Access Level Violations

Marking Access Violations

htmlOKay deals with access violations by marking up a page in the places where the access violations occur. Examples of this markup for each of the access levels are found in the following pages, which show the same HTML but at different access levels:

- Level "L"
- Level "M"
- Level "H"

The following page has two access settings, **M** and **L:**

- Levels "M" and "L"

A page with multiple access levels will always display to the visitor at the most restrictive access level, which in this case is **M**. If you click on the **Developer's toolbar**, and then click on the **Refresh** button, the page will display at **M**, if it is not already at **M**. If you then log on as *developer* with the password *developer*, the page will display as **L**. When you log out, the page will still be at **L**, but if you then hit the **Refresh** button again, it will revert to **M**. See the section on the Refresh Button later on this page.

Errors Window

In addition to the marking up of access violations, there will also be at the top of the page links for showing and hiding a window containing more detailed error messages, as in the example below:

show errors close errors

User Info:
hmtlOK_access_level: 2
client: wsmith
$conf['htmlok']: 1
Scope: html
---End User Info---
ID Selectors not supported at current HTML access level: **#htmlO_K_div_1.**
Element or Attribute not supported at current HTML access level: **TABLE**
Element or Attribute not supported at current HTML access level: **div**
Internal Window Elements Not Supported. External files cannot be included in wiki documents: **IFRAME.**

The **User Info** section contains four pieces of information–the current access level of this user, the name of the current user (or 'client'), the value of the *$conf['htmlok']* variable, which should always

be 1 (i.e. true), and the 'scope', which is the namespace under which the current page falls and from which it gets its access permissions. (See the explanation of namespace scope.) For a sample page with both the errors window and marked up access violations, see this page.

Show Errors Button

It is possible to replace the links show errors and close errors
with a custom errors button that conforms in style to the current template. The code is as follows:

```
 <?php if&#40;$INFO&#91;'htmlOK_client'&#93; &&
$INFO&#91;'hmtlOK_access_level'&#93; > 0&#41; &#123; ?>
   <form name="htmlOK_errform">
    <div class="no">
     <input type="button" value="Show errors"
        class="button"
        onclick="htmlOK_show_errorswindow();" />
    </div>
  </form>
   <?php &#125; ?>
```

This code, which goes on *main.php*, prints the button if an HTML user is logged in and an HTML access level exists for the page. You can see this at work, if you log in as user *developer*, using the password *developer*. Then click on the **Developer's toolbar** at the top of the page. You can also go to any other of the sample pages with access violations, and you will be able to open the errors window as well as the marked up violations.

If you do create a custom button, you will want to go the the “Configuration Settings” on the **Administration** page, scroll down to “HtmlOK Plugin Settings” and tick off the box beside this item:

- Check off this box if you are installing a custom error button for display of HTML errors

Refresh Button

More than one access level can be applied to the same namespace. For visitors to such a namespace, **htmlOKay** uses the the Display Level to determine the access level at which to display the pages in that namespace. (For an example See Marking Access Violations above.)

Because DokuWiki caches its pages, the HTML displayed to a visitor may be that of the last cached version of a page, instead of the version assigned to the *display level*. To get around this problem if you see it as a problem, you can create a refresh button, to re-parse the page after the developer has logged out:

```
  <?php if&#40;$conf&#91;'htmlok'&#93;&#41; &#123; ?>
```

```
    <form class="button" method="get" action="/wiki/doku.php">
      <div class="no">
        <input type="hidden" name="do" value="show" />
        <input type="hidden" name="refresh" value="yes" />
        <input type="hidden" name="id" value="<?php echo $ID; ?>" />
        <input type="submit" value="Refresh" class="button"  />
      </div>
    </form>
  <?php &#125; ?>
```

The developer *must* first log out before hitting the refresh button, so that the page refreshes at the *display level*; otherwise the page will simply refresh using the developer's access level.

This button will conform in style to the current template. See the **Developer's toolbar** on this installation for an example of the *Refresh* button.

For an example of how this works, go to this page. Log in as *developer* with the password *developer*. Then log out and hit the **Refresh** button.

From:
http://192.168.0.101/htmlOKay/ - **htmlOKay**

Permanent link:
http://192.168.0.101/htmlOKay/doku.php?id=html:access:violations

Last update: **2007/08/31 07:14**