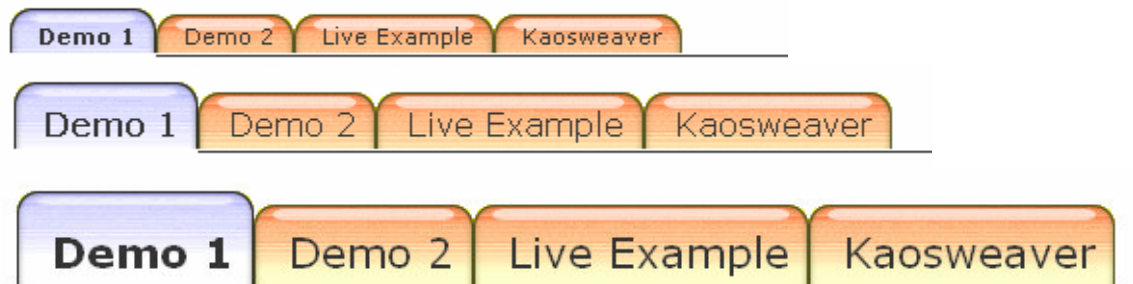


# Complete CSS Menu



## A product by Kaosweaver and TJK Design

Add one set of CSS code, two images, one unordered HTML list and a Dreamweaver extension and end up with a menu system that scales according to the font size you set or to the browser font size selection. Completely accessible for those with visual disabilities who need larger text. Search engine friendly! 100% CSS solution without any CSS hacks and **no JavaScript!** The three images above were all captured off of the same page; the only difference was the font size setting.

Complete CSS Menu is a single command extension.

For Dreamweaver MX and higher, Windows 2000 & XP or higher and OS 9.x and higher

# Table of Contents

## Start here

Overview	3
Setup and Installation	5
License	9

## Complete CSS Menu Extension Guide

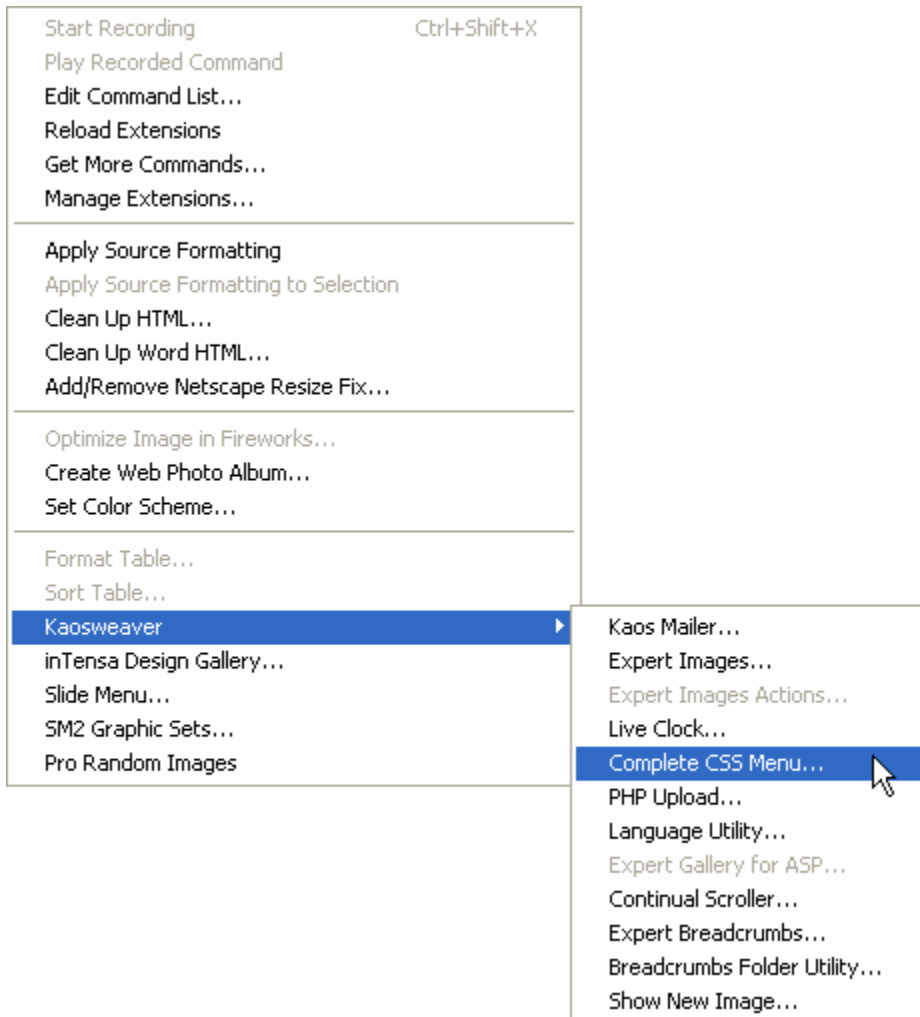
Interface	10
Setting up a menu	11
Editing an existing menu	14
Setting up a design time CSS	15

## Wrap Up

Troubleshooting	19
-----------------	----

## Overview

Complete CSS Menu is a Dreamweaver command, found under the Command menu and Kaosweaver sub panel.



The command produces a menu system that will allow your visitors to navigate your site with ease. The menu is accessible, valid CSS, fast to download, scalable and fast to implement.

## **Complete CSS Menu Features:**

- Scalable Menu to browser or selectable font size
- Rollover menu tabs
- Current page selection to let the visitors know where they are
- Extension aware design time CSS for close to WYSIWYG design
- External CSS for rapid download and global menu changes
- Editable menu items, no need to rework the menu for each change
- Preview of menu tab within the extension, including menu text
- Font family and color selection within extension
- Comes with 9 pre-made menu tabs, including rollover tabs
- Search engine friendly
- Extensive tutorial on the CSS scripting for the menu

# Setup and Installation

## Loading the extension into Dreamweaver

All extensions for Dreamweaver are loaded the same way, regardless of platform (PC or Mac) that the extension is on. One package (KW\_cCSSMenu.mxp) is used for both Windows and Macintosh distribution. Some requirements exist for our extensions as detailed below:

- Dreamweaver MX
  - Extension Manager v1.6
  - Dreamweaver MX v6.01 update or later
- Dreamweaver MX 2004
  - Extension Manager v1.6
  - Dreamweaver MX v7.01 or later

All of these are available as updates for free from the Macromedia website in their download section. This location has changed in the past or we'd provide a URL for you to reference. Our policy is to support the current release of Dreamweaver and the prior version. All versions behind the current version by two releases may be supported, however, we are unable to troubleshoot issues un-reproducible on the supported versions of Dreamweaver.

To load the extension, save it from the email you received with it attached. If you've requested the delivery email and it has not come yet, please check your email filtering software and check your junk bin. We do know most of the free email services will have email delivery delays of one to four hours. If time is absolutely critical, please send our support staff an email and we may be able to assist in a quicker delivery. Another alternative is to change the email on your account at the Kaosweaver site and resend the product to yourself. The delivery

system is subject to the limitations of email delivery and may sometimes experience delays because of the latest virus, worm or spam on the Internet. Once the extension is saved from the email, find it on your drive. Once found, double clicking on the extension will load Extension Manager that will automate the process of installation. Some possible messages will include a request to overwrite existing files. You can answer Yes or No to any of these and the install should not be impacted. This is usually due to shared images or resource files that Kaosweaver uses to prevent installing unique files for each extension which, in reality, are identical. Certain extensions will have resource files that you can edit through the extension and these extensions will require a little caution when re-installing to prevent your past modifications from being removed. All resource files from Kaosweaver will be saved with .xml extensions. Always answer no to overwriting these files when asked by the Extension Manager unless you're willing to lose the modifications (sometimes needed if the modifications crash the system). This process of installation is also the method we recommend when the extension (and most likely your system) becomes unstable and possibly corrupt. Dreamweaver is a great program, however, occasionally, it fails to run properly and has been known to corrupt third party (Kaosweaver for instance) extensions.

## **Troubleshooting**

If the installation encounters some problems, these solutions tend to work. Try them prior to emailing Kaosweaver for support (because this is what we'll tell you to try anyway):

1. Load Dreamweaver while attempting to install the extension.
2. Check your Extension Manager version with Macromedia to ensure you've got the latest version.
3. Load Extension Manager, select Dreamweaver (your version) from the dropdown, select Install Extension from the File menu and navigate to the extension package and select it.

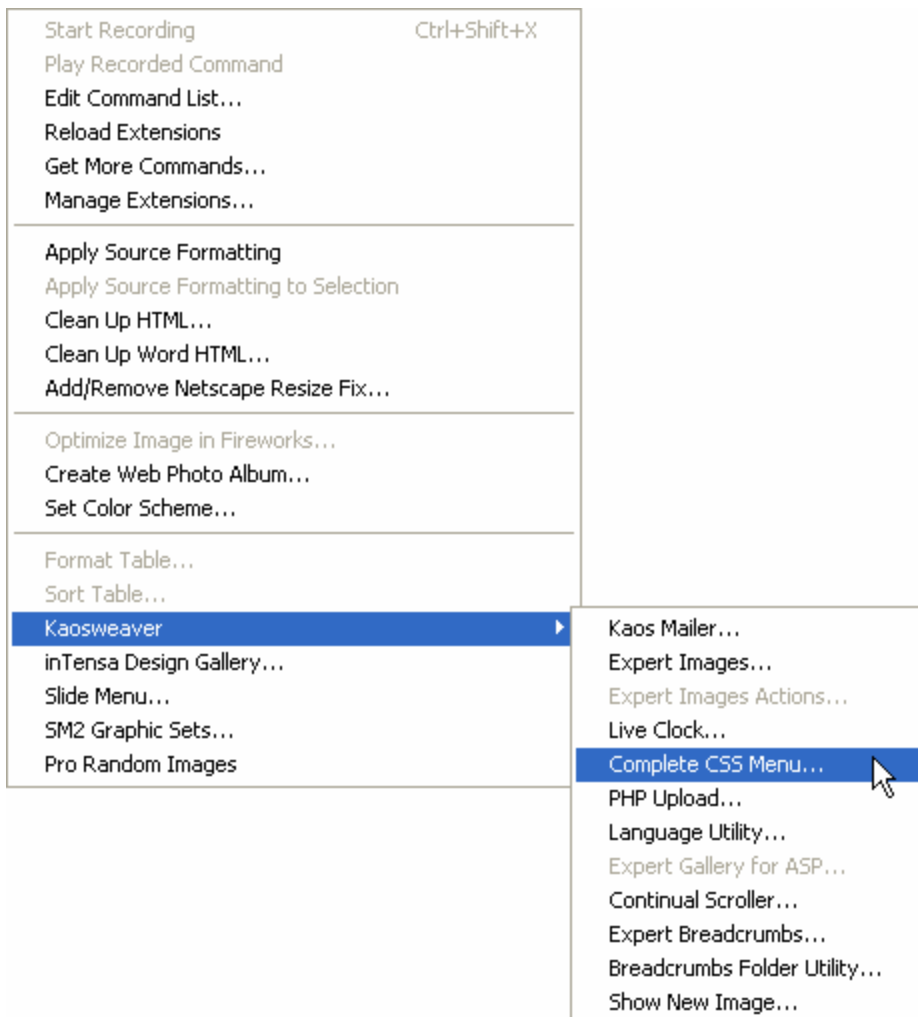
4. Check your file associations (this is system dependant, check your user manual or local techie on how to do this) and see if .mxp as a file extension is associated to Extension Manager.

## Restarting Dreamweaver

Once the extension is installed (or re-installed), restart Dreamweaver. This is critical for the extension to show up in the menus and be available.

## Finding Complete CSS Menu

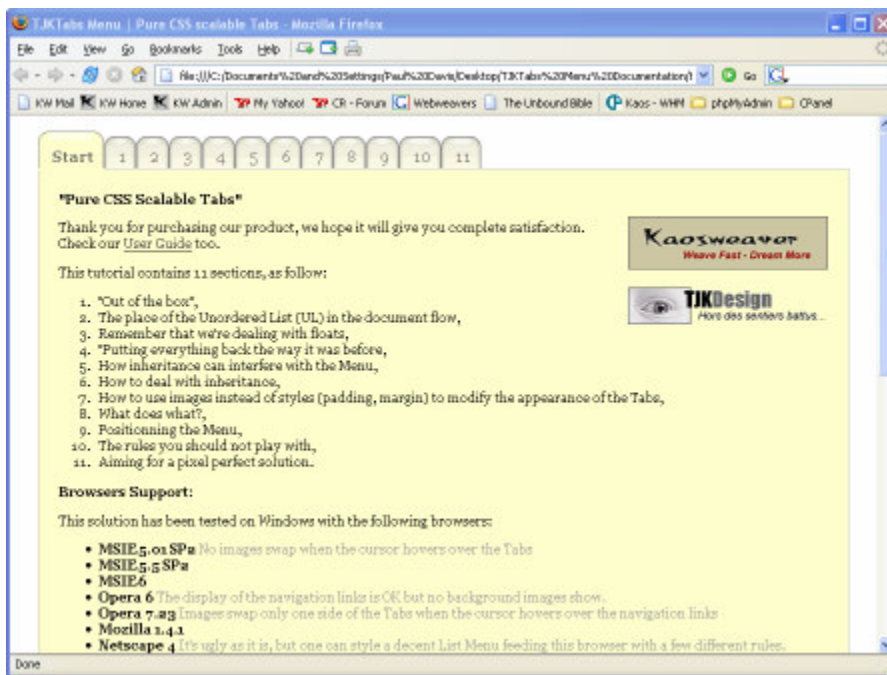
This command is found under the command menu as shown below:



## Working with the demo files

This extension comes with an excellent tutorial on the CSS properties and how to get the most out of the CSS, written by Thierry Koblentz from TJK Design. This will be the TJKTabs.zip file included in the zip file with this document pdf file. Also included is the tabs.zip file which contains a PNG file with all of the tabs from the extension in it. This can be used for modifying the tabs to suit your site's color scheme or help you to create new ones.

Unzip the files into a folder. The files all expand into one folder and the images, CSS files and HTML files are accessible from this folder. Click on the file named tutorial.htm to start looking at the tutorial



Which also contains a User Guide for assistance on page construction to get the most out of the Complete CSS Menu.

**For the best understanding of the resulting code, please consult these documents.**

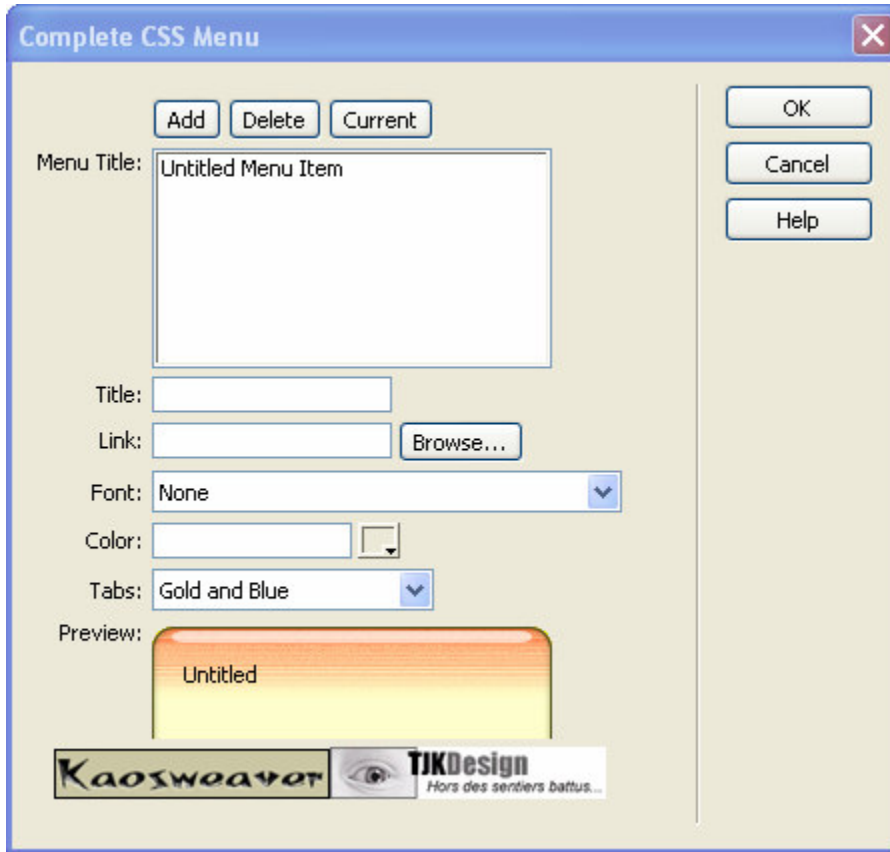


# License

1. This is an agreement by and between Kaosweaver, Inc. ("Licensor") and the End User ("Licensee"), who is being licensed to use the Complete CSS Menu extension.
2. Licensee acknowledges that this is only a limited nonexclusive license. Licensor is and remains the owner of all titles, rights, and interests in the Software.
3. This License permits Licensee to install the Software on more than one computer system, as long as the Software will not be used on more than one computer system simultaneously. Licensee will not make copies of the Software or allow copies of the Software to be made by others, unless authorized by this License Agreement. Licensee may make copies of the Software for backup purposes only.
4. This Software is subject to a limited warranty. Licensor warrants to Licensee that the physical medium on which this Software is distributed is free from defects in materials and workmanship under normal use, the Software will perform according to its printed documentation, and to the best of Licensor's knowledge Licensee's use of this Software according to the printed documentation is not an infringement of any third party's intellectual property rights. To the extent permitted by law, THE ABOVE-STATED LIMITED WARRANTY REPLACES ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND LICENSOR DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING ANY IMPLIED WARRANTY OF TITLE, MERCHANTABILITY, NONINFRINGEMENT, OR OF FITNESS FOR A PARTICULAR PURPOSE. No agent of Licensor is authorized to make any other warranties or to modify this limited warranty. Licensee has specific legal rights pursuant to this warranty and, depending on Licensee's jurisdiction, may have additional rights.
5. Notwithstanding the foregoing, LICENSOR IS NOT LIABLE TO LICENSEE FOR ANY DAMAGES, INCLUDING COMPENSATORY, SPECIAL, INCIDENTAL, EXEMPLARY, PUNITIVE, OR CONSEQUENTIAL DAMAGES, CONNECTED WITH OR RESULTING FROM THIS LICENSE AGREEMENT OR LICENSEE'S USE OF THIS SOFTWARE, whether based on lost revenue or otherwise, regardless of whether LICENSEE was advised of the possibility of such losses in advance. In no event shall LICENSOR's liability hereunder exceed the amount of license fees paid by Licensee, regardless of whether Licensee's claim is based on contract, tort, strict liability, product liability or otherwise. Licensee's jurisdiction may not allow such a limitation of damages, so this limitation may not apply.
6. Licensee agrees to defend and indemnify Licensor and hold Licensor harmless from all claims, losses, damages, complaints, or expenses connected with or resulting from Licensee's business operations.
7. Licensor has the right to terminate this License Agreement and Licensee's right to use this Software upon any material breach by Licensee.
8. Licensee agrees to return to Licensor or to destroy all copies of the Software upon termination of the License.
9. This License Agreement is the entire and exclusive agreement between Licensor and Licensee regarding this Software. This License Agreement replaces and supersedes all prior negotiations, dealings, and agreements between Licensor and Licensee regarding this Software.
10. This License Agreement is valid without Licensor's signature. It becomes effective upon the earlier of Licensee's signature or Licensee's use of the Software.
11. Kaosweaver extensions are trademarks of Kaosweaver Inc. Macromedia, Dreamweaver, and/or Fireworks, are registered trademarks, trademarks, or copyrights of Macromedia, Inc. in the United States of America and/or other countries.

# Complete CSS Menu Extension Guide

## Interface



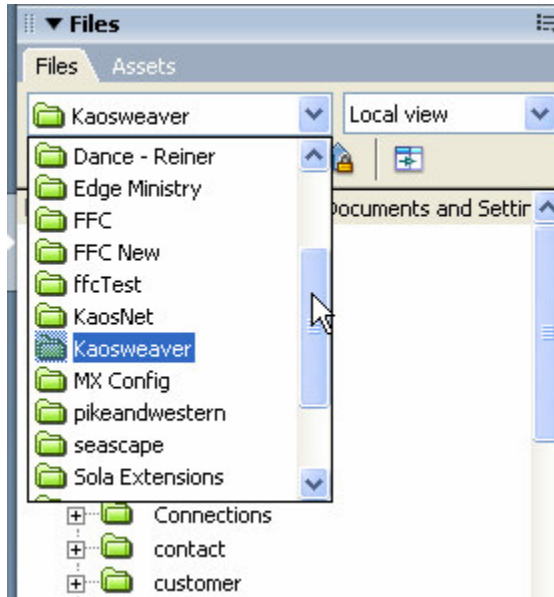
The Complete CSS Menu has several entries for the proper display of your menu. The first part is the buttons, the Menu Title list box, Title and Link. These all combine to control the number of tabs in the menu, which one is current (or active) for the page and where the tabs will link to. The Font, Color and Tabs control how the tabs will be displayed. The Link, Font and Color are optional all other fields are required. The Command buttons are OK, Cancel and Help.

## Setting up a menu

1. Load Dreamweaver:

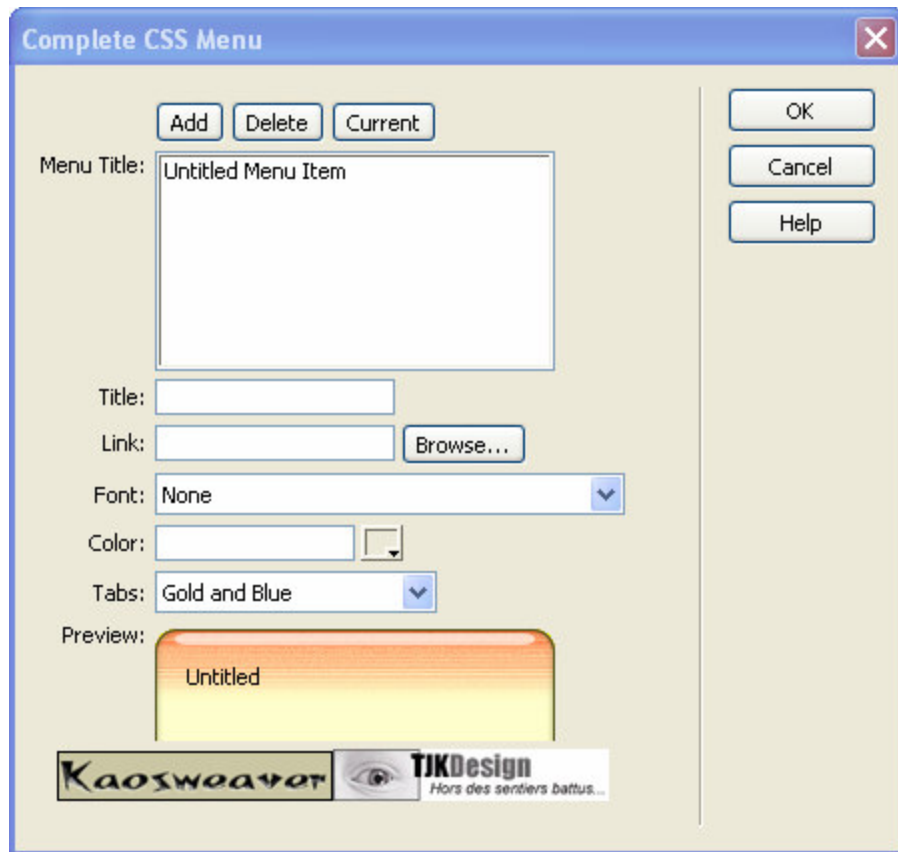


2. Select a site:



3. Open a file or create a new one.
4. If creating a new file **SAVE THE FILE PRIOR TO RUNNING COMPLETE CSS MENU** this will ensure the file path to your external CSS files will be recorded properly. Dreamweaver doesn't update file paths inside IE Conditional Comment.
5. Select the command menu.
6. Select Kaosweaver sub menu (or Kaosweaver.com sub menu, depending on your last installed extension selection)

7. Select Complete CSS Menu and the extension will load:



8. Click on the Add button for as many menu tabs as you need.
9. Click on an entry in the Menu Title list box, the existing title will appear in the Title text field and the link will appear in the Link text field. If no link has been selected, nothing will appear when creating the menu.
10. Change the title to match the text you would like on the tab.
11. Type in the link or use the Browse... button to find the file in your local site.
12. If the page being worked on is one of the menu items, select that Menu Title from the Menu Title list box and click on the Current button which will

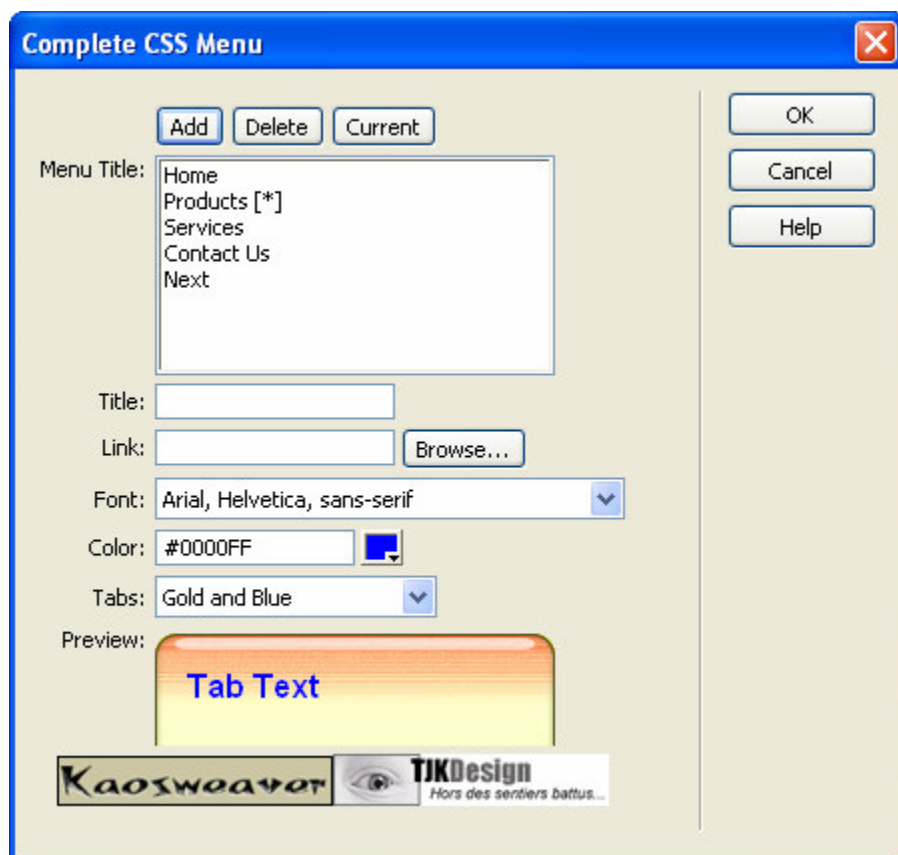
append the name with “ [\*]”, this is the designator for the current page and this will not appear on the title for the page.

13. If you have an entry in the Menu Title list box that you need to delete, clicking on the Delete button will change the title by adding a “<” in front of the title and a “>” after the title. If the deleted entry was selected as the current tab, this designation will be removed.
14. Repeat the process until all of the menu items have been titled and linked as needed.
15. If you would like the extension to manage the font, select the font for the menu from the dropdown. This font will apply to all of the tabs in the menu.
16. If you would like the extension to manage the color, select the color for the menu from the color picker or enter in the hex code color (with the # preceding the 6 digit hex code). The extension does not validate short hand CSS color entries (#fff for #ffffff) so long hex code entries will be required.
17. Finally, select the tab style you would like from the list. A preview of the selected tab style will be shown below, complete with the text from the last selected menu item in the Menu Title select box. If you would like to use a custom menu graphic, select one of the graphics from the list and then consult the tutorial and user guide in the HTML to replace the image. Do not name the image the same as the images used in the extension, they could be overwritten by the extension.

18. Click OK and the menu will appear on the page. Check the instructions on setting up a design time style guide to make the inserted menu look better in the Dreamweaver interface.

## Editing an Existing Menu

1. Load Dreamweaver.
2. Load the page that has Complete CSS Menu on it.
3. Load the Complete CSS Menu command (as described in steps 5, 6 and 7 in the prior section)
4. The Complete CSS Menu will appear with all of the options last entered shown within the extension.



5. The preview will have default text in it until a Menu Title is selected
6. If you are using a custom image, the preview section will display a gray image instead of the image you're using for the menu.
7. At this point, the menu is ready to be altered as if you had just finished entering all of the information into the extension for the first time. Any updates to the tabs, font or color will affect all menus attached to the CSS for the menu.

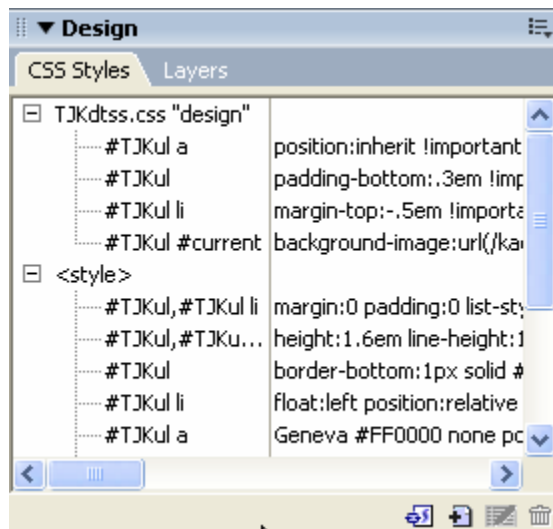
### Setting up a Design Time CSS

These instructions are for Dreamweaver MX 2004 and have not been tested in Dreamweaver MX

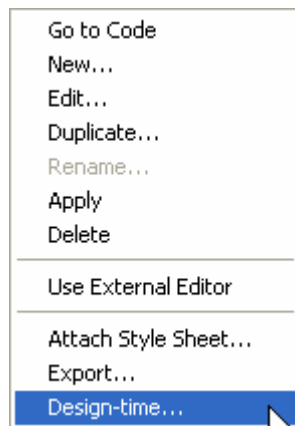
1. Load Dreamweaver
2. Load the page with the Complete CSS Menu on it
3. Find the documentation and then locate the file named TJKdtss.css and copy this to the kaosjs folder in the site where the page from step 2 is.
4. Load the CSS Panel from Dreamweaver:

✓ Insert	Ctrl+F2
✓ Properties	Ctrl+F3
CSS Styles	Shift+F11
Layers	F2
Behaviors	Shift+F4
Snippets	Shift+F9
Reference	Shift+F1
Databases	Ctrl+Shift+F10
Bindings	Ctrl+Shift+F11

5. Which opens the CSS Panel:

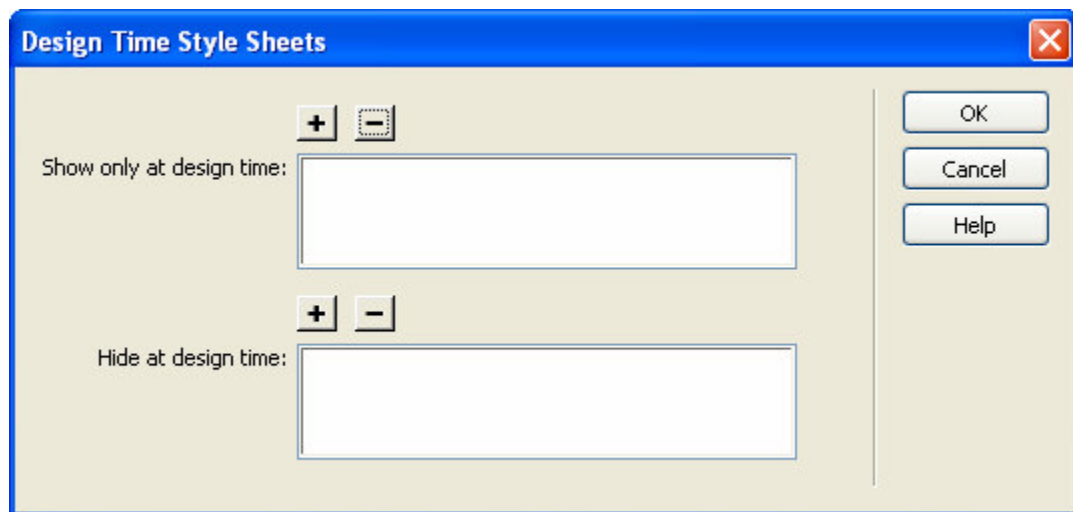


6. Right click on the white space in the CSS Panel which will bring up this context sensitive menu:

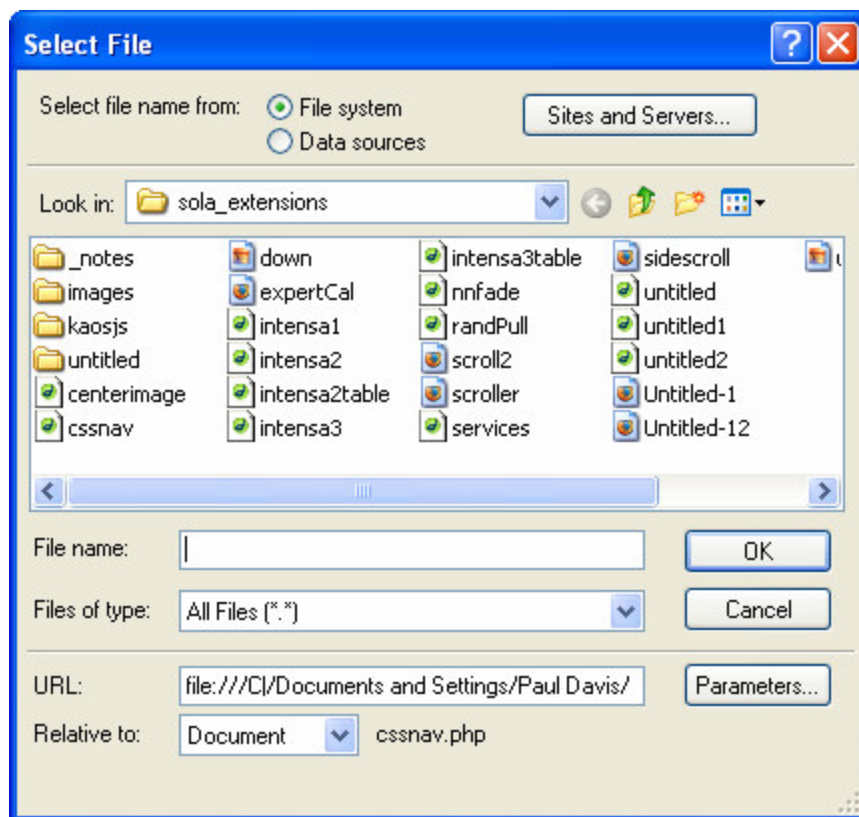




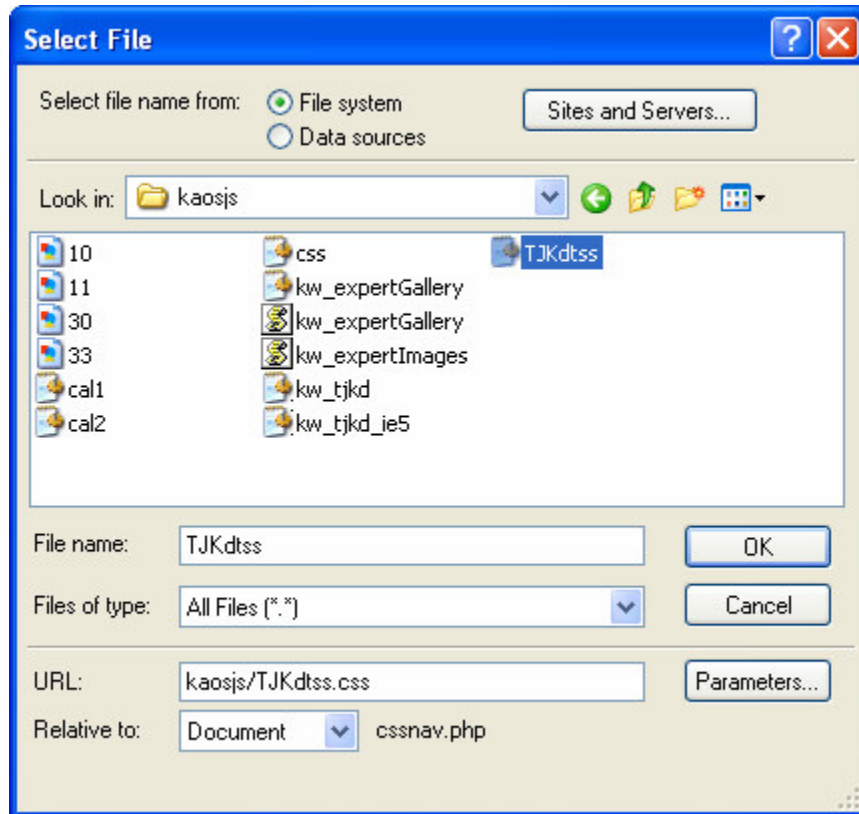
7. Select the Design-time... option which will load the selector:



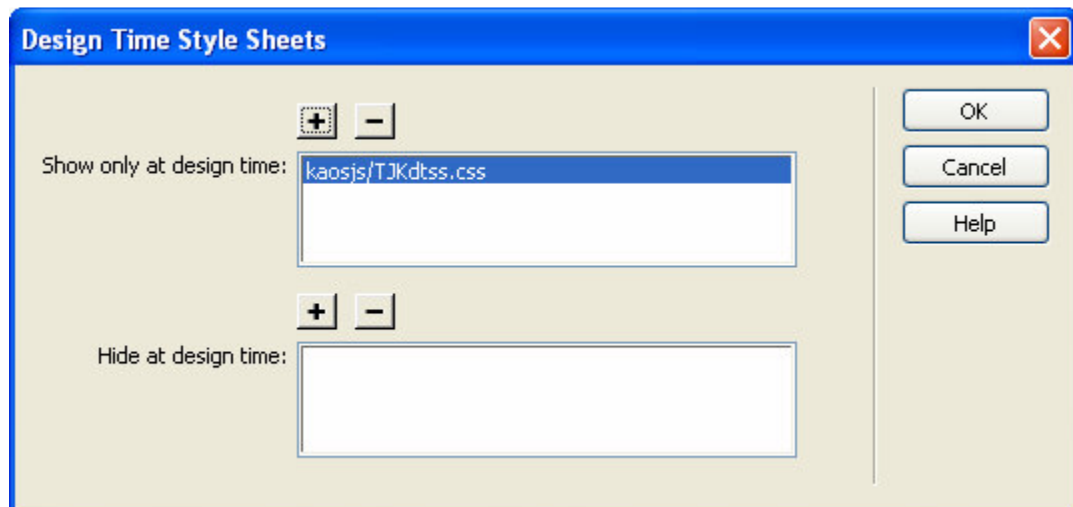
8. On the first box, Show only at design time:, click on the + above this box which will open up the file selector:



9. Navigate to the kaosjs folder and select the file from step 3 we copied to this folder:



10. Click OK and the file will be moved to the Design Time dialogue box:



11. Click OK and the menu will be changed to look better. Whenever you update the menu in Dreamweaver through the extension, the design time sheet will also be updated. However, Dreamweaver doesn't always refresh this in the design view for the page. If this is the case, closing the files and reopening them will reset the look of the page. Always save the files prior to closing them.

The design time CSS file does not need to be uploaded to the server as it is only used through the design Dreamweaver view of the page and not on the browser display of the page. The design time isn't an exact or perfect representation of the actual display in the browser, it is just designed to give you a visual look and feel for the site so you can have some good visual feedback for your page design.

## Wrap Up

Thank you for purchasing Complete CSS Menu. We hope you've found this documentation helpful and the extension profitable for your business or enabling you to make your web site work how you want it to. We hope we've covered all of the possible options, questions or usage issues with the manual. We have support available, either from email, forum or our ticket system, in case we've not explained something sufficiently enough so that you are able to use the product.

**We want to stress, please send us an email before investing hours working on this extension if you encounter a problem we've not expected. It has been our experience that we're able to resolve almost all issues within minutes of receiving the email (which could take some time to receive, depending on when it gets sent).** We are active and aggressive in releasing fixes, updates or modifications when we have something better. In order to provide the best support we can for you is have you provide us meaningful messages.

These bullet points are from an article on Effective Bug Reporting by Simon Tatham published under his [OpenContent Licence](#). The copy has been modified to suit our applications.

## How to Report Bugs Effectively

by [Simon Tatham](#), professional and free-software programmer

### Introduction

Anybody who has written software for public use will probably have received at least one bad bug report. Reports that say nothing ("It doesn't work!"); reports that make no sense; reports that don't give enough information; reports that give wrong information. Reports of problems that turn out to be user error; reports of problems that turn out to be the fault of somebody else's program; reports of problems that turn out to be network failures.

There's a reason why technical support is seen as a horrible job to be in, and that reason is bad bug reports. However, not all bug reports are unpleasant: I maintain free software, when I'm not earning my living, and sometimes I receive wonderfully clear, helpful, informative bug reports.

In a nutshell, the aim of a bug report is to enable the programmer to see the program failing in front of them. You can give them careful and detailed instructions on how to make it fail. If they can make it fail, they will try to gather extra information until they know the cause. If they can't make it fail, they will have to ask you to gather that information for them.

In bug reports, try to make very clear what are actual facts ("I was at the computer and this happened") and what are speculations ("I think the problem might be this"). Leave out speculations if you want to, but don't leave out facts.

When you report a bug, you are doing so because you want the bug fixed. There is no point in swearing at the programmer or being deliberately unhelpful: it may be their fault and your problem, and you might be right to be angry with them, but the bug will get fixed faster if you help them by supplying all the information they need.

### "It doesn't work."

Give the programmer some credit for basic intelligence: if the program really didn't work at all, they would probably have noticed. Since they

haven't noticed, it must be working for them. Therefore, either you are doing something differently from them, or your environment is different from theirs. They need information; providing this information is the purpose of a bug report. More information is almost always better than less.

If you are not reporting a bug but just asking for help using the program, you should state where you have already looked for the answer to your question. ("I looked in chapter 4 and section 5.2 but couldn't find anything that told me if this is possible.") This will let the programmer know where people will expect to find the answer, so they can make the documentation easier to use.

### **"Show me how to show myself."**

If you have to report a bug to a programmer who can't be present in person, the aim of the exercise is to enable them to *reproduce* the problem. You want the programmer to run their own copy of the program, do the same things to it, and make it fail in the same way. When they can see the problem happening in front of their eyes, then they can deal with it.

So tell them exactly what you did. If it's a graphical program, tell them which buttons you pressed and what order you pressed them in. If it's a program you run by typing a command, show them precisely what command you typed. Wherever possible, you should provide a verbatim transcript of the session, showing what commands you typed and what the computer output in response.

Give the programmer all the input you can think of. If the program reads from a file, you will probably need to send a copy of the file. If the program talks to another computer over a network, you probably can't send a copy of that computer, but you can at least say what kind of computer it is, and (if you can) what software is running on it.

[Kaosweaver: send all files to us by zipping them into one file, with subfolders as necessary – unzipped files are deleted by virus and spam filters]

[Kaosweaver – If possible, upload the page if the error isn't in the extension and provide the URL to the page, this is very very helpful]

### **"Works for me. So what goes wrong?"**

If you give the programmer a long list of inputs and actions, and they fire up their own copy of the program and nothing goes wrong, then you haven't given them enough information. Possibly the fault doesn't show up on every computer; your system and theirs may differ in some way. Possibly you

have misunderstood what the program is supposed to do, and you are both looking at exactly the same display but you think it's wrong and they know it's right.

So also describe what happened. Tell them exactly what you saw. Tell them why you think what you saw is wrong; better still, tell them exactly what you expected to see. If you say "and then it went wrong", you have left out some very important information.

If you saw error messages then tell the programmer, carefully and precisely, what they were. They *are* important! At this stage, the programmer is not trying to fix the problem: they're just trying to find it. They need to know what has gone wrong, and those error messages are the computer's best effort to tell you that. Write the errors down if you have no other easy way to remember them, but it's not worth reporting that the program generated an error unless you can also report what the error message was.

In particular, if the error message has numbers in it, *do* let the programmer have those numbers. Just because you can't see any meaning in them doesn't mean there isn't any. Numbers contain all kinds of information that can be read by programmers, and they are likely to contain vital clues. Numbers in error messages are there because the computer is too confused to report the error in words, but is doing the best it can to get the important information to you somehow.

At this stage, the programmer is effectively doing detective work. They don't know what's happened, and they can't get close enough to watch it happening for themselves, so they are searching for clues that might give it away. Error messages, incomprehensible strings of numbers, and even unexplained delays are all just as important as fingerprints at the scene of a crime. Keep them!

[ Kaosweaver: Screen shots of error messages are **perfect!** If just sending one or two images of errors, you don't have to zip them, please use jpeg or gif formats for size considerations]

## **"So then I tried . . ."**

There are a lot of things you might do when an error or bug comes up. Many of them make the problem worse. A friend of mine at school deleted all her Word documents by mistake, and before calling in any expert help, she tried reinstalling Word, and then she tried running Defrag. Neither of these helped recover her files, and between them they scrambled her disk to the extent that no Undelete program in the world would have been able to recover anything. If she'd only left it alone, she might have had a chance.

Users like this are like a mongoose backed into a corner: with its back to the wall and seeing certain death staring it in the face, it attacks frantically, because doing something has to be better than doing nothing. This is not well adapted to the type of problems computers produce.

Instead of being a mongoose, be an antelope. When an antelope is confronted with something unexpected or frightening, it freezes. It stays absolutely still and tries not to attract any attention, while it stops and thinks and works out the best thing to do. (If antelopes had a technical support line, it would be telephoning it at this point.) Then, once it has decided what the safest thing to do is, it does it.

When something goes wrong, immediately stop doing *anything*. Don't touch any buttons at all. Look at the screen and notice everything out of the ordinary, and remember it or write it down. Then perhaps start cautiously pressing "OK" or "Cancel", whichever seems safest. Try to develop a reflex reaction - if a computer does anything unexpected, freeze.

If you manage to get out of the problem, whether by closing down the affected program or by rebooting the computer, a good thing to do is to try to make it happen again. Programmers like problems that they can reproduce more than once. Happy programmers fix bugs faster and more efficiently.

### **"I think the tachyon modulation must be wrongly polarised."**

It isn't only non-programmers who produce bad bug reports. Some of the worst bug reports I've ever seen come from programmers, and even from good programmers.

I worked with another programmer once, who kept finding bugs in his own code and trying to fix them. Every so often he'd hit a bug he couldn't solve, and he'd call me over to help. "What's gone wrong?" I'd ask. He would reply by telling me his current opinion of what needed to be fixed.

This worked fine when his current opinion was right. It meant he'd already done half the work and we were able to finish the job together. It was efficient and useful.

But quite often he was wrong. We would work for some time trying to figure out why some particular part of the program was producing incorrect data, and eventually we would discover that it wasn't, that we'd been investigating a perfectly good piece of code for half an hour, and that the actual problem was somewhere else.

I'm sure he wouldn't do that to a doctor. "Doctor, I need a prescription for Hydroyoyodyne." People know not to say that to a doctor: you describe the symptoms, the actual discomforts and aches and pains and rashes and fevers, and you let the doctor do the diagnosis of what the problem is and what to do about it. Otherwise the doctor dismisses you as a hypochondriac or crackpot, and quite rightly so.

It's the same with programmers. Providing your own diagnosis might be helpful sometimes, but always state the symptoms. The diagnosis is an optional extra, and not an alternative to giving the symptoms. Equally, sending a modification to the code to fix the problem is a useful addition to a bug report but not an adequate substitute for one.

If a programmer asks you for extra information, don't make it up! Somebody reported a bug to me once, and I asked him to try a command that I knew wouldn't work. The reason I asked him to try it was that I wanted to know which of two different error messages it would give. Knowing which error message came back would give a vital clue. But he didn't actually try it - he just mailed me back and said "No, that won't work". It took me some time to persuade him to try it for real.

Using your intelligence to help the programmer is fine. Even if your deductions are wrong, the programmer should be grateful that you at least *tried* to make their life easier. But report the symptoms as well, or you may well make their life much more difficult instead.

### **"That's funny, it did it a moment ago."**

Say "intermittent fault" to any programmer and watch their face fall. The easy problems are the ones where performing a simple sequence of actions will cause the failure to occur. The programmer can then repeat those actions under closely observed test conditions and watch what happens in great detail. Too many problems simply don't work that way: there will be programs that fail once a week, or fail once in a blue moon, or never fail when you try them in front of the programmer but always fail when you have a deadline coming up.

Most intermittent faults are not truly intermittent. Most of them have some logic somewhere. Some might occur when the machine is running out of memory, some might occur when another program tries to modify a critical file at the wrong moment, and some might occur only in the first half of every hour! (I've actually seen one of these.)

Also, if you can reproduce the bug but the programmer can't, it could very well be that their computer and your computer are different in some way and this difference is causing the problem. I had a program once whose



window curled up into a little ball in the top left corner of the screen, and sat there and *sulked*. But it only did it on 800x600 screens; it was fine on my 1024x768 monitor.

The programmer will want to know anything you can find out about the problem. Try it on another machine, perhaps. Try it twice or three times and see how often it fails. If it goes wrong when you're doing serious work but not when you're trying to demonstrate it, it might be long running times or large files that make it fall over. Try to remember as much detail as you can about what you were doing to it when it did fall over, and if you see any patterns, mention them. Anything you can provide has to be some help. Even if it's only probabilistic (such as "it tends to crash more often when Emacs is running"), it might not provide direct clues to the cause of the problem, but it might help the programmer reproduce it.

Most importantly, the programmer will want to be sure of whether they're dealing with a true intermittent fault or a machine-specific fault. They will want to know lots of details about your computer, so they can work out how it differs from theirs. A lot of these details will depend on the particular program, but one thing you should definitely be ready to provide is version numbers. The version number of the program itself, and the version number of the operating system, and probably the version numbers of any other programs that are involved in the problem.

## **"So I loaded the disk on to my Windows . . ."**

Writing clearly is essential in a bug report. If the programmer can't tell what you meant, you might as well not have said anything.

I get bug reports from all around the world. Many of them are from non-native English speakers, and a lot of those apologise for their poor English. In general, the bug reports with apologies for their poor English are actually very clear and useful. All the most unclear reports come from native English speakers who assume that I will understand them even if they don't make any effort to be clear or precise.

- *Be specific.* If you can do the same thing two different ways, state which one you used. "I selected Load" might mean "I clicked on Load" or "I pressed Alt-L". Say which you did. Sometimes it matters.
- *Be verbose.* Give more information rather than less. If you say too much, the programmer can ignore some of it. If you say too little, they have to come back and ask more questions. One bug report I received was a single sentence; every time I asked for more information, the reporter would reply with another single sentence. It took me several weeks to get a useful amount of information, because it turned up one short sentence at a time.

- *Be careful of pronouns.* Don't use words like "it", or references like "the window", when it's unclear what they mean. Consider this: "I started FooApp. It put up a warning window. I tried to close it and it crashed." It isn't clear what the user tried to close. Did they try to close the warning window, or the whole of FooApp? It makes a difference. Instead, you could say "I started FooApp, which put up a warning window. I tried to close the warning window, and FooApp crashed." This is longer and more repetitive, but also clearer and less easy to misunderstand.
- *Read what you wrote.* Read the report back to yourself, and see if *you* think it's clear. If you have listed a sequence of actions which should produce the failure, try following them yourself, to see if you missed a step.

## Summary

- The first aim of a bug report is to let the programmer see the failure with their own eyes. If you can't be with them to make it fail in front of them, give them detailed instructions so that they can make it fail for themselves.
- In case the first aim doesn't succeed, and the programmer *can't* see it failing themselves, the second aim of a bug report is to describe what went wrong. Describe everything in detail. State what you saw, and also state what you expected to see. Write down the error messages, *especially* if they have numbers in.
- When your computer does something unexpected, *freeze*. Do nothing until you're calm, and don't do anything that you think might be dangerous.
- By all means try to diagnose the fault yourself if you think you can, but if you do, you should still report the symptoms as well.
- Be ready to provide extra information if the programmer needs it. If they didn't need it, they wouldn't be asking for it. They aren't being deliberately awkward. Have version numbers at your fingertips, because they will probably be needed.
- Write clearly. Say what you mean, and make sure it can't be misinterpreted.
- Above all, *be precise*. Programmers like precision.

---

*Disclaimer:* I've never actually seen a mongoose or an antelope. My zoology may be inaccurate.

*\$Id: bugs.html,v 1.24 2004/06/01 14:06:37 simon Exp \$*

Copyright © 1999 Simon Tatham.

This document is [OpenContent](#).

You may copy and use the text under the terms of the [OpenContent Licence](#).

Please send comments and criticism on this article to [anakin@pobox.com](mailto:anakin@pobox.com).

We would like to add some things that we've noticed help us help you a lot better:

1. Always restart Dreamweaver after installing an extension.
2. Always work from a saved page that is part of a site, unconnected pages don't work well with paths to files it can't find. Dreamweaver has no clue where you want to save the file and it will insert machine paths instead of relative paths which enable the web server to find the file.
3. If you get an error working on a complex page, try the same process on a blank page with just the minimum elements needed to accomplish the same task. For instance, in Expert Calendar, just have a form, the form elements used, the link/button/image and then a layer (if needed) and see if the error can be duplicated.
4. Put it on a test page and on a web server.
5. Try it on another machine.
6. Try it on a machine not using Windows 98 or Windows ME or Mac OS 8.x or Mac OS 9.x.
7. Always state what version of Dreamweaver you have (including if you've upgraded to the latest bug fix from Macromedia), the Extension Manager if you're having install problems, and the operating system with version. If these are stated, we will ask before even looking into the problem.

Following these suggestions will assist us in providing you with the quickest solution to the problem. We really want to solve whatever problem you've encountered with our software. We don't spending money on things we don't think are working like what they said it would do so we don't want to have our software fail you and not fix it. If we have the opportunity to fix the problem, we both win – you get what you paid for, we get the chance to fix a bug we don't know about.

## Credits

This program was made possible by the input of several important people. Some people really stood out and we'd like to thank them here:

- Nancy Gill – participated in the beta testing of this extension.
- Kim Kruse - participated in the beta testing of this extension. Provided an excellent example of the product in a live production site.
- Nadia Perre – participated in the beta testing of this extension.

Paul thanks Samantha, his wife, who sacrifices time with him. Without her sacrifice, Kaosweaver wouldn't exist. Paul also thanks God for the skill and ability to do what he does.

Thank you for purchasing Complete CSS Menu and let us know what else you'd like to see in an extension!

Paul Davis

<http://www.kaosweaver.com/>

Thierry Koblentz

<http://www.tjkdesign.com/>

All screen shots taken with TechSmith's Snagit v7, visit them on the web here:

<http://www.techsmith.com/>