $75.00.   Upgrades from version 1.x cost $15.00.

Any "slipstream" updates, will be free.   That is, until a major upgrade (not expected until version 3.0 of *Word for Windows,* all intermim releaases will be free.

Once you register I will send you a key to disable the single message box reminder which displays when you execute **FormatScript.**

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).

History

Version 1.1 -- first release

Version 1.1a -- bug fix update

Version 2.0 -- upgrade to take advantage of new features in *Word for Windows* 2.0.

Possible additions to future versions

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

Reaching the Author

Electronic Mail Addresses:
Mailing Address
Bugs and Suggestions

Electronic Mail Addresses:

CompuServe ID# - 71171,3555

Internet gg2@columbia.cu

SmartNet Network, Windows Conference

InterLink Network, Windows Conference

WGA BBS #938

Mailing Address

Guy Gallo
219 East 69th Street
New York, NY 10021

Bugs and Suggestions

This release (2.0) is a fairly major revision. Please hammer on the program and report any problems or bugs or suggestions to the above address.

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

About the Author

Lest you think this template is the creation of a *Word for Windows* programmer who knows nothing of screenwriting, I have written screenplays, including *Under The Volcano* (1984), an adaptation of *Adventures of Huckleberry Finn* (PBS 1986), a single episode of *Tales from the Darkside: The Enormous Radio* (1987), and, of course, the large handful of scripts withering in development hell.

I also teach screenwriting at Columbia University.

Putting this together is one of the myriad ways I avoid finishing the novel...

An Auto Macro is a macro that is executed every time a condition is met.   AutoNew is executed every time a new document is based on the template containing the AutoNew Macro.   AutoOpen is executed every time a document based on the template containing AutoOpen is opened.   Other Auto Macros are AutoClose, AutoExec, and AutoExit.   See the user manual or HLP file for more detail.

The directory containing all of your DOT files. By default this is your *Word for Windows* directory. It can be any directory, and if you have moved your templates then you must have a Dot-Path variable set in Win.ini.

Word for Windows allows you to define a style as "hidden".   Text formatted with this style will display on the screen only when ToolsOptionsView is set to either display hidden or ShowAll.   Text formatted as hidden will not print unless you explicitly tell *Word for Windows* do do so.

A style is a collection of indentation, font, and spacing information saved under a specific name.   For instance, "Char" contains the indentation and font information for the character name paragraphs.

A template is a specialized Word for Windows file which contains style information, macros, glossaries, menu, key, and toolbar assignments to be associated with a class of documents (like screenplays).When you create a Word for Windows document you base it on a template. GSCRIPT.DOT is one such template.

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.

These styles are also listed on the Format menu

Note: selecting Para up 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").

You could also make this change permanent by editing GSCRIPT.DOT directly, and altering the text found in either of the table cells.

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 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.