

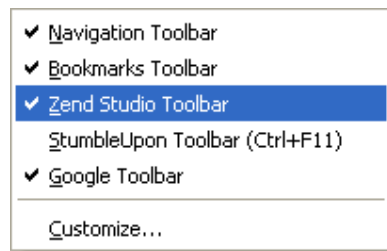
# Installing the Zend Browser Toolbar

## The Zend Toolbar

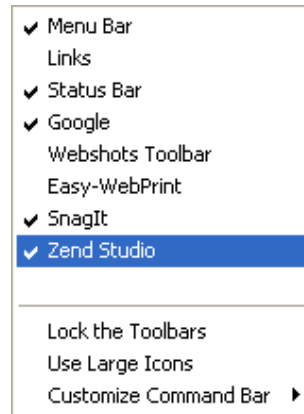
The Zend Toolbar provides an easier way to debug or profile your sites and applications simply by initiating debug sessions directly from the browser and browsing the different development versions of your projects.

## Automatic Installation

Installation of this extension is automatically offered when installing Zend Studio. The user simply has to check that the toolbar is activated by making a right click in the menu bar of the browser



*With Firefox*



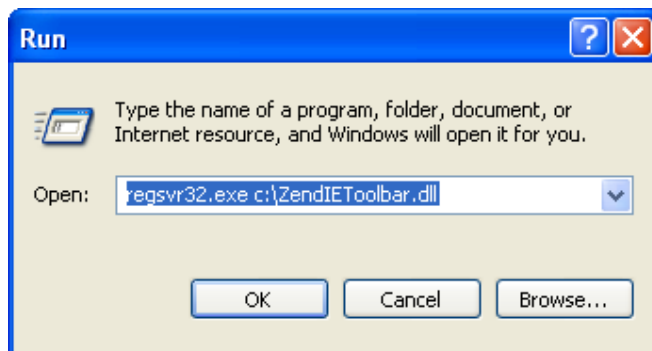
*With Internet Explorer*

## Manual installation of the toolbar

The user may also install or re-install the toolbar manually. This is very simple with both Internet Explorer and Firefox:

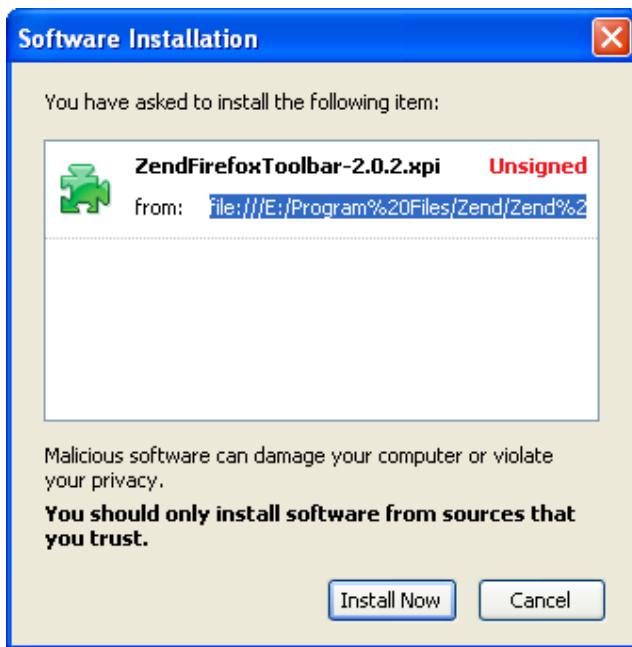
### For Internet Explorer

Download the plug-in from <http://www.zend.com/products/studio/downloads>. Then, using « Start - >Run », execute this command: `regsvr32.exe [full path to]ZendIEToolbar.dll`



### For Firefox

Download the .xpi file for the Firefox plug-in from <http://www.zend.com/products/studio/downloads> and allow the installation of the extension. Once the extension is installed, restart Firefox.



## Configuration

In most cases the Zend Toolbar doesn't require any configuring (the default configuration means it is immediately functional in most situations)

Most of the options in the toolbar mirror the debug preferences defined in Studio, either for all the software or for the configuration of each individual project (for which the parameters linked to the debugging may be customized.)

To find out more about this subject, refer to the Zend Studio documentation which has a complete chapter on the configuration options of the debugger (the documentation is in the "docs" sub-folder of the installation directory).

## Using the Zend Toolbar



There are various ways of starting a debug session from the toolbar:

### « Debug » or « Debug Current Page »

This is the default option and is the most easily accessed (there is a button in the toolbar). A click on this button will start the debugging of the page currently shown in the explorer. To do this, the page is reloaded, and instructions are automatically transmitted to the server for the execution to be done in debug mode.

This means that if the POST data has been transmitted to the current page, the browser will ask if the user wishes to post them again before executing the page in debug mode. In the same way, if the page uses *frames* (including hidden ones), the toolbar will request which *frame* the user wants to debug.

### « Next page on site »

This second option is also available, as are all the following ones, in the debugging sub-menu. It will automatically start executing the next page consulted in debug mode. This means that the debug session will be started once a form has been validated or a link clicked, or an Ajax request has been executed.

### « All forms (POST) on this site »

By selecting this option, the debugging will be activated once a POST form has been validated. This means that the user will debug the script designated as the action of their form (only if they use the POST method)

Note that this option will also provoke a debug session if an Ajax request using the POST method is solicited.

### « All pages on this site »

As implied by the name, this option provokes debugging of each page on the site. Once again, this concerns the Ajax request. This option is particularly useful when the user wants to debug an Ajax request which may itself be solicited by another Ajax request.

### « Profile »

This button, like the button « Debug » allows the user to restart the execution of the current page, but with this time in order to profile the page and not to debug it.

The result of profiling will be shown in Studio and it allows the user to see in detail which files and/or functions use the resources required for generating the current page. This enables the user to track down any bottlenecks which slow down the generation of certain pages.