Content Debugger – manual

Plugin	content_debugger (for Elgg 1.7+)
Version	1.0.0
Author	András Szepesházi
License	GNU GPL v2.0

Intended audience

This is a plugin for Elgg theme/plugin developers. Site owners, who do not customize their Elgg site other than downloading and installing plugins, will get nothing from this plugin.

Introduction

This plugin will give you on-screen information about the current page's view hierarchy. Upon hovering over any part of the browser window, you'll see a tooltip, displaying detailed information about what views were called - at the server side - to produce the currently visible output (see Figure 1).
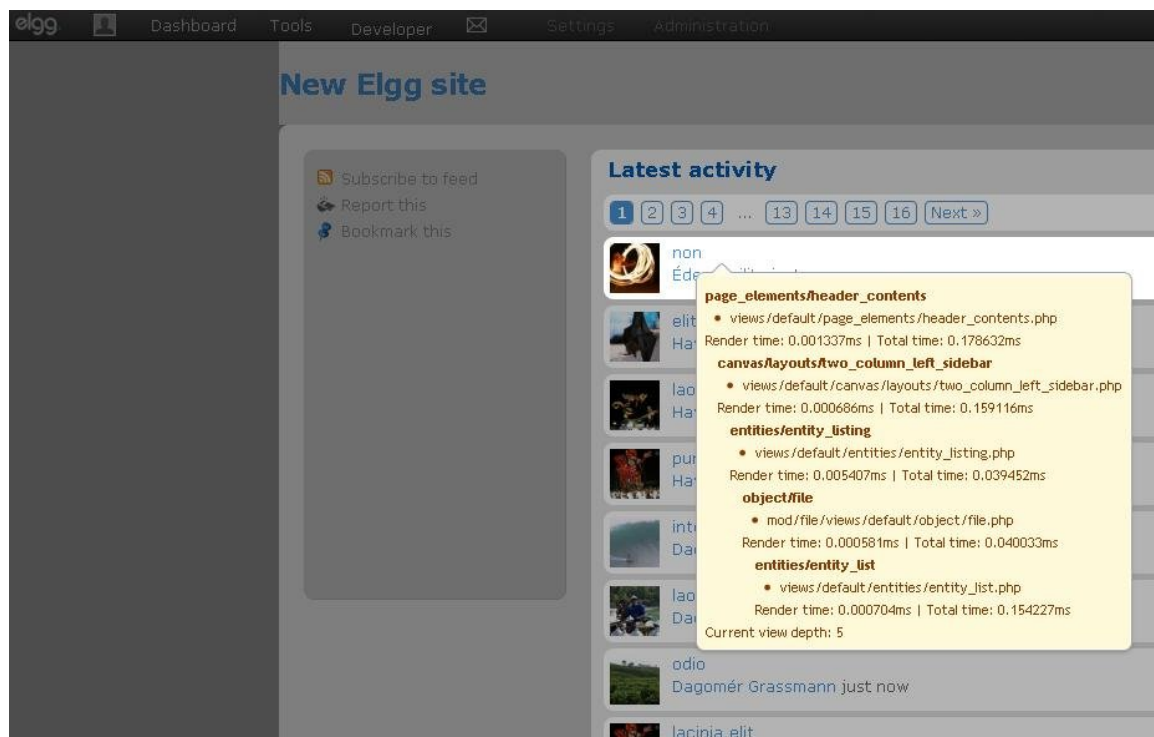

Figure 1 - Content Debugger in action

I think the plugin will provide the greatest benefits to those developers who need to work on a heavily customized site. With Elgg's view override and view extension system, it is very easy to get lost among dozens of views that produce a given piece of output. The Content Debugger will put an end to your hair-pulling hours of trying to figure out which view was extended or overridden by which other view. You'll get all this information, literally, at the tip of your mouse.

Requirements

- You'll need the (preferably) latest release of the Elgg 1.7 branch for this plugin to work. I've tested it with 1.7.6 and 1.7.8, and it proved to be working fine. I strongly suspect that it will work with anything from 1.7.0 upwards, but I haven't tried it.

- It will definitely not work with Elgg 1.8. The plugin extends a view that is nonexistent in Elgg 1.8, so you *won't be able to turn the plugin's functionality on*. Other than that, it would work great in 1.8. I promise to upgrade the plugin as soon as 1.8 comes out.

- You'll need a CSS3 compliant, decent browser for this feature to work. I suggest Firefox 3.6+ or latest Chrome or latest Safari. As this is an admin/developer only feature, I don't think that non-compliant browsers should be supported. If you're wondering about my intentions to support any release of IE, please refer to the FAQ. Short answer: no.

Installation

Installing the plugin is pretty straightforward, as with all Elgg plugins. Please refer to the official Elgg documentation at http://docs.elgg.org/wiki/Configuration/Plugins

Configuration

After installing the plugin, you'll have a number of configuration options to fine-tune the plugin's behavior. You can access these options by going to Administration -> Tool Administration -> content_debugger -> Settings (see Figure 2).
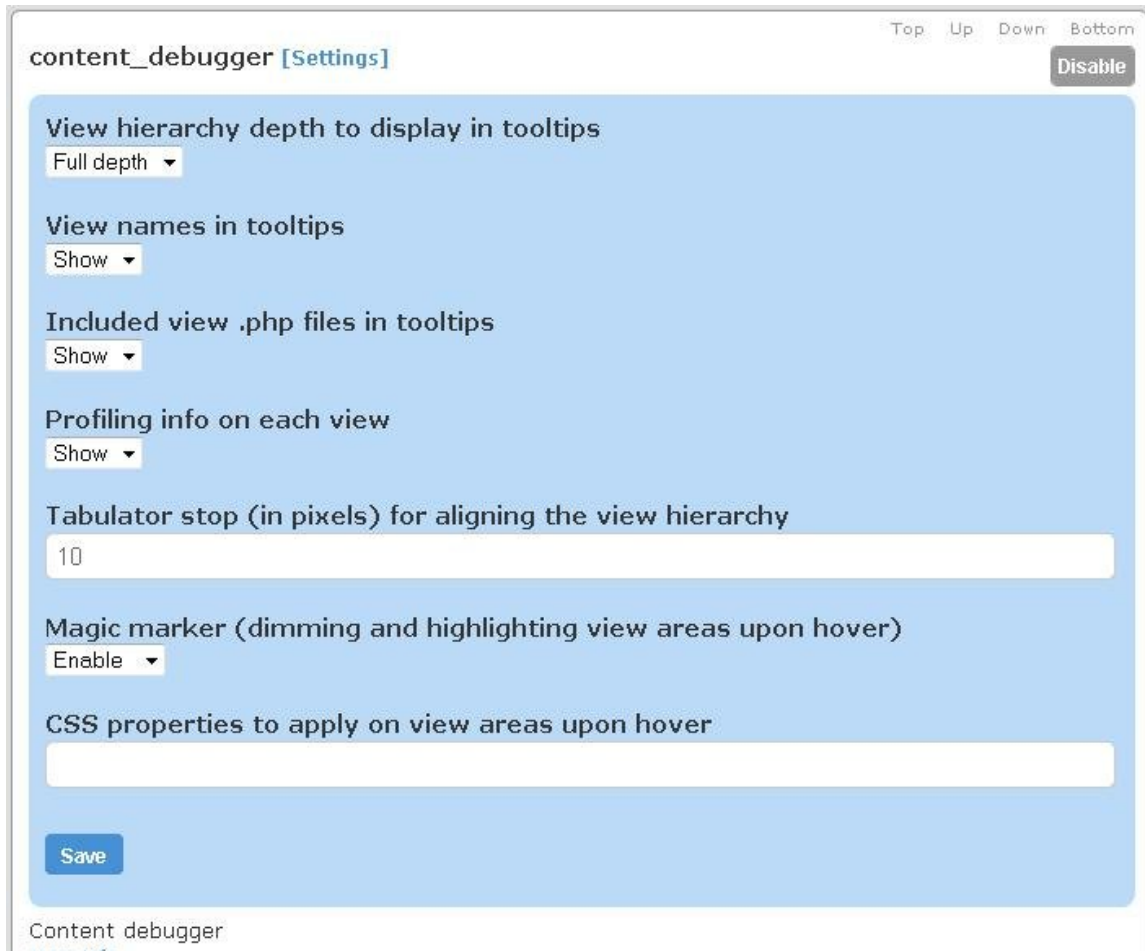
Figure 2 - Content Debugger configuration

View hierarchy depth to display in tooltips

Sometimes, when you have a really complex site, the view hierarchy tree can grow to a depth of 15-20, or even more views for certain visible blocks. That many level will simply not fit into the hovering tooltip, so this is your chance to reduce the number of view levels. If you limit the number of view levels to be displayed by the tooltip, it will always cut off the uppermost views, so you'll still get the information you're really interested in, for any given block.

View names in tooltips

All views in Elgg have a distinct name, like "object/file" or "entity/entity_list". With this option you can decide if the view name would be displayed or not in the tool tip.

Included view .php files in tooltips

All views in Elgg are produced by view php files like "/mod/files/object/file.php". With this option you can decide if the view file path would be displayed or not in the tool tip. When extending a certain view, the view extensions will be included by Elgg's templating view system. In that case,

4

a single view name will contain more than one actual view files. You'll see all the included view extensions under the given view name. See Figure 3 for an example of displaying view extensions (the owner_block/extend view is extended by two different view files).
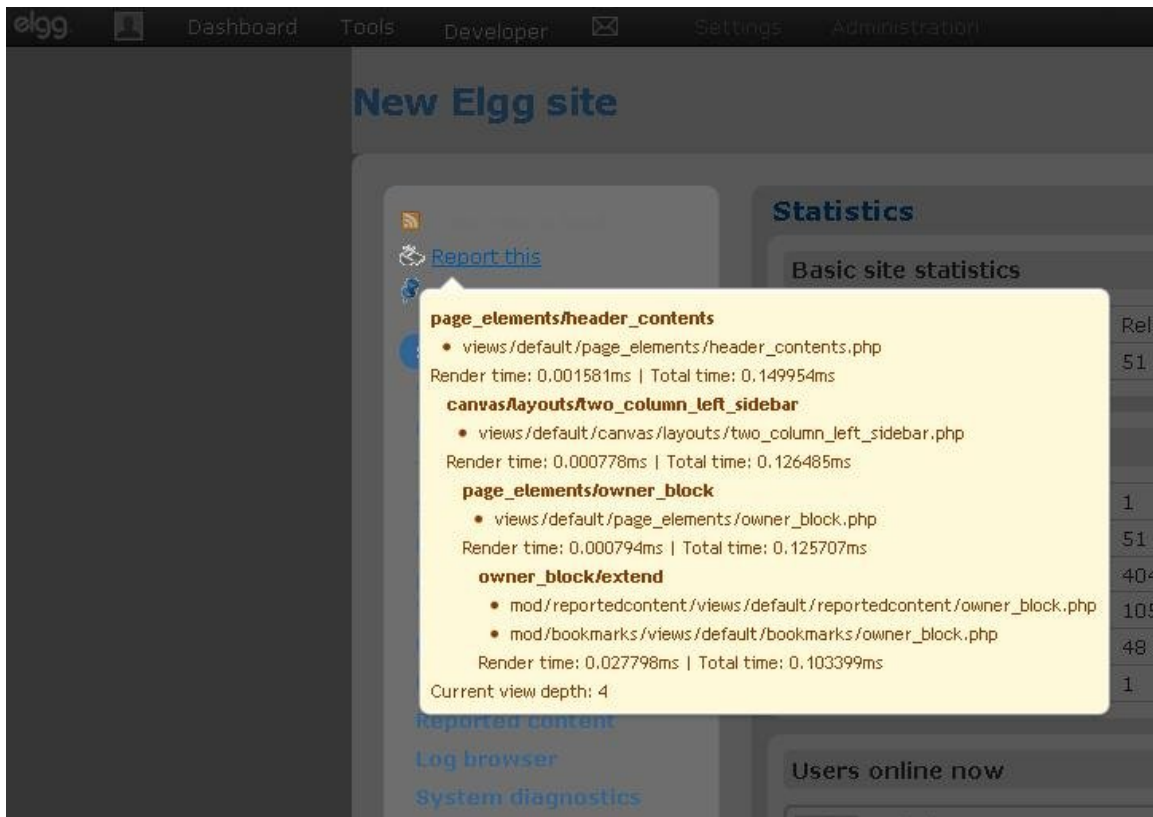

Figure 3 - Displaying view extensions

Profiling info on each view

For each displayed view, you can enable or disable profiling information with this option. Profiling will show you roughly how much time it took to produce a given view (this is the "Render time" value) and how much time took the Elgg engine to produce all the views up to the current view. These values are not entirely accurate for various reasons, but they should still give you a good indication which views are resource intense.

Tabulator stop (in pixels) for aligning the view hierarchy

This is to configure the tabbing of the view hierarchy in the displayed tooltip. Enter an integer value without any further additions (so no "px" or "point" after the number). Going down the view hierarchy, each view level will be shifted by the amount of pixels you enter here.

Magic marker (dimming and highlighting view areas upon hover)

Enabling this effect will dim all view areas that are currently out of focus. The dim is applied hierarchically, so the further down you go in the view hierarchy, the more the higher level view areas will get dimmed.

CSS properties to apply on view areas upon hover

You can enter here further css properties that will be applied to the current view area. For example, entering: `border: #c04030 dashed 2px; background: #e0a090;` for this option will result in what you can see at Figure 4. Please make sure you enter valid css options separated by semicolon – having an error in your css syntax can result in broken plugin functionality.
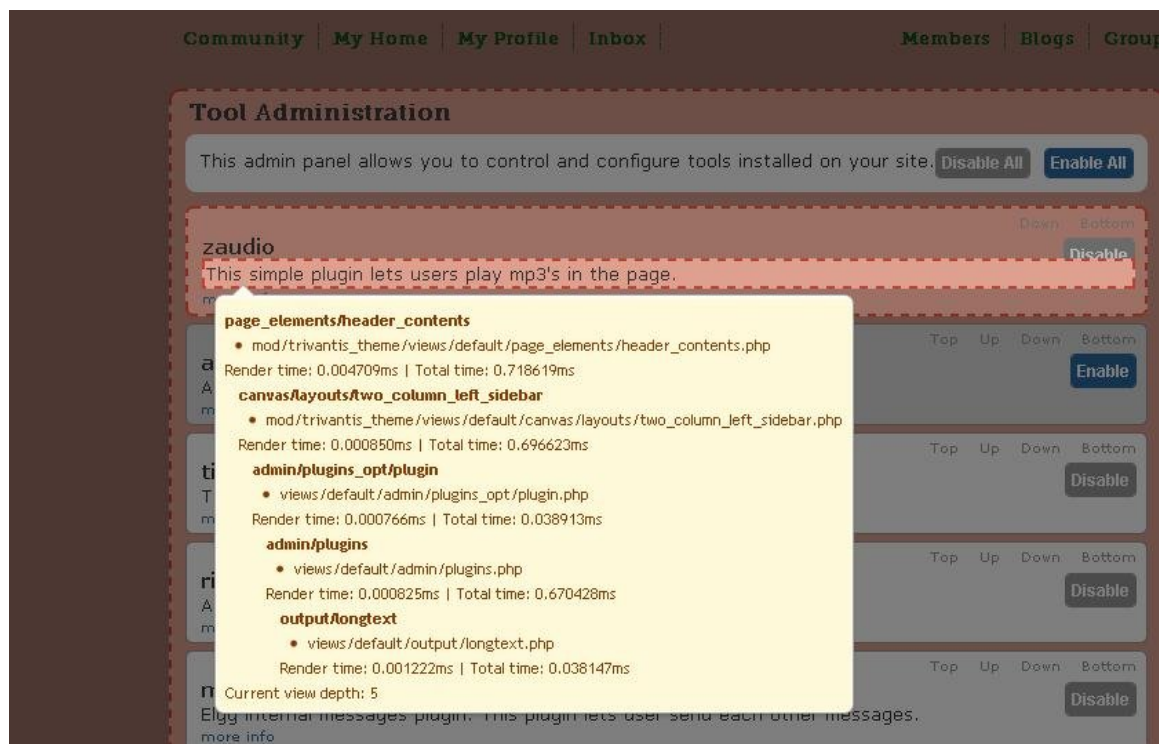

Figure 4 - Custom CSS for active view areas

Using the Content Debugger

Once you enabled the plugin, and are logged in as administrator, you'll get an extra menu item in your Elgg top bar called "Developer". Hovering over this menu will roll down a submenu, where you get "Toggle Debugger". Clicking on this will reload the current page and enable/disable the debugger (see Figure 5).

Figure 5 - Activating the Content Debugger

Troubleshooting and FAQ

Q: This plugin doesn't work in IE6, IE7, etc

A: Sorry to be rude, but couldn't care less. (Rant: This is a personal revenge on my side and yes, it gives me satisfaction. In recent projects of mine some customers insisted on having very advanced js/css functionality, but at the same time compatibility with IE6 and IE7. I spent countless hours pulling my hair at the horrible hacks this requirement forced me to introduce to my code.) Anyways, as this is an admin only feature, I think it's sensible to introduce browser requirements. All this plugin asks for is a standards compliant decent browser, like any gecko or webkit based browser.

Q: I'm using a decent browser, enabled the plugin, turned on the content debugger feature via the Toggle Debugger menu, and I still don't get any tooltip showing the view information.

A: I'd say this is probably due to a plugin conflict. The Content Debugger overrides the page_elements/header view to make this functionality work. If another plugin of yours overrides the same view, and comes later in the plugin order than Content Debugger, then this plugin will not work.

Q: Why do you override the page_elements/header view? It's a bad practice to override core views.

A: That is true and I'm very sorry about my misbehavior. However I need jQuery 1.4 for the advanced features this plugin uses for data visualization. And the only way to include jQuery 1.4 instead of the 1.3 (that is shipped with the Elgg 1.7.x branch) is to override the page_elements/header view.

Q: Your plugin breaks another plugin's functionality!

A: That is possible, if the other plugin also overrides the page_elements/header view (see previous question). Disable the other plugin, or disable this plugin, or try to merge the two views into a single view.

Q: This does not display the top level views (like page_elements/pageshell) and the views in the header (metatags, etc).

A: Unfortunately that is true again. I couldn't figure out a cross browser way to parse and display view information that is beyond the document body level. However, if you look at your page's source code, you'll see that all view (including pageshell and header) information is there in forms of json encoded comments. Look for "content_debugger_view_start::" and "content_debugger_view_end::" patterns. What follows these markers is a json encoded string holding all the information of any given view. It's a bit inconvenient, but if you really need the view hierarchy for elements outside the document body, you can figure it out from these comments.

Q: I added a css border value in the plugin configuration, and now the whole screen falls apart when I move around the mouse.

A: That is possible when you have fixed sized block elements in your template, and the current content already fills those containers exactly to the last pixel. When you add a border to that, the content size will increase, and you'll get a broken layout. Unfortunately, the Elgg 1.7 default layout uses a lot of these fixed size containers and contents. However I've worked with a number of better behaving custom layouts, that will survive the addition of a 1-2 pixel wide border.

Known bugs

- The plugin does not give information about top level views that fall outside the document body.