Mizer™ 1.8 Manual

ANTIMONY SOFTWARE
http://www.antimonysoftware.com
support@antimonysoftware.com
1-662-263-5412 (fax)

CONTENTS

WHY USE MIZER™?

If you aren't using **Mizer™**, your pages aren't as fast as they could be. They aren't downloading as fast as they could be, and they aren't rendering as quickly as they could be. Speed is crucial for Web pages, as most professionals are well aware.

The faster your page is readable by your visitors, the more will stick around to read what you have to say, and the better their impression will be of your site and your company and product or your client's site and product.

Everyone in business knows that more exposure with better first impressions translates to a lot more money generated from the marketing effort. That includes the first impression of all your clients if you design sites for others. Faster is better for business, and since **Mizer™** can make all your pages faster, it will improve your business.

That's why you should use **Mizer™**, the fact that if everyone used **Mizer™** on their Web pages, it would save more than 27 billion bytes an hour in traffic is a really nice thought, but that isn't why you want to use it.

HOW DOES MIZER™ DO IT?

Mizer™ automatically removes unneeded information in HTML and JavaScript syntax making it smaller and more easily parseable by browsers. Depending on how your pages were made there is from 5–50 % useless or redundant information in them. The most basic formatting for sake of readability typically introduces 5–10 % of information as spaces, tabs, and returns that a browser must parse through and ignore. That information has a purpose—to make your syntax easily readable for authoring—but it is useless, slow fat for servers and browsers. The other forms of unnecessary information are actual syntax elements that are either unneeded as the case with many quotes around attribute values, many comments, many tag attributes and also many tags. These get introduced in two ways. You put them through habit or misunderstanding of what was actually needed, or your WYSIWIG authoring tool did for whatever reasons it has.

There are two ways you can fix that problem. The first and much harder, is you can reread all the how-to information available about HTML, sorting out the good from the bad, until you can write pages by hand, provided you can take the eye strain involved, since you can't format it so you can read it anymore to do this, that are as lean and fast as possible.

Then again you could just use **Mizer™** and forget about it.

It won't matter if you go with choice number one or choice number two, but as long as you do, a domino effect will start that will result in your pages being much, much faster.

1: Since the source file is smaller, the server reads and sends it faster so the browser starts receiving the information sooner.

2: The browser is able to parse the arriving data much faster because all unnecessary data has been eliminated. This allows the browser to make http requests for images faster and allows text information to the rendered much more quickly by the browser. This helps hold attention spans of viewers because text elements appear almost immediately. As has been described by **Mizer™** users, the text "snaps" onto the page.

3: Since the browser was able to render the text portions of the page more quickly, rendering images is also faster because there is less conflict for cpu power between rendering the text and rendering the images. Images also arrive sooner to be rendered sooner because they were requested sooner.

4: Because everything involved in retrieving a page is speed up, overall server performance and connection performance is increased as well. Page retrievals take less time to complete so the average number of concurrent connections is reduced meaning more server cpu time and more connection bandwidth are available to satisfy each individual page request.

What **Mizer™** does is much more than just making HTML files a little smaller, as you can see now. Compared to optimizing images properly, which would cut much more raw size off your page than that can from your source with **Mizer™** - at least for a graphics intensive page - it may seem trivial to knock a k or two more off, but when you realize how and how much those particular bytes make a difference in overall page performance, the need for **Mizer™** is clear.

Here's an example below.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<HTML>

     <HEAD>
          <TITLE>Why use Mizer?</TITLE>
     </HEAD>

     <BODY BGCOLOR="#E6E6E6">

     <P>Because while this nicely formatted source is easy to read
     and pretty, it's bloody inefficient for the browser.</P>

     </BODY>

</HTML>
```

The above source is 323 bytes, but only 224 bytes are important. Eliminating the other 99 bytes, more than 30 % of the size, will not affect the way it renders at all.

It will only make it download in about 30 % less time and render in a browser upwards of 50 % faster overall. The cumulative effect of optimized HTML is much greater than the savings in file size alone.

The bottom line, whether **Mizer™** saved .5 % or 50 % is your page will be seen by your viewers faster, and as the maxim goes, "Fast enough never is."

Also, if you pay for bandwidth, **Mizer™** will pay for itself. Saving a few more K here and there adds up quickly to megabytes and gigabytes on popular sites. Taking the very popular and cool Version Tracker site as an example, using **Mizer™** on just one page would save over 650MB per day of bandwidth. (Version Tracker can be found at http://www.versiontracker.com and comes highly recommended as a great Macintosh resource.)

On top of that, **Mizer™** can save you quite a bit of time. With **Mizer™** you don't have to worry about how neatly or carefully you format your HTML syntax. You can space it out and arrange your source for easiest authoring, without any worry at all about page performance.

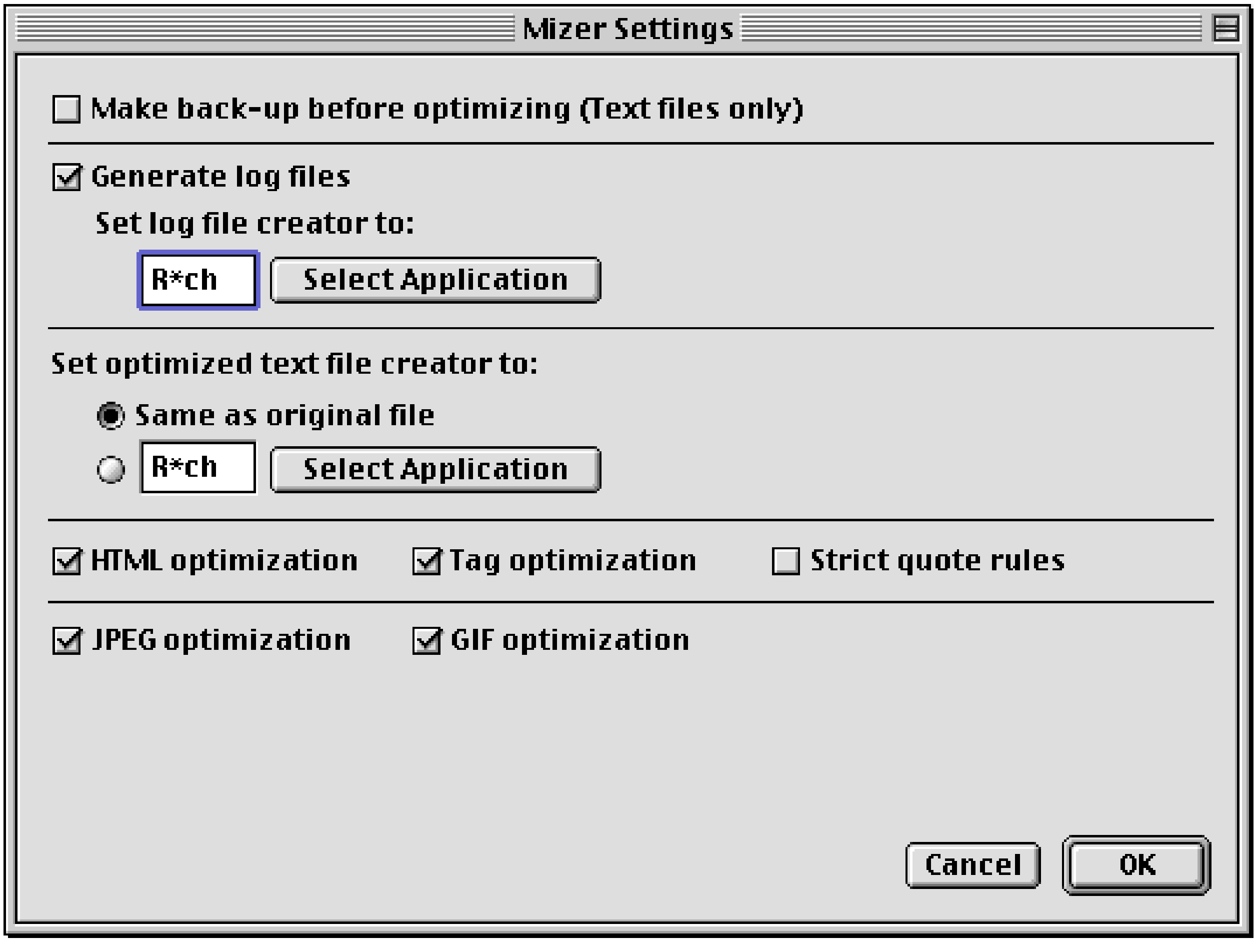
HOW EASY IS IT TO USE MIZER™?

It's as simple as Dragging & Dropping an HTML file or a Folder of HTML files onto **Mizer™** after you have finished your page or site.

Mizer™ automatically processes all HTML files and then you upload them to you server. If you use the included "Optimize & Upload Site w/Fetch" Applet the entire process is reduced to selecting your main site folder and clicking OK. Your optimized pages are uploaded to your server and your original pages are maintained intact locally for future editing.

Besides standard dialogs to select files and folders to process when you aren't using **Mizer™** as a Drag & Drop application the only dialog **Mizer™** has is a simple preferences dialog.

SETTING MIZER™'S PREFERENCES



☐ Make back-up before optimizing (Text files only)

☑ Generate log files

Set log file creator to: R*ch

Set optimized text file creator to:

● Same as original file

○ R*ch

☑ HTML optimization ☑ Tag optimization ☐ Strict quote rules

☑ JPEG optimization ☑ GIF optimization

1: Make Backup Before Optimizing

The backup feature in **Mizer™** IS NOT intended for general usage. It is a safety feature in the frequent case when people jump straight in before reading the documentation, and as such is enabled by default when you first use **Mizer™**.

The backup files created by **Mizer™** will always be located in the same Folder as the original files and have the file extension 'bac' instead of the original extension.

The HTML optimization process destroys original syntax formatting making the resulting file difficult to edit by hand.

JPEG and GIF optimization removes application specific information in files, which may cause certain settings to be lost when the images are reopened to edit, but will not effect the saved image data at all.

It is important to maintain an editing copy of your site locally when using **Mizer™**. You should wait until your files are ready to upload and then make a copy of them manually (Command D in the Finder will duplicate a file or folder) to perform optimization on. The optimized copies should be uploaded, then deleted locally leaving your original editing files on your hard drive, and the optimized published files on your server.

Once you are following this procedure it is best to turn off the backups setting to keep extra files from being created.

2: Generate Log Files

Mizer™ can keep a central log file detailing the results of files that it processes. This log is simply easy-to-read information on file sizes. There is no reason to ever turn logging off. It is unobtrusive and in the background even if you don't often examine the log file.

Examining files optimized by **Mizer™** in the Finder will usually show a much greater savings in file size than the log reports. This is because **Mizer™** also strips all resource forks from files and only reports actual savings in HTML size in its log file. Resource sizes aren't added into the total.

Resource forks are stripped for convenience so the Finder will report accurate sizes for your files, and to reduce the incidence of accidental file corruption from FTP clients performing automatic MacBinary encoding to preserve the resource fork of uploaded Macintosh files.

3: Set Log File Creator To:

You can set the application to open **Mizer™**'s log files by setting the file creator code. There are two ways to do this. If you know the creator code for the application, such as BBEdit, that you would like to open **Mizer™**'s log files, you can type it in the text box. For example the 4-byte creator code for BBEdit is 'R*ch'.

If you don't know the 4-byte creator code you can hit the "Select Application" button and a dialog will be shown allowing you to choose an application to open **Mizer™**'s log files with.

4: Set Optimized File Creator To:

There are two ways **Mizer™** can handle the file creators for optimized HTML files. (This only applies to optimized source files, Mizer will not change the creator code of image files.)

The "Same as original file" option will result in optimized source files belonging to the same application as the original files. The creator code will not be changed with this option.

The other choice sets all optimized files to belong to a predetermined application by assigning them the creator code for that application.

If you don't know the four byte creator code you can hit the "Select Application" button and a dialog will be shown allowing you to chose an application to use.

5: HTML Optimization

The HTML optimization check box enables **Mizer™**'s HTML optimizer. When checked, **Mizer™** will process and optimize HTML (and JavaScript) source files.

6: Tag Optimization

This enables tag level optimization that removes optional closing tags where they will not change the appearance of a page. The closing tags that are currently removed when tag optimization is enabled are shown below.

Tag optimization is now controlled by the contents of STR# resource 257 which contains a list of all HTML tags to be removed when this setting is enabled. You can configure which tags will be removed by tag optimization by editing this resource in ResEdit, a freeware resource editor available from Apple.

The default tags removed by the tag optimization setting are

```
<X-CLARIS-TAGVIEW
</DD
</DT
</LI
</OPTION
</COLGROUP
<X-SAS-WINDOW
</HEAD
</HTML
</BODY
```

Individual tags can be removed or others added as needed. When adding tags you should include only the opening part of the tag, as above, and not add the closing '>' to the string.

The corresponding setting interface for direct configuration of tag optimization within the application will be added in the next update. This is a tip to make use of a currently hidden feature.

7: Strict Quote Rules

When this setting is enabled **Mizer™** will strictly follow the letter of the HTML specification when determining if quotes around attribute values should be removed or not.

Generally, you will want to let **Mizer™** use normal liberal quote rules that determine if quotes can be removed from an attribute value based on actual browser behavior instead, as savings will be much greater without compatibility issues.

However, due to popular request, the option of strict quotes is available now.

8: JPEG Optimization

This setting enables the new JPEG optimization.

This is a lossless process that removes all unneeded information from the JPEG without changing its compression. It has been tested with all current browsers, and most old browser versions to be fully compatible with browsers supporting in-line JPEG display.

It has also been tested with all versions of Photoshop with JPEG support to be safe, but this process may interfere with some applications reading JPEG files causing them not to believe the file is a valid JPEG because a common marker used by many applications to test file type will be removed. This does not make the JPEG invalid, but it may make some applications that check file type based on this marker to not recognize the file as a JPEG, and therefore should not be performed on JPEG files that may be exchanged with others for editing.

9: GIF Optimization

This setting enables the new GIF optimization.

Like JPEG optimization, GIF optimization is also a lossless process. It will not prevent any applications from continuing to edit GIF files optimized, and poses no potential problems to editing applications in file recognition.

SOME HIDDEN FEATURES IN VERSION 1.8

File filtering in this version is completely changed to be more easily configurable, though the corresponding interface has not been added. File extensions that are recognized as HTML files are now contained in STR# resource 256. You can add or remove extensions now using ResEdit or Resorcerer. File extensions recognized for other file types are also controlled this way, all STR# resources that are user-editable are clearly labeled with their purpose.

HTML comment removal was added as of version 1.5, and the strings that are used to identify comments that are server side includes or other preprocessing directives are also contained in a STR# resource, which may be edited using ResEdit or Resorcerer.

Tag optimization, previously described in the preferences section, is also controlled in a like manner.

A future update will complete the interface to allow configuration of these features from an application dialog, but you can still make use of the functionality now.

OPTIMIZING HTML FILES AND FOLDERS WITH MIZER™

Mizer™ recognizes HTML files by their file extensions. There are seven extensions that it recognizes.

```
.html
.htm
.shtml
.txt
.cfm
.incl
.lasso
```

Mizer™ will scan and attempt to optimize the HTML syntax in any file ending in one of those three extensions (.html, .htm, or .shtml) that you open with **Mizer™** or Drag & Drop onto it.

For Drag & Drop use, simply drag the file onto **Mizer™**. The application will launch, optimize the file, and perform any other actions that are set in **Mizer™**'s preferences such as creating a backup file and setting the optimized file's creator type.

You should not use **Mizer™** on your HTML file until you are done with creating and testing it. **Mizer™** is the last step before Web publishing, and not a tool to employ before your HTML is in final form.

If there is any possibility that you HTML may need to be revised in the future you should make sure to retain a local backup copy for editing. Optimized HTML files are very difficult to read because the process of optimization removes formatting for human readability.

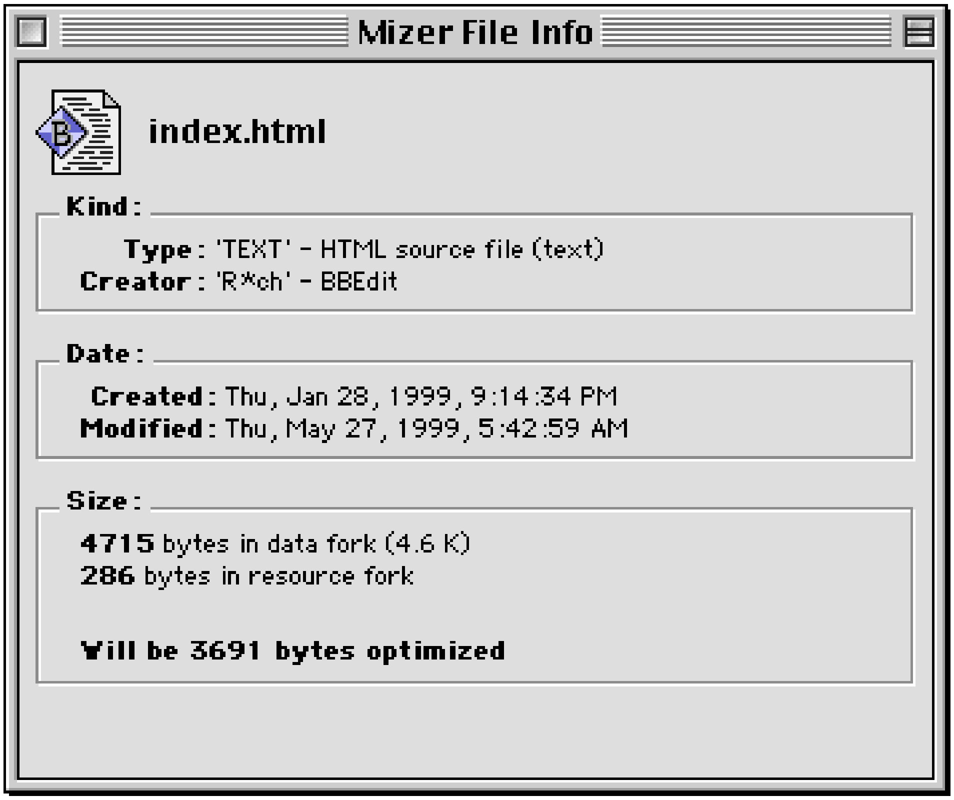
The best approach is to maintain a local copy of your entire site for editing. When you make changes, copy the changed files, optimize the copies and upload the optimized files to you server preserving the integrity of your local HTML for future editing.

Mizer™'s automatic backup capability is provided only for safety, and is not necessarily the best approach for good organization of your files.

USING THE GET FILE INFO... COMMAND IN MIZER™

Mizer 1.6 added the Get File Info command to the edit menu. With this you can view information about optimized and unoptimized files.

This command (File -> Get File Info... or Command - I) only displays information. It will not change files in any way.



index.html

Kind :

Type : 'TEXT' - HTML source file (text)
Creator : 'R*ch' - BBEdit

Date :

Created : Thu, Jan 28, 1999, 9:14:34 PM
Modified : Thu, May 27, 1999, 5:42:59 AM

Size :

4715 bytes in data fork (4.6 K)
286 bytes in resource fork

Will be 3691 bytes optimized

APPLE SCRIPTING MIZER™

Mizer™ is fully and easily Apple Scriptable. Like most Drag & Drop applications it only supports the Required Apple Events Suite and processing is initiated be sending **Mizer™** an open Apple Event.

A simple Apple Script to do that follows.

```
-- Very simply scripting example for Mizer 1.0

property gSourceFolder : ""

-- Uses the finder to show a dialog to select a source
-- folder to optimize

on run
     tell application "Finder"
          try
               set folderAlias to choose folder with prompt
                    "Select the folder to optimize:"
               set gSourceFolder to folderAlias
          on error -- e.g. user cancelled
               return
          end try
     end tell

     -- Mizer is a Drag & Drop application, so opening
     -- a file or folder will initiate all processing and
     -- optimization of files. Scripting it is as easy as it gets.

     tell application "Mizer 1.0"
          open gSourceFolder
          quit
     end tell
end run
```

You can easily incorporate **Mizer™** into more complex scripts utilizing many applications. A sample script "Optimize & Upload w/Fetch" is included that demonstrates the potential for scripting **Mizer™** to work with other applications.

It fully automates the entire process of backing up, optimizing, and uploading a Web site. (Take note that your destination URLs need to be in the form of ftp://login:password@host/path/ for the script. See the script source for detailed comments.)

WHAT MIZER DOESN'T DO

Mizer™ does a lot to optimize your HTML files, but there are things that could be done to make them even smaller that it doesn't.

Those things require an element of human judgment. Should you wish to make your files as small as possible you should also

1: Keep Meta tag and other nondisplaying information to the bare functional minimum.

2: Use shorter relative paths in URLs instead of longer absolute ones.

3: Use the shortest naming conventions for your files that you can to maintain adequate organization: "home.html" rather than "homepage.html".

4: Consider using cascading style sheets instead of many occurrences of modifier tags.

Remember, that every byte saved counts to speed downloading and rendering, and it isn't any more trouble to make HTML with file size in mind than it is to not.

GETTING HELP WITH MIZER™

Mizer™ is simple and safe. There is little chance that you will need to ask for help with it. However, given the fast pace at which HTML is changing, it is possible that unanticipated changes might create new issues to address and require changes in Mizer's engine.

If you notice any change in appearance to your Web pages after it has been optimized with Mizer please let us know immediately.

Send email to support@antimonysoftware.com with a description of the visual change and include the original and optimized files as an email attachment.

That way you can help us stay on top of change.

Should you need help with **Mizer™**, prompt support is available by email.

support@antimonysoftware.com

Contact Information for Antimony Software

Antimony Software
2041 Sturgis-Maben Road
Maben, MS 39750
1-662-263-5412 (fax)

http://www.antimonysoftware.com
support@antimonysoftware.com