Applying Access Levels

Introduction

The **htmlOKay** plugin enables the DukuWiki administrator to set permissions on embedded HTML access. Through its Embedded HTML Access Manager, it applies an extra layer of access controls on top of DukuWiki's built-in ACL. The ACL takes priority. Before users can embed HTML in a page, they must first have write access to that page.

The permissions allow for up to four levels of access and can be applied to both users and groups. **htmlOKay** requires the default setting for embedded HTML, which is that HTML is not allowed.[1]. Only users and groups which have been granted HTML access from within the **htmlOKay** Access Manager will have the right to use HTML in their files.

A file which supports embedded HTML can alternate between standard DokuWiki mark-up and HTML mark-up. HTML is included in a DokuWiki file by placing it between the two HTML tags:

```
=====Dokuwiki Standard Header=====
<html>
    <H2>HTML Header</H2>
</html>
```

Objectives

Access levels determine the extent to which users and groups can use various features of HTML. There are two objectives:

1. to provide security
2. to maintain the integrity of the Wiki's page template

Pages which enable embedded HTML inevitably run a security risk, not unlike pages which allow external files to be included by means of the include plugin. And pages with embedded HTML risk inadverently overriding the template's CSS or misaligning page structure. An unmatched

, for instance, inserted into the default template can remove all the formatting from the DokuWiki footer.

The Access Levels

There are four access levels, each with increasing degrees of restriction. These are designated in the most obvious of terms as **H**, **M**, **L**, and **U**. That is, as **H**igh, **M**edium, and **Low** degrees of restriction, and a relatively **U**nrestricted degree.

1. **H** Allows in-line formatting, including the in-line “style” attribute and onclick and mouse handlers in-line. It excludes scripting, forms, tables, iframes, ilayers, divs.
2. **M** Allows restricted use of javascript, of forms and of css. Excludes tables, iframes, ilayers, divs.

3. **H** Allows full use of css and restricted use of forms and javascript. Excludes iframes and ilayers.
4. **U** Allows full range of HTML and Javascript techniques. Users at this level are subject only to the naming conventions descibed below. This is in effect a 'super user' access level. There is a beta version of **htmlOKay** which enables **U** to have completely unrestricted use of HTML through a setting in the Dokuwiki **Configuration Manager**. Simply scroll down to the **htmlOKay** configuration settings and tick off “Unrestricted Super User.” (While technically 'beta', the changes for this version were minor, and it has been operating successfully.)

Naming Conventions

All access levels are subject to the **htmlOKay** naming conventions. All CSS class names, all ID's and all javascript function names must be prefixed by **htmlO_K_**. This prefix helps to assure that there will be no naming conflicts between embedded HTML and any other plugins and the Wiki itself. Here is an extended example which touches on all of these points:

```
<span id="htmlO_K_hidden" class="htmlO_K_bold">I am Hidden</span>

<a href="javascript:htmlO_K_show();">Show</a>
```

It is also possible to set styles in the template's CSS files. The names of classes and id's referenced in the template's CSS files must follow the naming conventions described above.

Feature Tables

The following tables detail the features which are allowed and disallowed and the restrictions placed on certain allowed features for several of the access levels.

Allowed and Disallowed Features for Each Access Level

Y = YES, the feature is allowed. **N** = NO, the feature is not allowed.

Feature	H	M	L	U	Restrictions
DIV	N	N	Y	Y	
FORM	N	Y	Y	Y	**M** and **L**
JAVASCRIPT	N	Y	Y	Y	**M** and **L**
CSS	N	Y	Y	Y	**M**
TABLE	N	N	Y	Y	
IFRAME	N	N	N	Y	
ILAYER	N	N	N	Y	
JAVASCRIPT URLS	N	N	Y	Y	

Restrictions

In the table below are the restrictions placed upon certain features designated as **Y** above. If a technique is listed here, then it is not available at the specified access level. For instance, **M** can use the

element but it cannot use 'action' or 'onsubmit'. This still gives it access to all the other features of the Form element and the Javascript which supports them.

Access Level	FORM	JAVASCRIPT	CSS
M	action, onsubmit	event listeners,captureEvent,createEvent, Ajax, location=url	ID's [#id_name]
L	action	event listeners,captureEvent,createEvent, Ajax, location=url	

How Permissions are Applied

Access permissions are set by namespace. That is, only one set of permissions can apply to the files of any one namespace. Various groups and users can be included in this set of permissions and they can each be assigned different access levels. Stated slightly differently, more than one group or user can have HTML access to the same file or namespace, each with different access levels.

When applying permissions, the administrator has the option of applying them to ALL of the files in the namespace or to selected files in the namespace. Let us assume that there are 3 files in the namespace:

- *namespace:start*
- *namespace:our_organization*
- *namespace:our_policies*.

If the administrator selects ALL from the permissions menu, then the permissions will apply to all three files. But if *namespace:our_policies* is selected, then permissions for files in *namespace* apply only to *namespace:our_policies*. The other two files will not have HTML access. You cannot apply different permissions to different files in the same namepace. However, any user or group may have HTML access to the files in more than one namespace.

The inability to apply permissions on a file by file basis, as opposed to namespace basis, may be seen as a shortcoming. But the original idea behind **htmlOKay** was that the Wiki would be largely organized by projects and that each project would fall under its own namespace. The project would then be developed by a group or a single individual, who would be assigned an HTML access level.

The Display Level

Each page and/or namespace is assigned a **display** level. Normally, this is the same as the access level of the group or user who has created the page. But the Embedded HTML Access Manager will allow administrators to assign users and groups to the same namespace who have different HTML access levels for that namespace. Consequently, **htmlOKay** has a policy of applying the most restrictive permissions when a page is being viewed by an external visitor to the page, i.e. someone without HTML access to that page. If one developer has **M** access and another has **L**, the page will be displayed at **M**.

Moreover, if the user with **L** access uses features that are not supported by the **M** set of features, then those features will not work when **M** goes to add something to the page. It may not seem to make sense to apply more than one access level to a namespace, yet it can have uses when developing a project. We will look more closely at these issues when dealing with the Access Manager.

[1] This means that the “Allow embedded HTML” option in the Configuration Manager is left unchecked.

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

Permanent link:
http://192.168.0.101/htmlOKay/doku.php?id=htmlok

Last update: **2007/11/08 07:22**