DocBreaker

An application to subdivide
text documents
on the Macintosh

Version 1.0 - March 1996

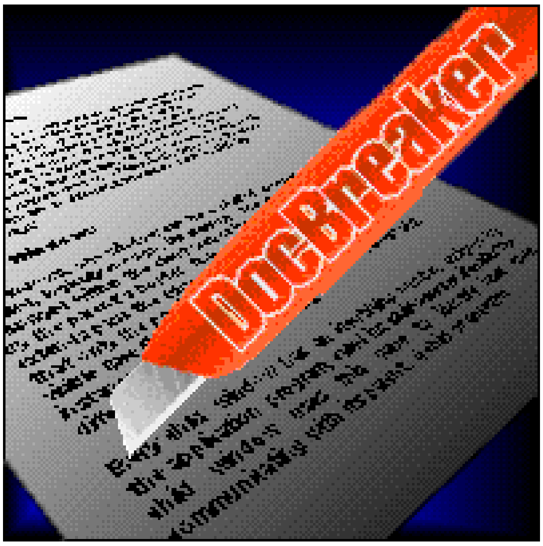


Table of contents

What is DocBreaker?

DocBreaker is an application that breaks text documents in pieces. Its interface is minimal and is based on a single window used both to specify the breaking rules and to control execution.

DocBreaker accept one text document as input and produces one or more text document as output. Both the input and the output documents can be provided in a variety of formats, thanks to Claris XTND System support. DocBreaker will read and write any text document for which an XTND translator is available. This is the same technology used by Claris as well by a number of commercial and shareware products.

The rule followed to break the input document is based on text attributes, including font, size, style, justification and color.
The user first choose the particular attributes combination, then starts the breaking process. DocBreaker starts reading the requested document and when the current text characteristics match those specified, a new file is created. If the specified condition is never encountered, the result file will be a copy of the input file.

Although DocBreaker is a standalone application, it was mainly conceived as a pre-processor for another application I wrote called DocAssembler, which groups and structures document collections and supports the XTND System as well. Togheter, they form a strong document management tool.

DocBreaker supports all the standard Apple Event suites, and is both scriptable and recordable. Besides, it defines a few custom events so that you can completely remote control it using AppleScript. To learn more about this subject, see the "AppleScript reference" chapter and the sample scripts that come with the package.

System requirements

DocAssembler requires a Macintosh with:

- MC68000 processor or better
- 4 Mb RAM
- Color QuickDraw
- Mac OS 7.1 or later
- Claris XTND System

- AppleScript (optional)

The minimum required amount of memory is 550 Kb and should be adequate for most tasks. Complex documents containing pictures will require more generosity.

Shareware information

Amedeo Farello
Loc. Vereytaz, 7
11018 Villeneuve (AO)
ITALY

e-mail: <farello@mbox.vol.it> or <farello@kagi.com>

This software is distributed as SHAREWARE. This means that after a 15 days trial period, if you decide to continue using it, you are required to register your copy by sending the requested fee to the author.

Of course, if you don't do that and you still use the software, it is unlikely that you will ever face a trial. What is likely instead, is that I will stop its development. So, if you find it useful, it could be a nice idea to register.

Remember that shareware is only an alternative distribution method. It simply lets you try the product before you buy. Please, honor this concept and register your copy.

Limitations to unregistered copies

Each time you launch an unregistered copy of DocBreaker, a startup screen is shown recalling the principles of shareware and you can’t continue for at least 10 seconds. Besides, unregistered copies won’t produce more than 5 files from the input document. These limitations won’t prevent you from evaluating this software.

If you register your copy, you will receive a code that fully enables it and removes that annoying startup screen (and your sense of guilt). Registration gives you the legal right to use this application and its possible next versions.

Available licenses

When you register, you can ask for 3 different types of license: “Single User Licenses”, “Site Licenses” or a “World-Wide License”.

A “Single User License” allows the use of the registered copy by any number of people, as long as there is no possibility of it being used simultaneously on more than one machine.

A “Site License” covers all locations for your organization within a 160 kilometer radius of your site (100 miles). One big advantage of a “Site License” is that you do not need to keep track of how many people at your site are using the software.

A “World-Wide License” covers all locations for your organization on the planet earth.

How to register

Paying for my software is fairly simple. Open the **Register** program that accompanies my software. Enter your name, your e-mail or postal address, and the type and number of licenses you desire for each program you wish to purchase. Save or Copy or Print the data from the Register program and send the data and payment to Kagi Shareware. Kagi Shareware handles my payment processing.

If paying with **Credit Card** or **First Virtual**, you can e-mail or fax the data to Kagi Shareware. Their e-mail address is <shareware@kagi.com> and their fax number is +1 510 652-6589. You can either Copy the data from Register and paste into the body of an e-mail message or you can Save the data to a file and you can attach that file to an e-mail message. There is no need to compress the data file, it's already pretty small. If you have a fax modem, just Print the data to the Kagi fax number.

Payments sent via e-mail are processed within 3 to 4 days. You will receive an e-mail acknowledgement when it is processed. Payments sent via fax take up to 10 days and if you provide a correct internet e-mail address you will receive an e-mail acknowledgement.

If you are paying with **Cash** or **USD Check** you should print the data using the Register application and send it to the address shown on the form, which is:

Kagi Shareware
1442-A Walnut Street #392-AF
Berkeley, California 94709-1405
USA

You can pay with a wide variety of cash from different countries but at present if you pay via check, it must be a check drawn in US Dollars. Kagi Shareware cannot accept checks in other currencies, the conversion rate for non-USD checks is around USD 15 per check and that is just not practical.

If you have a purchasing department, you can enter all the data into the Register program and then select Invoice as your payment method. Print three copies of the form and send it to your accounts payable people. You might want to highlight the line that mentions that they must include a copy of the form with their payment.

Kagi Shareware can not invoice your company, you need to act on my behalf and generate the invoice and handle all the paperwork on your end.

Please do not fax or e-mail payment forms that indicate Cash, Check or Invoice as the payment method. As far as we know, there is still no technology to transfer physical objects via fax or e-mail and without the payment, the form cannot be processed.

Payments sent via postal mail take time to reach Kagi Shareware and then up to 10 days for processing. Again, if you include a correct e-mail address, you will hear from Kagi Shareware when the form is processed.

About the registration code

If you entered an e-mail address in the registration form, I will use it to send you the code as soon as I am informed of your payment. If you do not have e-mail, I will have to send you a letter: please understand that this will take more time.

I want your feedback!

I would like to know what you think about my work, so, even if you are not going to register, let me know your impressions, suggestments and bugs discoverings.

Distribution restrictions

This software can be freely distributed as long as it is not modified, the original package is included in its entirety and there's no charge for it. It may not be included in any commercial package without my written consent.

All online services and bulletin boards may make it available to their users at no charge other than the normal connection fees.

All non-profit user groups may distribute it at no charge.

All magazines may publish it on floppy disk without asking me first, as long as I get a copy of the issue containing my software.

All CD-ROM shareware collections and CD-ROM magazines may include it without my prior consent, as long as I get a copy of the CD-ROM.

Software license

PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THIS SOFTWARE. BY USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE.

The software accompanying this document and the related documentation are licensed to you by Amedeo Farello, who retains all rights upon them. The software contains copyrighted material and other proprietary material. In order to protect them, and except as permitted by applicable legislation, you may not decompile, reverse-engineer, disassemble or otherwise reduce the software to a human-perceivable form.

You expressly acknowledge and agree that use of this software is at your sole risk. This software and the related documentation are provided “AS IS” and without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Amedeo Farello does not warrant, guarantee or make any representations regarding the use or the results of the use of the software in terms of its correctness, accuracy, reliability or otherwise. The entire risk about the results and performance of this software is assumed by you.

Under no circumstances, including negligence, shall Amedeo Farello be liable for any consequential, incidental or special damages that result from the use or inability to use the software or related documentation, even if advised of the possibility of such damages.

About the author

Although I have a degree in Architecture, many years ago I discovered that my real passion was to tinker with computers, so I work full time as a software developer now. I started programming on an Apple II in 1982, then switched to MS-DOS in 1985 and finally to Macintosh in 1990.

I currently use the C++ and C languages for all my development projects, but I have some little experience with Pascal and Fortran as well.

I am very interested in Computer Graphics, information storage and rendering methods, human-computer interaction problems and everything related to the software development process in the design and communication fields. Of course, all is Internet related also interests me.

In the past I have written software dealing with:

- geometric modeling (2D-3D curves and surfaces, boolean operations between polygons)
- user interface ("intelligent" support to technical drafting)
- rendering (using flat shade, Gouraud, Phong, ray-tracing and radiosity techniques)
- software localization (resources-to-glossary-to-resources)
- CAD systems integration (parametric generation of models)
- programming utilities (source code collection and formatting)
- structural engineering (linear and non-linear analysis of plane frames and sections, linear analysis of 3D truss structures)

Note: I am currently considering job offers as a Macintosh programmer.

Acknowledgements

I wish to thank Alessandro Levi Montalcini for his intelligence and generosity.

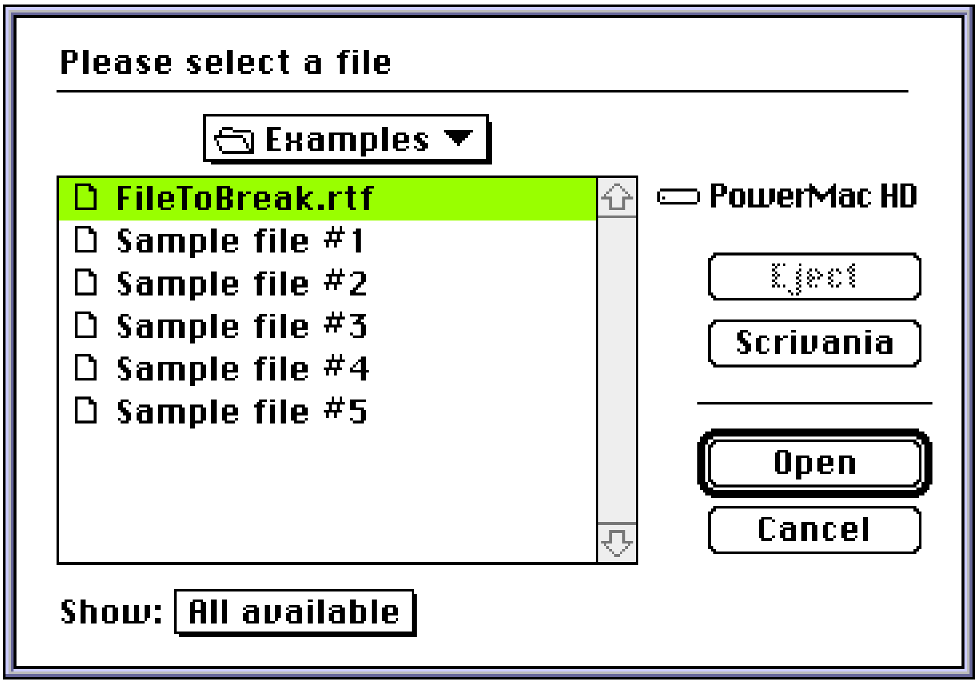
Using DocBreaker

Please be aware that this application cannot function unless the Claris XTND System is installed on your machine. The XTND software is commonly distributed along with many, if not all, Claris software products and many others. Given the XTND System is available and installed, you don’t need to do any particular operation in order to use this program, I just suggest you to copy it onto your hard disk.

When you choose the "Break a File..." command, either from the “File” menu or by pressing the equivalent button in the application window, DocBreaker asks for an input file, then starts reading the requested document and generates a temporary file containing the data rearranged in a way suitable for subsequent processing.

The temporary file is created in the same folder containing the application and is deleted when the translation process ends.

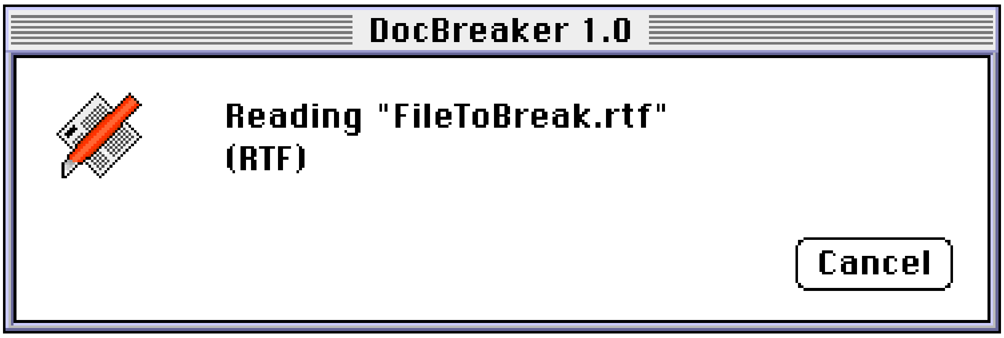
The dialog for input selection



You can open any file for which an XTND translator is available. The popup menu at the bottom of the dialog shows the list of the available translators and is initially set to “All available”. This means that the list will present all the files that could be read. If you want to filter the list, simply select a particular translator with the popup menu.

During the input phase, a window shows the name of the file being read and the translator in use. You can stop the process at any time by pressing the “Cancel” button.

The window shown during the input phase



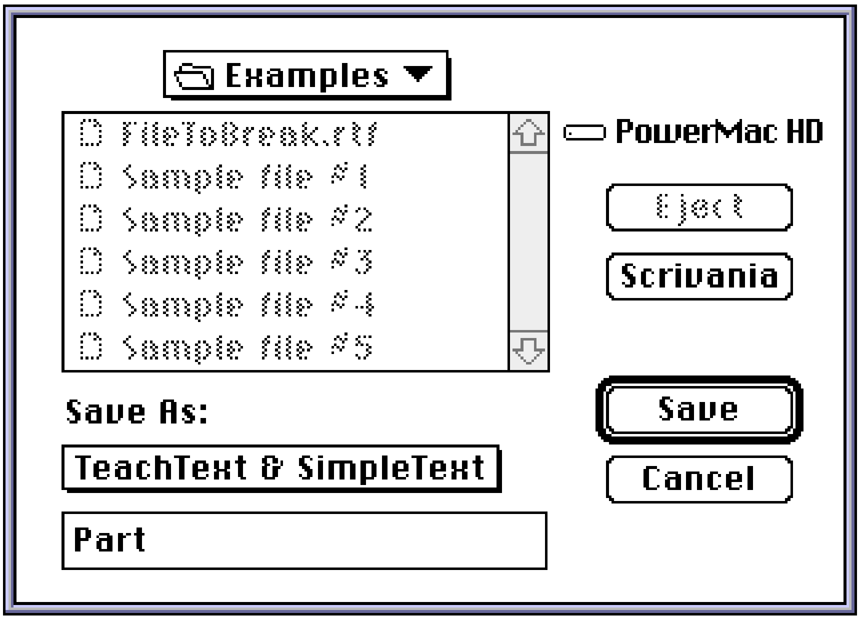
Reading "FileToBreak.rtf"
(RTF)

Each time DocBreaker finds at least one change in the current text attributes of the input file, it verifies if the new attributes exactly match those specified by the user as the “breaking rule”. If it is so, it marks the point.

When input is over, you are requested to specify a “root” name for the group of “subfiles” that are going to be written.

The output files will be named "root.###", where "root" is the name specified by the user in the dialog and "###" is an autogenerated index starting at "001". If DocBreaker cannot find a match with the user specified rules, it will produce a single "root.001" file whose content is equivalent to the input.

The dialog for output selection

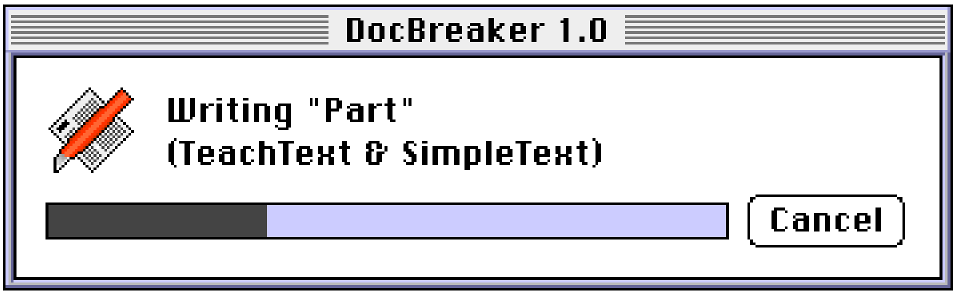


Save As: TeachText & SimpleText

Part

Finally the output phase begins and the subfiles are generated. At this stage, a window shows the “root” name of the files written and the translator in use. You can stop the process at any time by pressing the “Cancel” button.

The window shown during the output phase



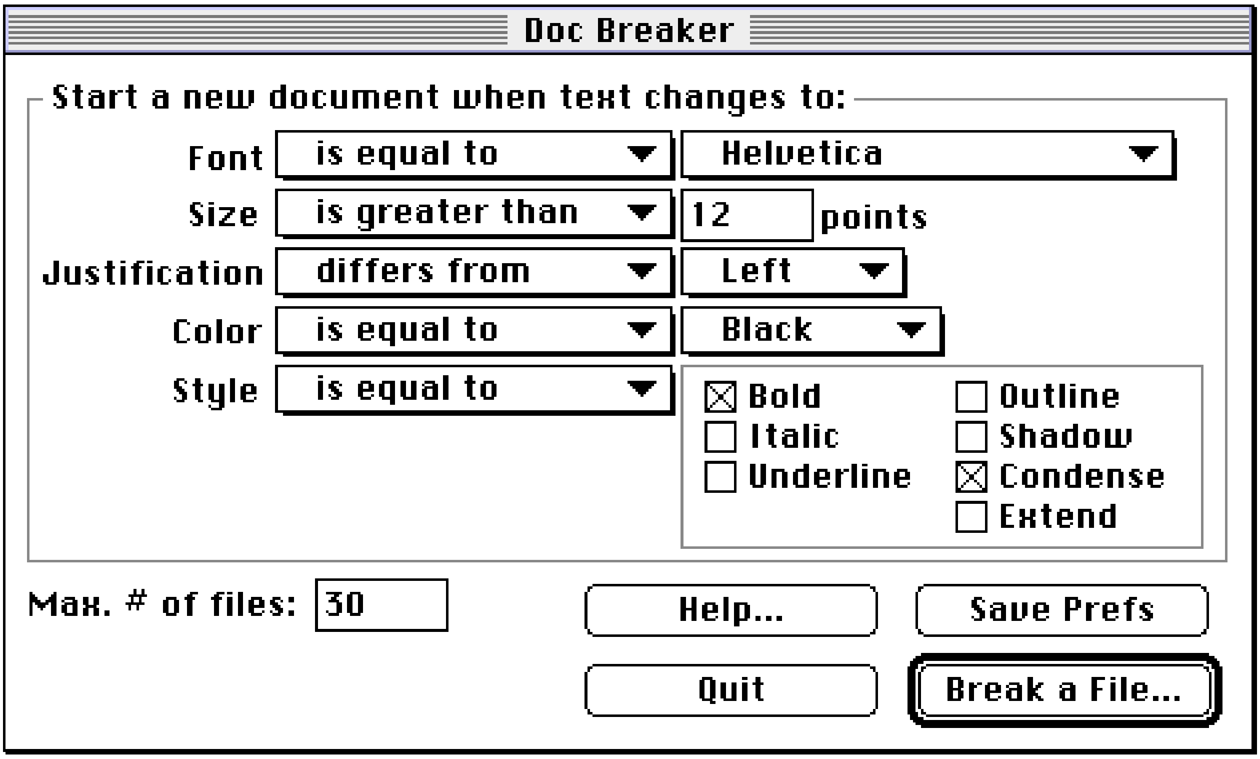
DocBreaker 1.0

Writing "Part"
(TeachText & SimpleText)

The DocBreaker window

DocBreaker is what is called a "dashboard" application, an application that does not create documents and has a single window that allows the user to specify his/her preferences and control the actions of the program. Here follows a description of each item in the window.

The DocBreaker window



Start a new document when text changes to:

Font: is equal to — Helvetica

Size: is greater than — 12 points

Justification: differs from — Left

Color: is equal to — Black

Style: is equal to

☒ Bold
☐ Italic
☐ Underline
☐ Outline
☐ Shadow
☒ Condense
☐ Extend

Max. # of files: 30

The breaking rules

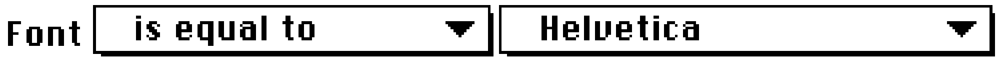
The panel labeled "Start a new document when text changes to:" lets you specify the global rule used by DocBreaker to break an input document. This rule is based on a particular combination of text attributes, including font, size, style, color and justification.

Each rule can also be remotely controlled using AppleScript, see the "AppleScript reference" section to learn more about this.

The "Font" rule

The "Font" rule is controlled by two popup menus. The first menu allows three choices: "ignore", "is equal to" and "differs from".

The “Font” rule popup menus



Font is equal to ▼ Helvetica ▼

• If you choose "ignore", the font family of the text is not taken into account and the next popup menu is unavailable.

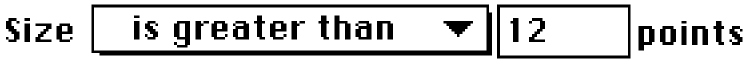
• If you choose "is equal to", DocBreaker will break the file only if the current text font family corresponds to that specified in the second popup menu, where those available in your system are listed.

• If you choose "differs from", it will break the file only if the current text font family is not equal to the selected one.

The "Size" rule

The "Size" rule is controlled by a popup menu and an edit field. The menu lets you choose between: "ignore", "is equal to", "is greater than" and "is smaller than".

The “Size” rule popup menu and edit field



Size is greater than ▼ 12 points

• If you choose "ignore", the font size of the text is not taken into account and the edit field is unavailable.

• If you choose "is equal to", DocBreaker will break the file only if the current text size corresponds to that specified in the edit field, where it can be specified in points.

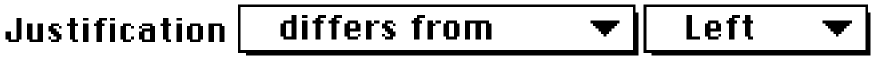
• If you choose "is greater than", it will break the file only if the current text size is greater than the selected one.

• If you choose "is smaller than", it will break the file only if the current text size is smaller than the selected one.

The "Justification" rule

The "Justification" rule is controlled by two popup menus. The first menu allows three choices: "ignore", "is equal to" and "differs from".

The “Justification” rule popup menus



Justification differs from ▼ Left ▼

• If you choose "ignore", the justification of the text is not taken into account and the next popup menu is unavailable.

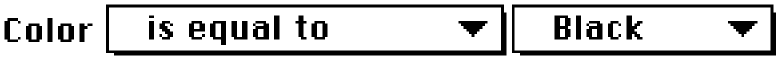
• If you choose "is equal to", DocBreaker will break the file only if the current text justification corresponds to that specified in the second popup menu, where you can choose between left, right, center and full.

• If you choose "differs from", it will break the file only if the current text justification is not equal to the selected one.

The "Color" rule

The "Color" rule is controlled by two popup menus. The first menu allows three choices: "ignore", "is equal to" and "differs from".

The “Color” rule popup menus



Color: is equal to ▾ Black ▾

• If you choose "ignore", the color of the text is not taken into account and the next popup menu is unavailable.

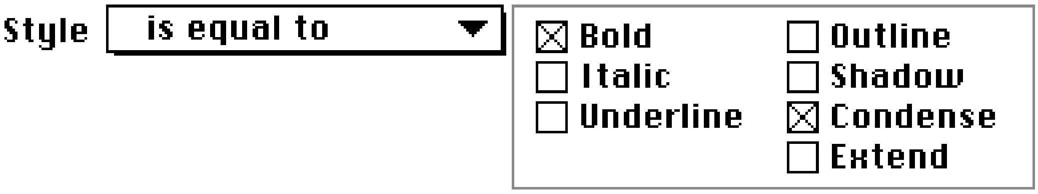
• If you choose "is equal to", DocBreaker will break the file only if the current text color corresponds to that specified in the second popup menu, where you can choose between the 8 standard XTND colors: white, black, red , green, blue, cyan, magenta and yellow.

• If you choose "differs from", it will break the file only if the current text color is not equal to the selected one.

The "Style" rule

The "Style" rule is controlled by a popup menu and a group of check boxes. The menu allows three choices: "ignore", "is equal to" and "differs from".

The “Style” rule popup menus



Style: is equal to ▾

☒ Bold
□ Italic
□ Underline
□ Outline
□ Shadow
☒ Condense
□ Extend

• If you choose "ignore", the style of the text is not taken into account and the check boxes are unavailable.

• If you choose "is equal to", DocBreaker will break the file only if the current text style attributes correspond to those specified with the checkboxes, with whom you can control the usual bold, italic, underline, outline, shadow, condense and extend properties.

• If you choose "differs from", it will break the file only if the current text style attributes combination is not equal to the selected one.

Other controls

The "Max. # of files" field

This field accepts the maximum number of output files you are willing to consent DocBreaker to generate. This is a useful precaution against not well thought out breaking rules, which can virtually generate thousands of files!

If, during the reading phase, DocBreaker should find out that the number of output files would exceed the one allowed, it would tell you and stop the operation.

The "Break a File..." button

This button starts the breaking process by asking you to choose the input document file. After the reading phase, you will be asked again to choose the "root" name for the output file series. You will be able to cancel the operation at any moment.

The "Save Prefs" button

If pressed, this button will record the current settings in a preferences file in the preferences folder of your system folder. This file will then be used each time you open the DocBreaker application to initialize its preferences. To revert to the default initialization, just delete the preferences file.

The "Quit" button

This quits DocBreaker.

Known bugs and problems

XTND limitations (nice pun, huh?)

Although the Claris XTND System is based upon a real powerful concept and deserves to be further developed, its implementation suffers from several shortcomings.

Obviously, if one formatting feature doesn't have its counterpart into the XTND architecture, it cannot be translated properly. This architecture is beginning to show its age: it knows nothing about features that have become almost standard today, like tables or borders; color management is crude at best, since all is mapped to the 8 standard XTND colors (white, black, red, green, blue, cyan, magenta and yellow), so you have to say goodbye to that delicate nuance that you liked so much...

Other problems occur due to the poor quality of the translators. I did several, I repeat SEVERAL, tests to discover that none of the translator available to me was capable to preserve all the characteristics of the original document.

Finally, although I was using the same set of translators, with some translator combinations I have obtained damaged files after an import-export cycle, where ClarisWorks has never shown such a behaviour. While is likely that the Claris development team is better than me at programming, I suspect they also have access to better documentation about XTND.

As you probably guessed, all this problems can drastically reduce the quality of the documents produced by this application. If you have read this far without giving up, I believe the best strategy is to experiment with the formatting of the original documents until a satisfying result is produced from the import-export translator combination. By the way, don't assume that the same translator will behave the same way both during import and export!

All that said, I know that have probably contributed with some exquisite bugs of my own. If you think you have discovered one or more of them, please let me know, even if you are not going to register.

Footnotes

Sorry, but I decided to skip over footnotes, because I could not figure out an easy way to manage them using the XTND System with multiple files. I hope you would not consider this a major flaw.

AppleScript

Currently, errors encountered during the execution of a script are not managed properly, that is, the caller does not receive error information back. I hope to fix this soon.

Commands reference

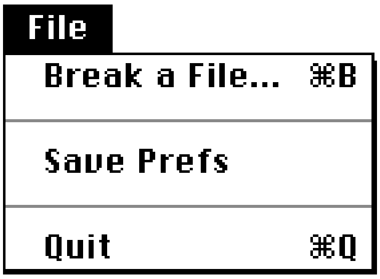
The apple menu

This menu contains a command to obtain informations about the DocBreaker application

About DocBreaker...

Shows a window that gives informations about the author, the version number, the name of the registered user... (If you use this program, please register!).

The File menu



This menu contains the file management commands.

Break a File...

Starts the break session, letting the user choose the source document, then the format and the root name for the output document file series. This command is always available and has a button equivalent in the DocBreaker window.

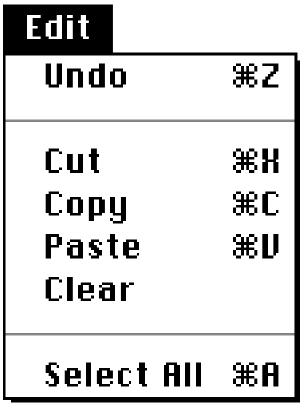
Save Prefs

Records the current settings in a preferences file in the preferences folder of your system folder. This file will then be used each time you open the DocBreaker application to initialize its preferences. To revert to the default initialization, just delete the preferences file. This command is always available and has a button equivalent in the DocBreaker window.

Quit

Quits DocBreaker. If you have any open document whose contents aren't already saved, you have a chance to do that. This command is always available and has a button equivalent in the DocBreaker window.

The Edit menu



This menu contains the standard edit commands.

Undo

Undoes the last text editing operation in the DocBreaker window. Its availability and name change according to the context.

Cut

Clears the current text selection and copies it to the clipboard. This command is only available when a selection exists.

Copy

Copies the current text selection to the clipboard. This command is only available when a selection exists.

Paste

Copies the contents of the clipboard at the current insertion point. This command is only available when the clipboard's contents are appropriate.

Clear

Clears the current text selection. This command is only available when a selection exists.

Select All

Well... it selects all. This command is only available when there is text to select.

Apple Events reference

This section describes the Apple Event support in DocBreaker. It will explain each Apple Event which can be sent to the application. Description for the required suite is omitted, you can assume the standard behaviour.

The DocBreaker Suite

Break

Breaks a document

Event Class: kAE_DB_Suite
Event ID: kAE_DB_BreakFile

keyDirectObject:

Descriptor Type: typeNull
Required / Optional: optional

Additional Parameters:

the -- specifies the file to break

Descriptor Keyword: keyAEFile
Descriptor Type: typeAlias
Required / Optional: required

into -- specifies the root file

Descriptor Keyword: keyAEFile
Descriptor Type: keyAE_DB_RootFile
Required / Optional: required

using translator -- the translator to use

Descriptor Keyword: keyAE_DB_TransName
Descriptor Type: typeChar
Required / Optional: required

with format -- the requested format

Descriptor Keyword: keyAEFileType
Descriptor Type: typeType
Required / Optional: required

Reply Parameters:

«none»

AppleScript example:

```
tell application "DocBreaker 1.0"
                break the file "sample.rtf" into file "part.rtf" using translator "RTF" with
format "TEXT"
end tell
```

Set Preference

Chooses options for DocBreaker

Event Class: kAE_DB_Suite
Event ID: kAE_DB_SetPrefs

keyDirectObject:

Descriptor Type: typeNull
Required / Optional: optional

Additional Parameters:

max files to -- set the maximum allowed number of files

Descriptor Keyword: keyAE_DB_MaxFiles
Descriptor Type: typeShortInteger
Required / Optional: optional

font rule to -- the font rule

Descriptor Keyword: keyAE_DB_FontRule
Descriptor Type: typeEnumeration
Required / Optional: optional

font to -- the name of the font

Descriptor Keyword: keyAE_DB_Font
Descriptor Type: typeChar
Required / Optional: optional

size rule to -- the size rule

Descriptor Keyword: keyAE_DB_SizeRule
Descriptor Type: typeEnumeration
Required / Optional: optional

size to -- the size of the text in points

Descriptor Keyword: keyAE_DB_Size
Descriptor Type: typeShortInteger
Required / Optional: optional

style rule to -- the style rule

Descriptor Keyword: keyAE_DB_StyleRule
Descriptor Type: typeEnumeration
Required / Optional: optional

style to -- the text style attribute

Descriptor Keyword: keyAE_DB_Style
Descriptor Type: typeEnumeration
Required / Optional: optional

justification rule to -- the justification rule

Descriptor Keyword: keyAE_DB_JustRule
Descriptor Type: typeEnumeration
Required / Optional: optional

justification to -- the text justification

Descriptor Keyword: keyAE_DB_Just
Descriptor Type: typeEnumeration
Required / Optional: optional

color rule to -- the color rule

Descriptor Keyword: keyAE_DB_ColorRule
Descriptor Type: typeEnumeration
Required / Optional: optional

color to -- the text color

Descriptor Keyword: keyAE_DB_Color
Descriptor Type: typeEnumeration
Required / Optional: optional

Reply Parameters:

«none»

AppleScript example:

```
tell application "DocBreaker 1.0"
            set preference max files to 5
            set preference font rule to is_equal_to
            set preference font to "Courier"
            set preference size rule to is_greater_than
            set preference size to 14
            set preference style rule to is_equal_to
            set preference style to bold
            set preference justification rule to differs_from
            set preference justification to center
            set preference color rule to is_equal_to
            set preference color to red
end tell
```

Constants and AppleScript enumeration terminology

```
kAE_DB_Suite            = 'Fbrk'

kAE_DB_BreakFile        = 'Brek'
kAE_DB_SetPrefs         = 'Pref'

keyAE_DB_TransName      = 'Tran'
keyAE_DB_RootFile       = 'Root'
```

```
keyAE_DB_MaxFiles       = 'PRF0'
keyAE_DB_FontRule       = 'CND1'
keyAE_DB_Font           = 'PRF1'
keyAE_DB_SizeRule       = 'CND2'
keyAE_DB_Size           = 'PRF2'
keyAE_DB_StyleRule      = 'CND3'
keyAE_DB_Style          = 'PRF3'
keyAE_DB_JustRule       = 'CND4'
keyAE_DB_Just           = 'PRF4'
keyAE_DB_ColorRule      = 'CND5'
keyAE_DB_Color          = 'PRF5'

enumFilterDontCare      = 'null' • ignore
enumFilterEqual         = 'equa' • is_equal_to
enumFilterNotEqual      = 'noeq' • differs_from
enumFilterGreater       = 'gret' • is_greater_than
enumFilterSmaller       = 'smal' • is_smaller_than

enumStylePlain          = 'plan' • plain
enumStyleBold           = 'bold' • bold
enumStyleItalic         = 'ital' • italic
enumStyleUnderline      = 'undl' • underline
enumStyleOutline        = 'outl' • outline
enumStyleShadow         = 'shad' • shadow
enumStyleCondense       = 'cond' • condense
enumStyleExtend         = 'exte' • extend

enumJustLeft            = 'jusL' • left
enumJustCenter          = 'jusC' • center
enumJustRight           = 'jusR' • right
enumJustFull            = 'jusF' • full

enumColorWhite          = 'colW' • white
enumColorBlack          = 'colN' • black
enumColorRed            = 'colR' • red
enumColorGreen          = 'colG' • green
enumColorBlue           = 'colB' • blue
enumColorCyan           = 'colC' • cyan
enumColorMagenta        = 'colM' • magenta
enumColorYellow         = 'colY' • yellow
```

Version history

1.0 (March 1996)

First public release.