MakeWrite

A MacWrite Document Generator
Version 1.02
27 January 1987

Introduction

MakeWrite is a Macintosh application for producing MacWrite documents from text files. MacWrite can itself read text files, of course, but the resulting document is featureless: all text is Geneva 12-point, plain style. *MakeWrite* allows imbedded commands to be placed in the text for the purpose of controlling formatting of the resulting document. The commands are fairly arbitrary so that you can define your own command set. Commands may be defined to specify any combination of font, point size and type style. A command may change some of these characteristics while leaving others unchanged. For instance, the style may be changed without altering font or size.

Example:
Suppose the markers }B{ and }P{ are defined to signify boldface and plain text, respectively. Then the input text:

    this is a sentence with a }B{boldface}P{ word in it

would be formatted like so:

    this is a sentence with a boldface word in it

Some of this application's characteristics are:

- Creates MacWrite 4.5 documents.
- Allows specification of multiple format changes.
- More natural paragraphing and line joining behavior than is obtained by converting text files with MacWrite.
- Formatting specifications may be saved to a file and reused later. Complicated specifications need not be reentered each time *MakeWrite* is used.
- Fonts to select from may be chosen either from a standard list or from the currently open resource files. The standard list may be edited with a resource editor to reflect personal preference.
- The specifications may be edited with standard cut and paste operations. Undo is

supported.

Motivation

MakeWrite was written to allow text from many sources to be easily incorporated into MacWrite documents. Since most other Macintosh word processors and page layout applications understand MacWrite format, *MakeWrite* allows formatted text to be generated, albeit indirectly, for those programs as well.

Here are some ways *MakeWrite* may be used:

- You can edit your document using your editor of choice on whatever machine you prefer, download it to a Macintosh and run it through *MakeWrite* to create a MacWrite document. The resulting document may then be read into MacWrite for final editing.

- Database report generators are often capable of generating trim between fields in reports. If the trim is set up to contain formatting commands, the report can be used as a MacWrite document source. For example, a database containing names and addresss could be listed like this:

  \para\para\12\boldDonald Duck\para\10\ital123 Elm Street\paraDallas, TX
  \para\para\12\boldFred Flintstone\para\10\italQuarry Ave.\paraBedrock, CO

  By defining \para as a paragraph initiator and \10, \12, \bold and \ital to mean 10-point, 12-point, boldface and italics, respectively, the following formatted output results:

  Donald Duck
  123 Elm Street
  Dallas, TX

  Fred Flintstone
  Quarry Ave.
  Bedrock, CO

  This may look a little complicated, but given that reports like this are typically run from canned scripts, a one-time effort is often sufficient to set up the report generator to produce text suitable for *MakeWrite*.

- If a report generator is not available, a column filter will often do just as well.

- Formatters that do not work on a *what-you-see-is-what-you-get* basis often use imbedded commands. The command set for every such application is different, but since *MakeWrite* allows any marker string to be used, it is often possible to do at least part of the work involved in converting input for that formatter to a form suitable for MacWrite by using *MakeWrite* to simulate the effect of at least some of the other formatter's commands.

  Example:
  The formatter nroff uses .br to force a new output line, which is equivalent to beginning a new paragraph in MacWrite. The nroff commands \fB, \fI and \fR mean

boldface, italic and Roman (plain) text, respectively. Each of these strings can easily be set up in MakeWrite to produce the same effect.

- Even for plain text files with no formatting commands at all, *MakeWrite* can produce better results than converting the text with MacWrite. The latter allows text conversion either by considering each line a separate paragraph or by joining lines to form paragraphs. When lines are joined, you use blank lines to force paragraph breaks. But you get an extra space at the end of each paragraph. That's a small disadvantage; more significantly, MacWrite doesn't join lines with two spaces when the end of a sentence occurs at the end of an input line. Thus you need to go through the document by hand inserting spaces between some sentences in order to obtain normal style.

That's enough hype. No doubt there are bugs, too; these should be reported to the address at the end of this document.

How *MakeWrite* Works

You specify a set of formatting commands, which is a set of lines, each associating a marker string with a format. When the marker string occurs in the input text, it signals a change to the assocated format in the output document. A format consists of a font, a point size and a type style (for example, "Helvetica 12-point bold"). Any or all these format characteristics may be "Same", which means that the characteristic should be left unchanged. If the current format is Helvetica 12-point bold, then a format of "Same/10/Same" would change the size to 10-point while leaving the font and style unchanged.

You also specify paragraphing and line joining behavior. These, together with the formatting commands, are known as the *map*. In other words, the map is the information specifying how marker strings in the input text should be mapped onto formats and paragraphs in the output document. The formatting lines are always displayed in the map window, along with selectors for modifying them (Figure 1). Paragraph and line joining options are selected in a separate dialog.

New map lines are created by selecting New from the Edit menu. The currently selected line, if any, is highlighted. Selectors for changing format values are displayed in the upper part of the window. The selectors are always set to reflect the values of the currently selected line. By clicking in the selectors, these values may be changed. The line also contains a field for the marker string. An editable text box is provided in the map window for changing the marker of the currently selected line. Whatever you type is entered into the marker. You can also use the mouse and the Edit menu to cut and paste text in the marker.

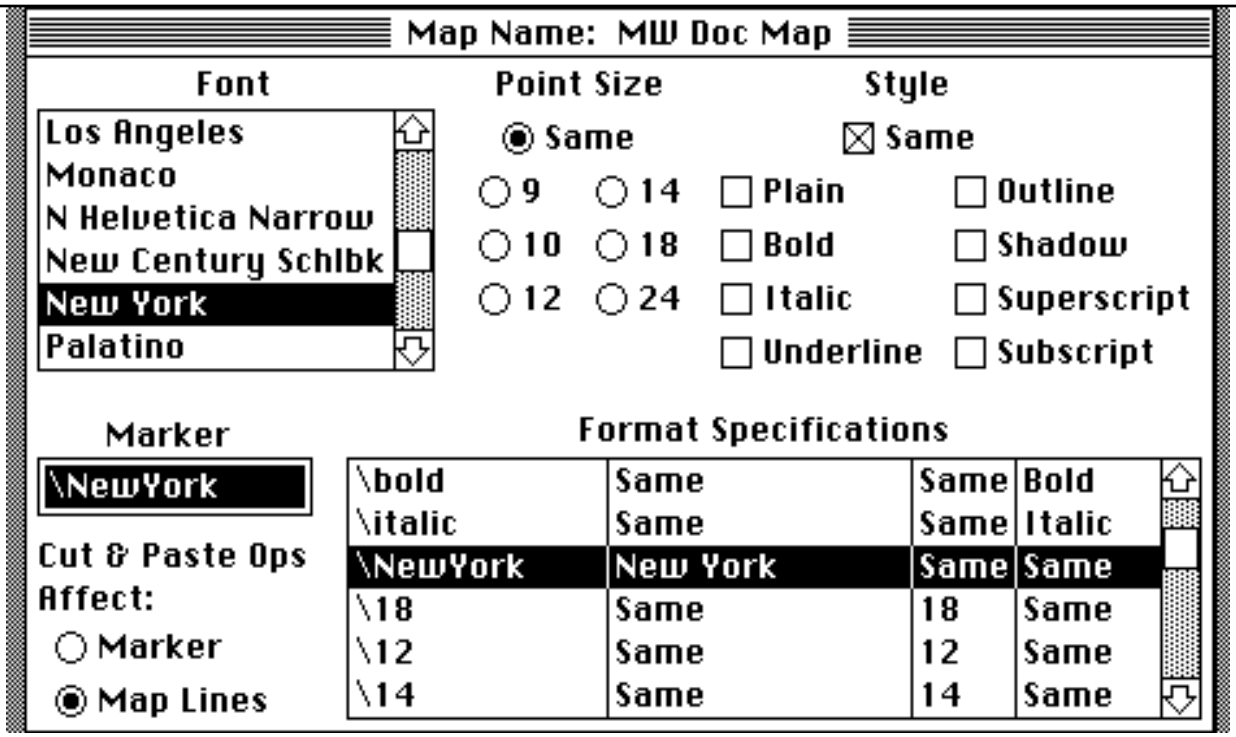

Figure 1. MakeWrite Map Window

When no line is selected, font, size and style indicators are deselected and the editable marker is blank.

To enter format specifications manually, the Edit menu is used to add, duplicate and delete lines from the map. The Edit menu may also be used to add and delete text to and from the editable marker. This leads to an ambiguity for the Edit menu items Undo, Cut, Copy, Paste and Clear (denoted here as the cut and paste operations). For disambiguation, two radio buttons in the lower left of the map window indicate whether the cut and paste operations apply to the marker text or to lines of the map. The cut and paste indicators may also be changed with the mouse to set the cut and paste target. Clicking in the marker or typing in it automatically changes the target to the marker text.

The other Edit menu items (New, Duplicate, Sort, Squish) apply only to map lines, hence are unambiguous.

You can sequence through the map lines by typing *tab*, *enter* or *return*. If the shift key is held down, you sequence through in reverse order. When you get to the first or last line, you wrap around to the last or first line.

Guidelines for Markers

Input text may be said to consist of two parts: content text and marker text. The content text comprises the body of the document. The marker text controls the format of the content text. A good marker cannot be confused with the content text and is easily distinguishable to the human reader. For example, “the” would not be a good marker, because it is virtually guaranteed to be used in the content text, and it does not stand out in any obvious way to a person looking at it. Typically a marker begins (and perhaps ends) with a character that never occurs in the content text (*e.g.*, \ or ~), or uses characters in an unlikely sequence (*e.g.*, }bold{).

Good markers:

\bold\
#italic10#
~courier~

Bad markers:

bold (occurs naturally in "their boldness was surprising")
*10 (occurs naturally in "x = y*10")

times (occurs naturally in "How many times do I have to tell you?")

An arbitrary and capricious twenty character limit on each marker is enforced.

Format and Layout Control

MakeWrite allows complete control over the format of text—font, point size and style. By contrast, very little control over layout on the page is provided. Page breaks cannot be specified. No allowance is made for specifying header or footer text, and in the main document, only one ruler is provided. The ruler is the same as the standard ruler you get when a new document is created in MacWrite (via New from the File menu), except that tabs are set at half-inch intervals. What you do is create a document with *MakeWrite*, then use MacWrite to change the settings and/or add new rulers. (Tip: it's more efficient to add rulers to a document in reverse order—that

is, by going though the document from the last page to the first. For every ruler addition, MacWrite recalculates the layout of all text from the new ruler down to the next ruler (or the end of the document), so less recalculation is incurred by going through backward than forward.)

The only layout control provided is specification of paragraphing and line joining behavior. In a MacWrite document, a text paragraph is a sequence of characters terminated with a carriage return. The paragraph may extend over several lines on the screen or on paper. This definition *may* also apply to text files—such as those created with Macintosh text editors allowing word wrap. However, it does not apply to all text files. Many text editors do not provide word wrap, or, if they do, implement it by inserting carriage returns at the ends of lines automatically as text is entered. For files created by such editors, the definition of a paragraph will be different, and varies according to the stylistic preferences of individual authors. Two common definitions are "all text from one indented line to the next," and "all text between blank lines."

MakeWrite deals with these considerations as follows: A new paragraph is always started when an indented line (a line beginning with spaces or tabs) is encountered in the input. This cannot be overridden. The optional ways of controlling paragraph behavior (none of which need be selected) are:

- You may specify that every line should begin a paragraph, not just indented lines. This preserves the line structure of the input text, and is appropriate for the following kinds of input:

- Text files created with editors allowing word wrap that do not insert carriage returns at the ends of lines automatically
- Program listings
- Forms
- Tables
- Indices
- Outlines, agendas
- Poetry
- Address lists, directories, etc. that should retain the line-by-line structure of the input file

- You may specify that blank lines should be considered to separate paragraphs. When you select this option, the "every line a paragraph" option is turned off automatically.

- You may specify a marker that forces a new paragraph wherever it occurs in the input

text.

If the "every line is a paragraph" option is not selected, input lines are joined together to form paragraphs in the resulting document. (Remember, too, that selecting blank lines as paragraph separators turns off the "every line" option.) If line joining is allowed, you may tell *MakeWrite* whether to try to join lines intelligently or not. Smart joining occurs when a sentence ending at the end of an input line is joined to the next input line with two spaces in between rather than one. For smart joining to occur, you must specify which characters are considered as sentence terminators when they occur at the end of an input line. These are known as the set of sentence "periods." The default set consists of the literal period, question mark and exclamation point (., ?, and !), although you may specify whatever characters you like (for example, a colon doesn't end a sentence, but is sometimes considered to properly require two spaces after).

Some characters should be ignored when looking for a period character at the end of an input line. A quotation or a parenthesized phrase, for example, has a quotation mark or parenthesis after the period, but the end of the line should still be considered the end of a sentence. For this reason, you may optionally specify "quoting" characters that *MakeWrite* ignores at the ends of lines. This is illustrated by the examples below.

The lines in the input text:

> …end of sentence.
> Beginning of next sentence…

should be joined with two spaces between the sentences:

> …end of sentence. Beginning of next sentence…

But so should these inputs:

> …end of quotation."
> Beginning of next sentence…
>
> …end of parenthesized sentence.)
> Beginning of next sentence…
>
> …end of parenthesized quotation.")
> Beginning of next sentence…
>
> …end of quote within quote.'"
> Beginning of next sentence…

which should be joined like so:

> …end of quotation." Beginning of next sentence…
>
> …end of parenthesized sentence.) Beginning of next sentence…
>
> …end of parenthesized quotation.") Beginning of next sentence…
>
> …end of quote within quote.'" Beginning of next sentence…

This may be effected by entering single- and double-quote and right parenthesis into the set of quoting characters. The default set is single- and double-quote (' and "), curly single- and double-quote (’ and ”) and right parenthesis.

In general, the default periods and quotes should give you about what you want most of the time without any special action being taken. At any rate, when you save a map to a file, the line and paragraph control settings are saved with it, so you don’t need to reset

them when the map is read back in later.

Notes

If you do not specify that blank lines are paragraph separators or that every line should begin a paragraph, blank lines in the input are ignored.

If you don't select any kind of paragraphing method and none of your input lines are indented, all lines will be joined together as a single paragraph. Unless your input text is very short, you will quickly run up against MacWrite's limit of 4000 characters per paragraph.

Paragraph Control Examples

The map used in the examples below is:

~plain~ plain text
~under~ underlined text

"Every line" paragraph option off
"Blank line" paragraph option on.
Smart line joining on, with default period and quote sets.

Example of paragraphing by indentation:

Input:

~under~Final Average Earnings~plain~-The average is calculated using your earnings for the three fiscal years (July 1 through June 30) of highest earnings covered under the WRS. If total service is fewer than three years, the average of the total years is used.

~under~Formula Factor~plain~-The formula factor is 1.6 percent.

~under~Creditable Service~plain~-You will receive a full year of creditable service for each fiscal year in which deposits are made for at least 165 days of employment covered by the WRS.

Output:

Final Average Earnings-The average is calculated using your earnings for the three fiscal years (July 1 through June 30) of highest earnings covered under the WRS. If total service is fewer than three years, the average of the total years is used.

Formula Factor-The formula factor is 1.6 percent.

Creditable Service-You will receive a full year of creditable service for each fiscal year in which deposits are made for at least 165 days of employment covered by the WRS.

Example of paragraphing by blank lines:

Input:

~under~Final Average Earnings~plain~-The average is calculated using your earnings for the three fiscal

years (July 1 through June 30) of highest earnings covered under the WRS. If total service is fewer than three years, the average of the total years is used.

~under~Formula Factor~plain~-The formula factor is 1.6 percent.

~under~Creditable Service~plain~-You will receive a full year of creditable service for each fiscal year in which deposits are made for at least 165 days of employment covered by the WRS.

Output:

Final Average Earnings-The average is calculated using your earnings for the three fiscal years (July 1 through June 30) of highest earnings covered under the WRS. If total service is fewer than three years, the average of the total years is used.

Formula Factor-The formula factor is 1.6 percent.

Creditable Service-You will receive a full year of creditable service for each fiscal year in which deposits are made for at least 165 days of employment covered by the WRS.

Some things to note about these examples:

- The examples control paragraphing by indentation *or* blank lines. The two techniques may be combined, of course: begin every paragraph with an indented line, and put a blank line between paragraphs as well.

- Smart line joining only takes place at the ends of lines. For sentences that end in the middle of lines, you put in two spaces as you would normally.

- Determination of whether an input line is indented or not is done by looking at the *first non-marker character* in the line. For instance, the second paragraph in the indentation example above could equally well have been indented as follows:

~under~ Formula Factor~plain~-The formula factor is
1.6 percent.

The result would be the same in each case.

Menu Descriptions

File Menu

New Map
Clears the current map.

Open Map…
Clears the current map and replaces it with the contents of a map file. The file becomes the current map file.

Add Map…
Adds the contents of a map file to the current map. Formatting specifications that duplicate those currently in the map are not added. Only format lines are added; the current paragraphing and line joining settings are not altered.

Save Map
Saves the map to the current map file.

Save Map As…
Asks for a file name and save the map to that file. The file becomes the current map file.

Close

This item is only active when a desk accessory or information display window is in front. It closes the window.

Text->Write…
Generates a MacWrite document from a text file, using the current formatting specifications.

Quit
Exit *MakeWrite*.

Edit Menu

The Edit menu is only active when the map window or a desk accessory window is frontmost. If an accessory is frontmost, only the standard items are enabled. If the map window is frontmost, other items may be enabled.

Cut, Copy, Paste and Clear operate either on the editable text marker or on map lines, depending on the setting of the cut and paste indicators in the map window. All except Copy are undoable. (Undo also applies to changes made to the currently selected line with the controls in the map window, and to changes made to the editable marker.) Cut, Copy and Paste affect the clipboard in the usual way.

Undo
When enabled, undoes the previous Edit menu operation, except Copy. Undo is undoable.

Cut
Deletes the selected line or marker text and places it in the clipboard. The text or line may be put back into the map with Paste.

Copy
Copies the selected line or marker text to the clipboard. The text or line may be put back into the map with Paste.

Paste
If applied to the marker text, replaces the selected text with the text in the clipboard (the text to which Cut or Copy was last applied). Otherwise, if a line is selected, it is replaced with the line in the clipboard (the line to which Cut or Copy was last applied). If no line is selected, the line in the clipboard is added to the end of the map.

Clear
Deletes the selected line or marker text without placing it in the clipboard.

New
Adds a new line to end of the map.

Duplicate
Duplicates the selected line and inserts the duplicate immediately following.

Sort
Orders the map according to the lexicographic (alphabetic) order of the markers.

Squish
Eliminates duplicate map lines.

Special Menu

Standard Fonts
Replaces the current font list in the map window with a list of standard fonts, or adds the standard fonts to the current list.

System File Fonts
Replaces the current font list in the map window with a list of fonts in the System file, or adds the System fonts to the current list.

Paragraph Style…
Presents the following dialog for allowing specification of paragraph initiators and control of line joining behavior (Figure 2).

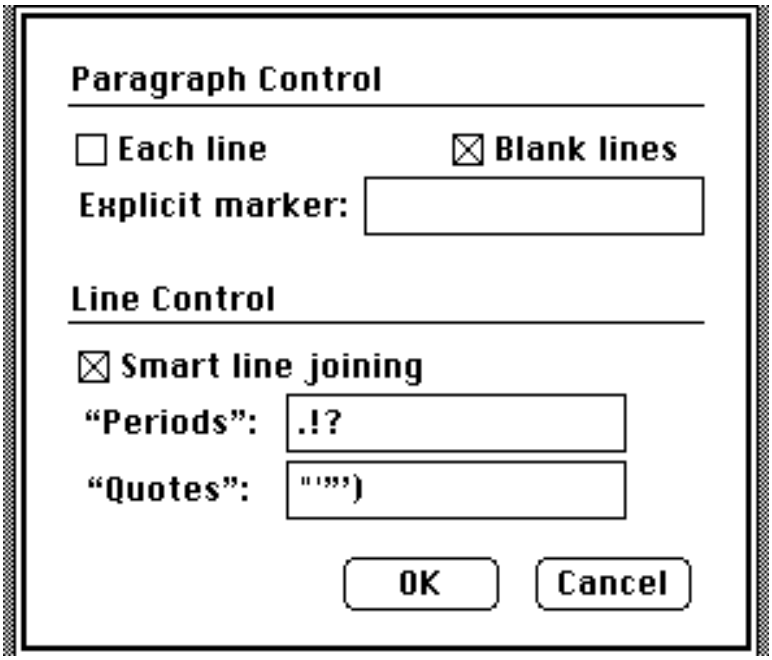

Figure 2. Paragraph Style Dialog

The paragraph control options have the following meaning:

Each line

Each input line is considered a separate paragraph. Use this option when you don't want input lines to be joined together. Selecting this option turns off the Blank lines option.

Blank lines

Blank lines in the input are considered paragraphs and appear as blank lines in the resulting document. If neither this option nor the Each line option is selected, blank lines in the input are

ignored. Use this option to format text that separates paragraphs with blank lines. Selecting this option turns off the Each line option.

Explicit marker
If you enter a marker in the text box to the right of this item, the presence of that marker in the input causes a new paragraph to be started. Use this marker anywhere the paragraphing behavior selected via the other options is inappropriate to produce the desired result.

If none of the above options are selected, no paragraphing at all is done, other than for indented lines.

The line joining control options have the following meaning:

Smart line joining
This option turns on intelligent joining of lines. If it is selected, the set of "periods" must contain at least one character.

"Periods"
The set of characters entered into the text box to the right of this item cause lines to be joined with two spaces rather than one when they occur at the end of an input line to be joined to the next.

"Quotes"
The set of characters entered into the text box to the right of this item are ignored when looking for a period at the ends of an input line that is to be joined to the next.

See the Format and Layout Control section for additional discussion of paragraph and line control.

Find Conflicts
Finds marker conflicts. A conflict occurs when one marker is a prefix of another. The longer marker can never be matched because the shorter one will be recognized first. For example, if you define \under to mean underlined text and \under10 to mean 10-point underlined text, the input:

    a botched \under10format

will be formatted like this:

a botched 10format

This is because as soon as \under is recognized, the format is changed, and *MakeWrite* starts looking for another match starting with 10format.

Get Info
Displays an information window which is a brief summary of this document.

Limitations

There is a limit of 100 fonts in the list of fonts available in the map window. If you have more than that in your System file (does anyone?) and you select System File Fonts from the Special menu, the excess fonts will be ignored.

There is a limit of 100 formatting specifications.

MacWrite has a limit of 4000 characters per paragraph; *MakeWrite* checks for this and gives a warning if the limit is exceeded. MacWrite also has a limit of something over 2000 paragraphs (somewhere around 2048). *MakeWrite* puts a limit of 2000 paragraphs on documents it creates.

Miscellaneous Notes

One of the fonts in the map window font list is Application. The application font is not an actual font *per se*, but a reference to a kind of default font, generally Geneva. You probably don't really want to use it; it's there for completeness.

Taliesin font is also known as Mobile.

MacWrite documents contain carefully constructed information about the position and height of paragraphs and lines in the document (page number, height of each line, etc.). *MakeWrite* does not do this. In general, it can't, because it allows you to format documents for fonts you might not even have; font height information is in such cases unavailable. *MakeWrite* fakes it by writing the bare minimum of information necessary and setting a flag in the document that signals MacWrite to recalculate everything.

Presumably this is inherently dangerous. Indeed, I have seen an instance of very strange looking text after cutting a large section and pasting it back in: each line of pasted text was squished down to about three screen lines where previously it had been about twelve. No doubt other odd things can occur. It is recommended that after you first read into MacWrite a document created by *MakeWrite*, you immediately save it, so that you have a document on disk with more accurate paragraph and line height information.

Distribution Information

MakeWrite is a TransSkel/TransDisplay application written in THINK C. It is public domain and may be distributed without restriction. Comments may be sent to the author at:

U.S. Mail: Paul DuBois
Wisconsin Regional Primate Research Center
1220 Capitol Court
Madison, WI 53715-1299 USA

Internet: dubois@primate.wisc.edu

The version of MakeWrite described in this document is written for Symantec C++/THINK C, which are trademarks of:

Symantec Corporation
10201 Torre Avenue
Cupertino, CA 95014
USA

Acknowledgement

Many thanks to John Lutz for inspiration and incentive to write this program.