

Continual Scroller

By Kaosweaver

Another professional image gallery product from Kaosweaver

Continual Scroller is a product which fills a niche for a quick, pleasant scrolling photo gallery. It is extremely light and can handle a lot of images. The images will continually scroll, wrapping around to the first when the last image has been released. This is all done in JavaScript so there is no need to have plug ins installed in order for the scroller to work, nor is there a delay in the downloading and installing of the customers who actually are willing to download and install a plug in for your site. Continual Scroller is browser friendly, working with all modern browsers on both the PC and the Mac platforms.

Continual Scroller is designed to work in Dreamweaver MX or MX 2004 on the PC and Mac platforms. It requires, for PCs, Windows 2000 or XP – either the Home or Professional editions. It may work in other Window OS versions, however these are not directly supported. On the Mac, OS 9.x or better is required. It may work in v8.x, however this isn't directly supported.

Table of Contents

Start here

Overview	3
Setup and Installation	4
License	8

Quick Start Guide

Scroller Quick Start Guide	9
----------------------------	---

***Continual Scroller* Command Guide**

Overview	10
Preparing the page for the scroller	12

Advanced Options

Overview	14
The JavaScript source code	14

Wrap Up

Troubleshooting	16
-----------------	----

Overview

Continual Scroller is a Dreamweaver command that can be found under the Kaosweaver.com submenu off of the Dreamweaver Command menu.

Application Use

Continual Scroller is suited for displaying graphic art – they can be photographs, composed images or other image media. The number of images is unlimited (with the exception of Dreamweaver or the browser failing), however keep in mind, all of the images that are placed in the scroller are also loaded on the page. This will impact the page weight once the images start to top 20 or 30 images (plus, no one is going to wait to see all of the images pass anyway).

Features

- Add images individually or by the folder
- Control the image positioning in the scroller
- Add links for each image
- Sort the image list
- Remove unwanted images from folder adds
- Control spacing between images
- Control the background color of the scroller
- Control the position of the scroller on the page (absolute positioned layer)
- Control the speed of the scrolling
- Editable

Setup and Installation

Installing the *Continual Scroller* extension in Dreamweaver

All Dreamweaver Extensions in MXP format, are installed the same way, regardless of platform (PC or Mac) used. The *Continual Scroller* Extension is one file, *kw_Extensions.mxp*, and is used for both Windows and Macintosh distribution.

Requirements for our extensions are detailed below. Please note, this extension may not work with all of the versions of Dreamweaver listed below.

- Dreamweaver 4 or UltraDev 4
 - Extension Manager v1.5
 - Dreamweaver/UltraDev update v4.01 or later
- 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

Extension Manager Files and Update files are available free of charge from the Macromedia web site, located at <http://www.macromedia.com>. The majority of our products do not work with Macromedia Dreamweaver v3 or UltraDev v1 or earlier. 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.

Installing the Continual Scroller Extension

Follow these simple steps to install the *Continual Scroller* extension;

1. Open the email message that you received from Kaosweaver@kaosweaver.com
2. Right click the MXP file.

3. Select **save as** and save the file. (Note the folder location)
4. Select the file and double click to open it.
5. The extension will load in the Extension Manager via an automated process.

Some possible messages will include a request to overwrite existing files. You can answer “**Yes**” or “**No**”, to any of these messages 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.

If you’ve requested the delivery email and it has not arrived, please check your email filtering software and check your junk bin. We do know that 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.

Troubleshooting

If the installation is not successfully completed, please follow these steps prior to emailing Kaosweaver for support

1. Load Dreamweaver prior to attempting to install the extension
2. Check your Extension Manager version to ensure you have the latest version from <http://www.macromedia.com>.
3. Open the Extension Manager, (Start -> Programs -> Macromedia -> Macromedia Extension Manager):
 - Select your latest version of Dreamweaver from the dropdown,
 - Select Install Extension from the File menu and
 - Navigate to the extension package and select it.
4. Check you file associations (this is system dependent, check your user manual or local techie on how to do this) and see if .MXP as a file extension is associated with the Extension Manager.

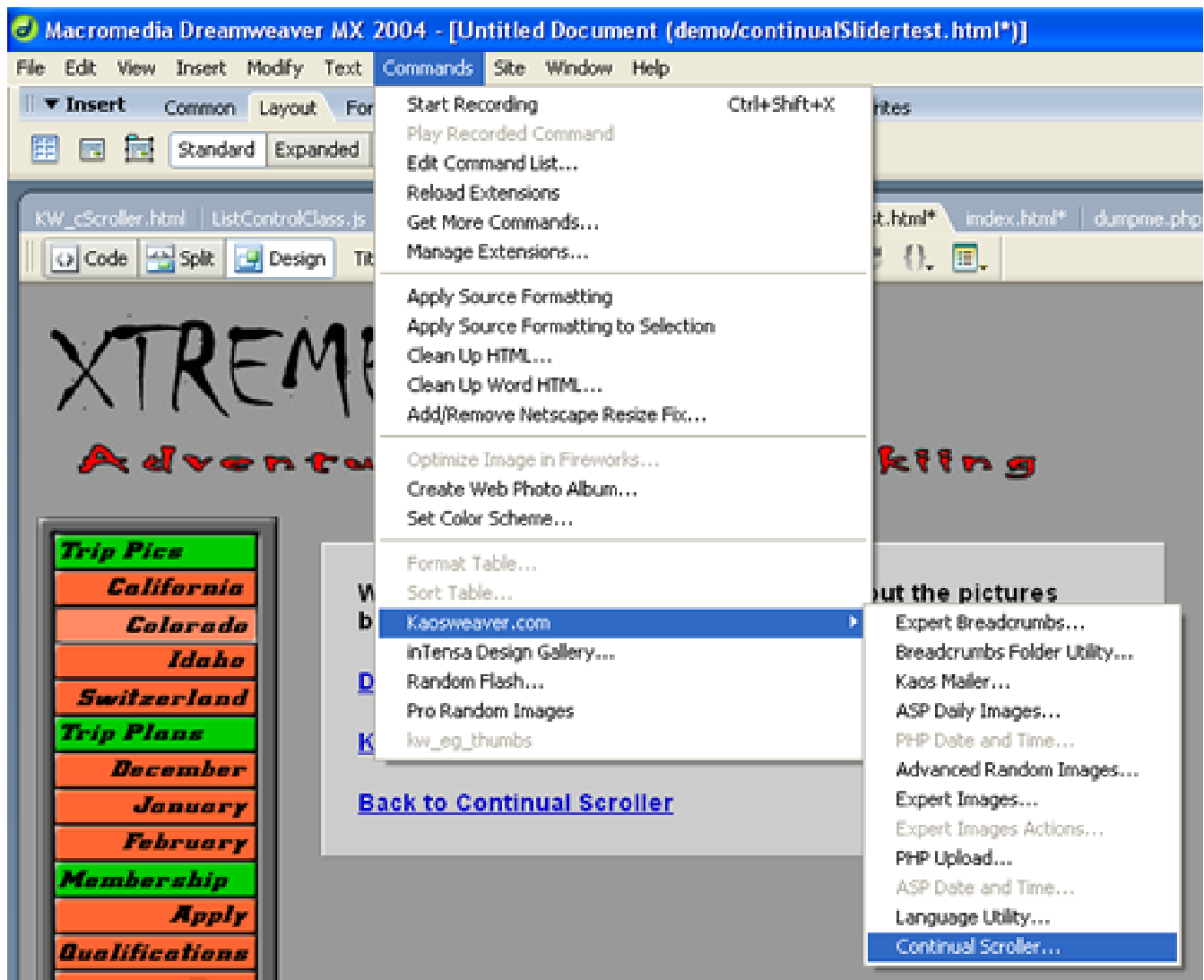
Restarting Dreamweaver

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 Continual Scroller

Continual Scroller is found under the command menu for Dreamweaver, then under the Kaosweaver.com submenu under the command menu.

If Continual Scroller is gray (inaccessible), open a page which has been saved into a site and make sure the site that the page is a part of is also open.



This will load the command in Dreamweaver. The command looks like:

Continual Scroller

Images:

Link: Browse... All

Alt: All

Scroller Position: Top Left

Scroller Background:

Scroller Width: ☐ Pixels ☒ Percent

Scroller Speed: ☐ Slow ☒ Medium ☐ Fast

Scroller Spacing: Pixels

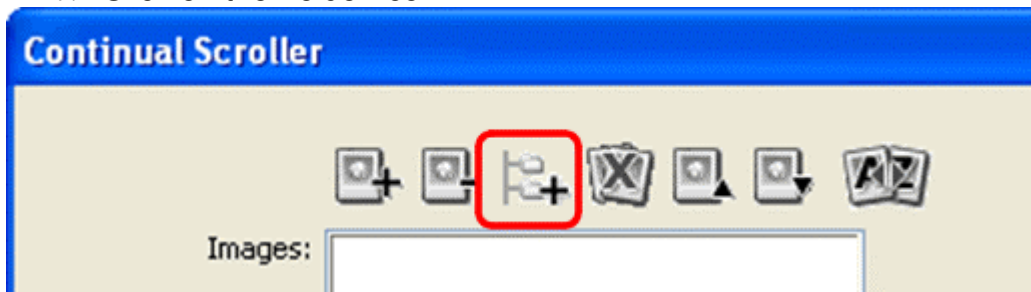
OK Cancel Remove

License

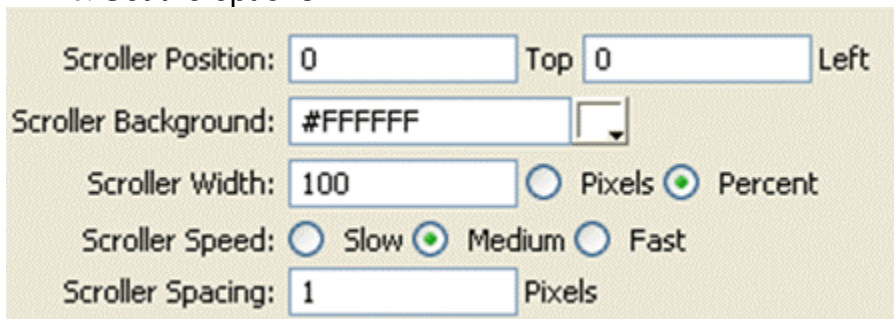
1. This is an agreement by and between Kaosweaver, Inc. ("Licensor") and the End User ("Licensee"), who is being licensed to use the Continual Scroller extension.
2. Licensee acknowledges that this is only a limited nonexclusive license. Licensor is and remains the owner of all titles, rights, and interests 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

Quick Start Guide

1. Create a new page in a site.
2. Save the new page.
3. Setup a folder in the open site with images you want in the scroller.
4. Select Commands from the Dreamweaver menu bar.
5. Select Kaosweaver.com
6. Select Continual Scroller...
7. Click on the Folder Icon:



8. Navigate to the folder with the images (from step 3).
9. Click OK on the image folder (which shouldn't list any of the individual images and just the folder name)
10. Set the options:



Top and left should be where you want the scroller on the page. The width is determined from the entry, however the height is automatic, it will be the maximum image height pulled from the images in the folder. The background color is only important if the spacing option is used. The color shows between the images (and under the shorter images). The speed controls how fast the scroller will move.

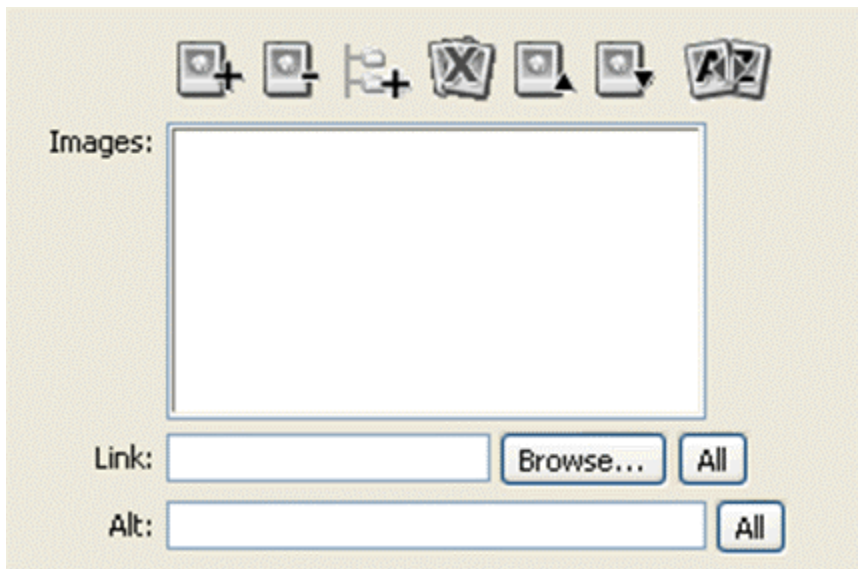
11. Click OK.

Continual Scroller Guide

The Continual Scroller is broken down into two main parts, the images section and the options section. The image section comprises of the image icon bar, the image list and the alt/URL entry forms. The options section control how the script will operate in the page. It controls the position, width, color, spacing and speed of the scroller. The only required fields are the images, the speed and the position. The alt, URL, color and spacing can be left as the defaults or changed as needed.

The other section is the extension control buttons. Ok, Cancel and Remove. The Ok confirms the extension as being inserted into the page with the entered options on the extension entries. The Cancel stops the extension from doing anything else. No changes will be made or saved to the page. Remove deletes the JavaScript from the page.

The Image Section



The image icon tool bar is the line of icons across the top of the image section, the images list is the list box with the Images: label and the Link and Alt fields are on the bottom and are used for entering in the Link and Alt entries for each image.

The image icon bar:



The icon bar consists of seven icons, from left to right, they are:

1. Add an image
2. Delete an image
3. Add a folder
4. Delete all images
5. Move selected image up one in the list
6. Move selected image down one in the list
7. Sort the images from A-Z

Add an image opens a new dialogue window for adding a specific image. The files in this dialogue box will be listed and individually selectable. Only one image at a time can be added this way. The image will be added to the end of the images list.

Delete an image deletes the selected image from the images list. It is done by surrounding the image name with < and > so that the deleted image appears like:

<deletedImage.gif>

If an image is already selected for deletion, clicking on the delete icon again will allow you to restore the image to the images list. The extension will ask the change in the deleted status of the image to be confirmed. Once this is confirmed the < and > will be removed. The properties of the image are retained until the extension is over, so deleting an image will not impact the stored information for that image.

Delete all images removes all of the images in the list in the same manner that a single deletion does. Each image can be individually restored by selecting the image and clicking on the delete an image icon. Again, the information stored for each image is not lost until the extension is completed.

Move selected image up one in the list icon will move the image up in the list until the image is the first in the list. The images are displayed from left to right on the web page and the top image in the list will be the first image on the left of the scrolling display (which means it is the first one to leave the screen as well). The last image will be in the hidden area of the scroller and won't show until the line of images has moved to display it.

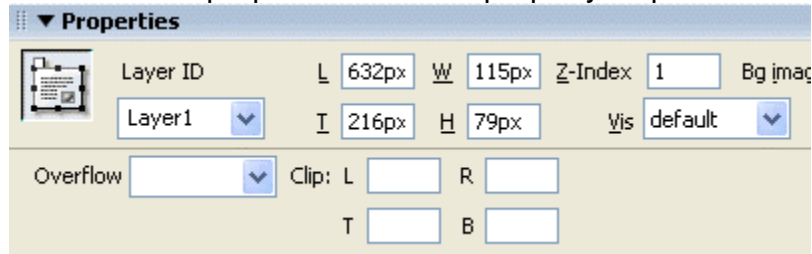
Move selected image down one in the list icon will move the image down in the list until the image is the last image in the list. This is identical to the prior function, just in the opposite direction.

Sort images will sort the image list in alphabetical order. Images added, deleted, moved up or down will not then be sorted alphabetically. If the change in images is needed to also be sorted alphabetically, this icon will need to be clicked on again.

Prepare a page for Continual scroller

Continual scroller does not require any pre-existing elements on the page in order to function, however, it would be beneficial to have the page layout completed prior to attempting to setup the scroller so that it will look right on the page and not have to be moved later. Some consideration that need to be addressed when setting up a page for Continual Scroller is the location of the focus (where the insertion point is at, or the cursor is at) when the extension is run. This is where Continual Scroller will insert the JavaScript. If this is in a layer, the scroller will be inserted into the page within that layer and the references to the top and left as well as the width restriction will be from the containing layer's top and left position. Setting a Continual Scroller inside another layer is possible; just make sure the top and left positions are in reference to the corner of the container layer and not the corner of the page. Using this little trick, one could find a way or an extension which would snap the layer to a fluid design so the scroller would move back and forth according to the fluid design instead of being locked into the initial positions.

The best way to determine what the top and left positions for the Continual Scroller will be is to place a temporary layer on the page where you want the Continual Scroller to be and then size it up to as wide as you need. The height is determined by the image sizes and doesn't need to be recorded. Take the top, left and width properties from the property inspector:



In the above image, the L (632px) is the left property, the W (115) is the width property and the T (216px) is the top property. Change these to match the needs of the design for your website.

Then set the speed of the scroller, the larger the scroller images, the slower the scrolling should be. Finally, determine if the images will have spaces between them and enter the number of pixels that will be between the images. Entering 0

(zero) will make it so the images are next to each other with no spaces. If you enter a space greater than zero, select the background color.

Click OK and the Continual Scroller will be on the page and ready to go.

Editing a page

If you have Continual Scroller on a web page and you need to change the image list, simply load the command on the page and the images, scroller position, speed, spacing and background color will be loaded and the extension will overwrite the existing scroller when finished. Otherwise it operates as detailed above.

Advanced Options

The Continual Scroller script, itself, is very light and doesn't consist of a lot of optional or changeable features. We will go over the JavaScript itself as well as the accompanying HTML & CSS.

JavaScript

```
var KW_iL = 0;
var KW_iLdr = 0;
function KW_csClass(w,h) {this.W=w;this.H=h;}
var KW_csIMG = new Array();
KW_csIMG[0] = new KW_csClass(348,248)
KW_csIMG[1] = new KW_csClass(348,248)
KW_csIMG[2] = new KW_csClass(348,248)
KW_csIMG[3] = new KW_csClass(348,248)
KW_initScroll(5,10);
KW_initDivs();
function KW_initScroll(s,sd) { // v1.0.0 Copyright Kaosweaver
    d=document;d.KW_seq=KW_csIMG.length-1;d.KW_timer = window.setInterval('KW_cScroller('+s+')',sd);
}
function KW_endScroll(){ // v1.0.0 Copyright Kaosweaver
    clearInterval(document.KW_timer);
}
function KW_initDivs() { // v1.0.0 Copyright Kaosweaver
    d=document;if(d.getElementById) for(var i=0;d.getElementById("kw_imgDIV"+i);i++){
        eval("d.kw_visDiv"+i+"=d.getElementById('kw_imgDIV'+i+'').style")
    }
}
function KW_cScroller(s){ // v1.0.0 Copyright Kaosweaver
    var KW_inc,iL;d=document;iL=KW_iLdr;KW_inc=0;for(var i=0;d.getElementById("kw_imgDIV"+i);i++){
        eval("d.kw_visDiv"+iL+"=d.getElementById('kw_imgDIV'+i+'').style");KW_inc+=KW_csIMG[iL].W+s;
        iL+=1;if (iL>(KW_csIMG.length-1)) iL=0; if (KW_iL<=-(KW_csIMG[KW_iLdr].W+s)){
            KW_iL=0;d.KW_seq+=1;if (d.KW_seq>KW_csIMG.length-1) d.KW_seq=0;KW_iLdr+=1;
            if (KW_iLdr > (KW_csIMG.length-1)) KW_iLdr=0;}KW_iL-=1
    }
}
```

The JavaScript has two parts, the declaration part and the functions. The declarations include everything up till the KW_initScroll function. The declarations page includes variables which are used throughout the script to hold the current position of the layer, image sizes, image scroll speed and image spacing. The declarations section also starts the scroller.

The line which starts with KW_csIMG[0] is the start of the image list. These lines will show how many images are in the scroller gallery (this one has 4 total). The 348 is the image width and the 248 is the images height. If you wanted to add another image, another line would have to be entered here. If you wanted to delete an image, make sure you select the right image set and then change the [n] for all images after the deleted one so as to have a continual incremented image list (0, 1, 2, 3 and so on). Other changes will also need to be made as well.

KW_initScroll(5,10) tells the scroller to have 5 pixels between the images and the 10 is the movement variable (go fast). If you want to change the speed of the scroller, this is the first place that needs to be changed. The second argument (10) is the one that should be changed.

That is it for the JavaScript, everything else should be left alone unless you're really proficient in JavaScript. The next thing we will look at is the layers (positionable div tags) on the page. This is an example of the <div>s:

```
<div id="kw_imgDIV3" style="z-index: 1; left:1044px; top:0px; visibility: visible; width: 348px; position: absolute; height: 248px">
  <a onmouseover="KW_endScroll();" onmouseout="KW_initScroll(5,10);" href="" target=_blank>
    
  </a>
</div>
```

As can be seen this is just an absolutely positioned layer with the style included on the <div> tag's style attribute. The left position is calculated based off of the left position of the placement of the gallery plus as many images that are between the first image and this particular image. The top property is set to equal the top property in the extension. The width is set to the image's width and the height is set to the image's height (each image has its own div). It is set to visible and position: absolute so it can be seen and is a positioned layer.

Within the layer is an image surrounded by a link. To change the link for the image, just change the href property for the a tag. If you want the URL to be in the same page, remove the target=_blank.

To change an image, change the URL in the tag and then change the width and height in both the and <div> tags. Once this is done, change the image properties in the KW_csIMG list. If you've added an image which was wider than the prior image, you'll need to add the difference to each left property in the <div> tags that come after the changed image <div> tag.

To add an image:

1. Copy the <div> tag from the last image on the Continual Scroller.
2. Paste the <div> tag after the last image taking care to keep it within the overall container <div>.
3. Add the width of the image from the last image to the new last <div>'s left property.
4. Change the image URL, width and height in the tag.
5. Change the URL in the <a> tag.
6. Change the width and height properties in the <div> tag.
7. Add the KW_csIMG entry after all of the other entries, increasing the [n] number by one and changing the width and height settings for the entry.

Troubleshooting

Thank you for purchasing the Kaosweaver *Continual Scroller*. We hope you've found this documentation helpful and we hope the extension is profitable for your business or personal web design endeavors. We hope we've covered all of the possible options, questions or usage issues with this manual. We have support available, either via email, the Kaosweaver forum or our ticket system, in the event we've not explained something sufficiently for you.

We want to stress, please send us an email message before investing hours working on this extension if you encounter a problem not covered in this document. 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 and product enhancements. In order to provide the best support possible for you please provide us with as much detail as possible if you should encounter an issue or problem using our extension. It would be especially useful to receive a copy of your file; preferable zipped, with a brief note outlining your errors.

The bullet points below 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.

We would like to add a list of items that we think will help us more efficiently assist you:

1. Always restart Dreamweaver after installing an extension.
2. Always work from a saved page that is part of a site. An Unconnected page won'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 the minimum elements required to accomplish the same task
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 which 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 not stated, we will need to contact you for the information before even looking into the problem.

Following these suggestions will assist us in providing you with the quickest solution to the problem. We want to solve whatever problem you've encountered with our software. If we have the opportunity to fix the problem, we both win. We get the chance to fix a bug we don't know about and you get what you paid for.

Credits

This program was made possible by the input of several important people. Many suggestions came in from the owners of Calendar Popup and that prompted the creation of this product. Some people really stood out and I'd like to thank them here:

- Samantha Davis, my wife. Without her sacrifice and understanding of my time to develop software, Kaosweaver itself wouldn't exist. I am forever in debt to her and I am so blessed to have her as my wife!
- Barbara O'Neal – from <http://www.centricweb.com/>. Barbara provided some early beta testing for the product.

Finally I give credit to God who has blessed me with the skill and ability to do what I do.

Thank you for purchasing *Continual Scroller* and let us know what else you'd like to see in an extension! I'd love to hear from you at

kaosweaver@kaosweaver.com

Paul Davis