



www.walterzorn.de

[Home](#) [VectorGraphics-Library](#) [Tooltips](#) [Function-Grapher](#)



JavaScript: DHTML API, Drag & Drop for Images and Layers

JavaScript Library. Developer: Walter Zorn

A Cross-browser JavaScript DHTML Library which adds Drag Drop functionality and extended DHTML capabilities to layers and to any desired image, even those integrated into the text flow.

To convert images or layers into draggable DHTML items, simply pass their names/IDs to the library's main function 'SET_DHTML()'. Optional [commands](#) allow to change and customize the behaviour of Drag&Drop items in multiple ways. For example, you can limit how far a drag&drop item can be moved, specify the cursor, or multiply drag&drop images. The DHTML API of this DHTML Drag&Drop JavaScript is easily understandable. It provides methods to [moveTo\(\)](#), [resizeTo\(\)](#), [hide\(\)](#) and [show\(\)](#) again drag 'n' drop elements, or to [copy\(\)](#) images right within the textflow of your page dynamically, and [much](#) more. Each DHTML item has properties such as x, y, w, h, z, defx, defy, defw, defh, defz (co-ordinates, size, z-index, and their initial default values, respectively) plus multiple others, which you can read whenever desired. For instance, to store the current position of a drag&drop item, you might write its x and y properties into a <input type="hidden"> form element, from where you could transmit them to the server. For more details, see the [DHTML API](#) and [commands](#) reference.

The idea behind wz_dragdrop.js was not merely to drag around some layers or images on a page in IE only, but also to be a cross-browser clientside API for interactive webpages and webbased applications.

[Cross Browser](#)

[How to include](#)

[DHTML API](#)

[Download](#)

Button

wz_dragdrop.js provides extended DHTML capabilities...

...even for images nested into layers (DIV elements).

For example, just a few simple JavaScript lines are necessary to implement the functionalities of this button.

[Jump to another demo-layer.](#)

Resize Instead of Drag?

Holding the <Shift> key [try it](#) while dragging an element lets wz_dragdrop.js work in Resize Mode rather than in Drag Mode. This functionality can be easily activated by passing to SET_DHTML() the [command](#) RESIZABLE, or SCALABLE to preserve the width/height ratio of the element. These commands may be applied, at your option, either to certain or to all of the drag drop elements. By the way, on these pages the RESIZABLE functionality has been activated globally.



On my 8 megameters bicycle trip through Norway, Northern-Finland and Sweden (Hamburg-Northcape-Kirkenes-Trelleborg-Munich)

Cross-Browser Functionality

Allmost all browsers (see the following list), except of a few rarely used ones, should be able to interpret this DHTML Drag 'n Drop JavaScript.

Linux:

Browsers based on the Gecko-engine (Mozilla, Netscape 6+, Galeon...), Konqueror 3.2+, Netscape 4, Opera 5+.

Windows:

Gecko browsers (Mozilla, Netscape 6+, Phoenix...), Netscape 4, Opera 5, 6 and 7, Internet Explorer 4+.

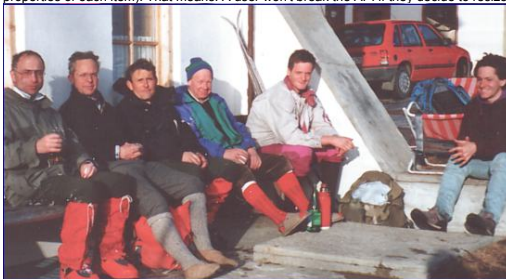
Other Systems:

Mac Safari works fine. Assuming that their behavior is essentially the same as with their Linux and Windows counterparts, I've given Gecko browser Netscape 4 and Opera 5+ unlimited access to execute the Drag and Drop JavaScript.



Window Resize Events

wz_dragdrop.js automatically re-integrates images and relatively positioned layers into the logical structure of the page. That is, back into the page flow even if line-breaks, table-sizes etc. have changed. Items that have previously been dragged keep their relative offset from their - also recalculated - default location within the page flow (accessible, by the way, through the defx and defy properties of each item). That means: A user won't break the API if they decide to [resize](#) their browser window.



Ski touring in South Tyrol

[Top of page](#)

[Cross Browser](#)

[How to include](#)

[DHTML API](#)

[Download](#)

How to include the Drag & Drop Script

1. HTML File: Names for Drag&Drop elements



Images: Each of the images to be set draggable requires a unique name, as for instance:

Width and height attributes are mandatory and should be absolute numbers like width="240", rather than relative ones like width="33%". **Layers:** Each one requires a unique ID and, contrary to images, must be positioned relatively or absolutely:

<div id="name2" style="position:absolute;...">Content</div>.

[Top of page](#)

[Cross Browser](#)

[How to include](#)

[DHTML API](#)

[Download](#)

2. Insert pieces of code into HTML file

Insert the following lines inside the <body> section of your html file.
This one immediately **after (not before!!!) the opening <body> tag**:

```
<script type="text/javascript" src="wz_dragdrop.js"></script>
```

And this one directly before the closing </body> tag:

```
<script type="text/javascript">
<!--
SET_DHTML("name1", "name2", "anotherLayer", "lastImage");

//-->
</script>
```

As evident from this example, simply the names of the drag-drop elements must be passed to SET_DHTML(), each in quotation marks and with commas separated from each other. Sequence of names may be arbitrary.

3. Save script file

[Download the JavaScript Drag Drop library](#), unzip it and save it as wz_dragdrop.js into the same directory as the html file. Or adapt the path src="wz_dragdrop.js" within the above code piece if you prefer to save the library to a different directory.

4. Save transparent 1x1 pixel GIF

transparentpixel.gif, coming with the download file wz_dragdrop.zip, is required by Netscape 4 to create spacers at the default places of the drag drop images. transparentpixel.gif should be saved into the same directory as the html file.

At the Northcape, on my second Scandinavia bicycle trip

5. Doesn't work?

Then everything should function. If not:

- Check the HTML and CSS syntax of your page, preferably with the W3C-Online-Validator on <http://validator.w3.org/>.
- Make sure that you've carefully followed [steps 1. through 4](#) (by far the most frequent reason for problems).
- Check your own JavaScript code for logical errors and syntax errors, the latter e.g. by looking at your browser's JavaScript error console.
- Occasionally, problems arise with XHTML DTD pages, since the XHTML specification - annoyingly - doesn't permit document.write(). Then it's inevitable to switch your page to another DTD, preferably HTML4.01.

Don't claim you've found a bug - unless you have everything checked carefully and you're very sure. Feedback is anyway welcome.

[Top of page](#)
[Cross Browser](#)
[How to include](#)
[DHTML API](#)
[Download](#)

DHTML API**Clone or multiply an image?**

Or change the cursor? Or convert an image into a slider with stops? There are several commands available which change, if passed to SET_DHTML(), the behavior of JavaScript Drag n' Drop items.

`swapImage()`

Scripting Drag Drop Elements

For example, reading the current coordinates of a certain drag-drop item and then calling its moveTo() method offers an easy way to animate images and layers. Moreover, you may even convert a simple web page, just containing a few images, into a graphical user interface application, working with the coordinates of the elements (as demonstrated with the slider example above)...

The DHTML Drag Drop Library provides an easily understandable, user-friendly interface to the properties and methods of the drag-drop elements. Using this library as DHTML API you haven't to bother with browser or DOM detection - all this is performed by the library. For instance, the line

```
dd.elements.image2.moveTo(300, 200);
```

would reliably move "image2" to the absolute coordinates (300, 200) in any browser that can interpret the library, as reliably as the expression dd.elements.image2.x will retrieve the current X coordinate of the image.

- [Optional Commands](#): Demonstration and Reference
- [DHTML API, Scripting Drag Drop Items](#): Demo and Reference
- [Examples](#) using the DHTML API
- [History of Updates](#) (read this if you aren't sure about the benefits from an update)

Download

wz_dragdrop.js 4.91, published under the **LGPL**:
[wz_dragdrop.zip](#) (12 KB)

Attention:

- If you use wz_dragdrop.js for applications or for extended scripting, and are going to upgrade from a version prior to ver. 3.4, please consider: With version 3.4 the array that contains the drag drop elements has been renamed from dd.element to dd.elements. You must change your code accordingly in case you've used any dd.element statement (see also [Scripting Drag Drop elements, DHTML API](#)).
- With version 4.0, the library has been renamed from dragdrop.js to **wz_dragdrop.js**.
- With version 4.2, the method activate() has been renamed to maximize().

[Top of page](#)
[Cross Browser](#)
[How to include](#)
[DHTML API](#)
[Download](#)

4.) With version 4.31, the command `MULTIPLY` has been renamed to `COPY`, and its usage has been changed as follows: To create 12 copies of an image named "imageName", you must write "imageName"+`COPY+12` instead of "imageName"+`MULTIPLY12`

5.) With version 4.41, `SET_DRAGGABLE()` has been renamed to `SET_DHTML()` and `ADD_DRAGGABLE()` to `ADD_DHTML()`.

Walter Zorn, Munich
[Impressum](#) / [about this page](#)