Ephox Editors for Drupal Integration Installation Instructions

Files Needed

Before you can get EditLive! running in Drupal you need to download the installation files from this project's GitHub:

http://www.github.com/projectNameHere

Separate from the project download, you need to download the Ephox editor you need (Textbox.io or EditLive!) from the Ephox product download page:

http://ephox.com/download

Prerequisites

The use of the Ephox Editors for Drupal integration requires that you have the following:

- Drupal 7
- WYSIWYG module (https://www.drupal.org/project/wysiwyg)
- A supported Ephox Editor
    - o Textbox.io 1.2+
    - o EditLive! 9+

Installation Steps

1. Install the ephox_image_upload **module**

The **ephox_image_upload** module needs to be installed in order to allow local images to be uploaded to the Drupal server. Installing this module should be no different than installing any other Drupal module. To install the module…

a) Copy the **ephox_image_upload** file – this will result in the existence of a single **ephox_image_upload** folder that contains 2 files.

b) Copy the **ephox_image_upload** folder to the **/sites/all/modules** folder within your Drupal installation. The **ephox_image_upload** folder should be in the same directory as any other modules you have installed such as the **wysiwyg** module:

   Note: If you use another location for contributed modules place the **ephox_image_upload** *folder in the location where you install contributed modules*

c) Login to Drupal

d) Go to the Modules page (`http://<server>/admin/modules`) and enable the module called "*Ephox Editors Image Upload Acceptor*". The module should appear in the User Interface grouping in the modules list.

ENABLED	NAME	VERSION	DESCRIPTION
☑	**Ephox Editors Image Upload Acceptor**	1.0	Manages the upload of local image content from Textbox.io and EditLive

▼ USER INTERFACE

e) Click **Save** to accept the enabling of the *Ephox Editors Image Upload Acceptor* module.

2. Add the editor definition files to the WYSIWYG module

Implementing Textbox.io or EditLive! requires you to have the **WYSIWYG** module previously installed. In this step we are going to add 2 files to that module that define either Textbox.io or EditLive! to the **WYSIWYG** module.

a) Locate the **WYSIWYG** module in your Drupal server installation. Modules are typically installed in the **/sites/all/modules** folder. The folder name for the **WYSIWYG** module is **wysiwyg** so the complete path would be **/sites/all/modules/wysiwyg**

   *Note: If you use another location for contributed modules the **wysiwyg** folder will be in the location where you install contributed modules*

b) Locate the appropriate **.inc** file from your download and place this file in the **editors** folder within **WYSIWYG** module. If you wish to use Textbox.io this file is called **textboxio.inc**. If you wish to use EditLive! this file is called **editlive.inc**.

   When you copy the file to this folder you should already see several other existing **.inc** files in this folder. Those are the editor definitions provided with the **WYSIWYG** module.

c) Locate the appropriate **.js** file from your download and place this file in the **editors/js** folder within **WYSIWYG** module.

   If you wish to use Textbox.io this file is called **textboxio-1.js**. If you wish to use EditLive! this file is called **editlive.js**.

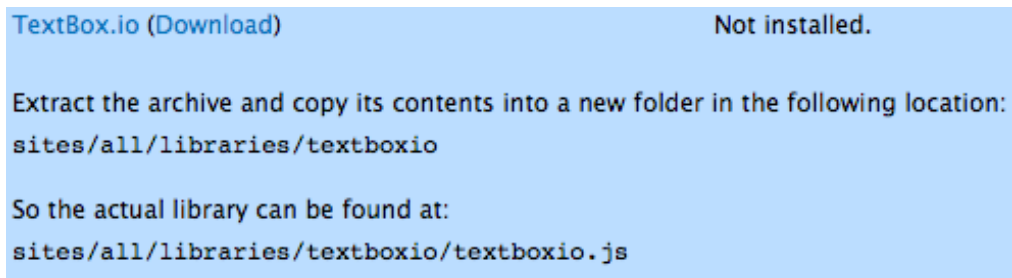## 3. Configure WYSIWYG to use the appropriate Ephox Editor

In order to use either Textbox.io or EditLive! as your editor we need to perform several configuration steps to make the editor available within the WYSIWYG module.

3a. Configuring Textbox.io

a) Navigate to the configuration screen for the WYSIWYG module

```
http://<server>/admin/config/content/wysiwyg
```

b) Click on the **Installation Instructions** link – you should see Textbox.io as an editor but it should show "Not Installed".

c) Follow the **WYSIWYG** module admin screen instructions to get the editor "installed" properly.   The module should provide you some details on where to place Textbox.io's files in order to make them available to the **WYSIWYG** module. On our test system the instructions were as follows:

TextBox.io (Download)   Not installed.

Extract the archive and copy its contents into a new folder in the following location:
sites/all/libraries/textboxio

So the actual library can be found at:
sites/all/libraries/textboxio/textboxio.js

(We will assume these paths for the remainder of these instructions)

d) Create a directory named **textboxio** in the **/sites/all/libraries** directory.

e) Copy the Textbox.io runtime files to the **/sites/all/libraries** directory.

   a. When you download the Textbox.io Client SDK from the Ephox web site you get a ZIP file.
   b. When you unzip that file you get a **textboxio** folder.
   c. Copy the **textboxio** folder into the **/sites/all/libraries** folder
   d. When done properly the main Textbox.io JavaScript file would be at **/sites/all/libraries/textboxio/textboxio.js**

f) Refresh the WYSIWYG configuration screen and click on the **Installation Instructions** link – now EditLive! should show as installed and a version number should appear on the right side of the row

TextBox.io (Download)	1

This signifies that Textbox.io was installed to the correct location and that the WYSIWYG module has located the files.

3b. Configuring EditLive!

a) Navigate to the configuration screen for the WYSIWYG module

`http://<server>/admin/config/content/wysiwyg`

b) Click on the **Installation Instructions** link – you should see EditLive! as an editor but it should show "Not Installed".

c) Follow the **WYSIWYG** module admin screen instructions to get the editor "installed" properly.   The module should provide you some details on where to place EditLive!'s files in order to make them available to the **WYSIWYG** module. On our test system the instructions were as follows:

EditLive (Download)   Not installed.

Extract the archive and copy its contents into a new folder in the following location:
`sites/all/libraries/editlive`

So the actual library can be found at:
`sites/all/libraries/editlive/editlivejava/editlivejava.js`

(We will assume these paths for the remainder of these instructions)

d) Create a directory named **editlive** in the **/sites/all/libraries** directory.

e) Copy the EditLive! runtime files to the **/sites/all/libraries/editlive** directory.

a. When you download the EditLive! SDK from the Ephox web site you get a ZIP file.
b. When you unzip that file navigate to the **/webfolder/redistributables** directory.
c. Copy the **editlivejava** folder into the **/sites/all/libraries/editlive** folder
d. When done properly the main EditLive! JavaScript file would be at **/sites/all/libraries/editlive/editlivejava/editlivejava.js**

   Note: *If you download the update SDK file the* **editlivejava** *folder is directly in the root of the unzipped content.*

f) Refresh the WYSIWYG configuration screen and click on the Installation Instructions link – now EditLive! should show as installed and a version number should appear on the right side of the row

EditLive (Download)   9

This signifies that EditLive! was installed to the correct location and that the WYSIWYG module has located the files.

4. Enabling Local Image Uploads

Both Textbox.io and EditLive! allow you to paste or insert local images into the editor. In order to support this capability you need to configure the editor properly to support this process.

4a. Enabling Image Upload for Textbox.io

The TBIO files that you copied to the **WYSIWYG** module in step 2 of this document already contain the needed configuration to allow local images to be uploaded to the Drupal server. If you want to know where this is stored read on…

a) Navigate to the **/sites/all/modules/wysiwyg/editors/js** directory and open the **textboxio-1.js** file.

b) In this file there should already be a **defaultConfig** variable near the top. In the JavaScript object defined by **defaultConfig** you should see an images section that defines a URL property:

```
images: {
    upload: {
        url: '/ephox_image_upload/ephox_process_image'
    },
    allowLocal: true
},
```

If you see this in the file (and you should) there is nothing to be done to configure Textbox.io for image uploads!

4b. Enabling Image Upload for EditLive!

Unlike Textbox.io, there is work to do to configure EditLive! for local image upload. Let's get started…

a) Navigate to the **/sites/all/libraries/editlive/editlivejava** directory. Copy the EditLive! sample configuration file (named *sample_eljconfig.xml*) and make a new file named **configuration.xml**.

b) Open **configuration.xml** with a text editing tool and locate the **<httpUpload>** element in the file. Uncomment the **<httpUpload>** element by removing the starting

and ending comment tags.

```
<!--
<httpUpload
    base="http://www.yourserver.com/userfiles/"
    href="http://www.yourserver.com/scripts/upload.jsp">
    <httpUploadData name="hello" data="world"/>
</httpUpload>
-->
```

… will become …

```
<httpUpload
    base=""
    href="/ephox_image_upload/ephox_process_image">
</httpUpload>
```

c) Edit the HREF and BASE attributes of the **`<httpUpload>`** element

   a. The BASE attribute should be the empty string
   b. The HREF attribute must point to the Ephox Editors Image Upload Acceptor
      module's **`ephox_process_image`** target. The module defines this as the path:

      `/ephox_image_upload/ephox_process_image`.

      This leads to the following configuration settings:

```
<httpUpload
    base=""
    href="/ephox_image_upload/ephox_process_image">
</httpUpload>
```

5. Create a Text Format that uses the installed editor

The final step in getting Textbox.io or EditLive! to work in Drupal is the creation and configuration of a Text Format. The Text Format defines which editor is to be used while editing content.

a) Navigate to the Configuration/Content Authoring/Text Formats screen

```
http://<server>/admin/config/content/formats
```

b) Click on **Add Format** and enter the following values:
   a. Name – enter **Textbox.io** or **EditLive!**
   b. Roles – select whatever roles are appropriate for your usage
   c. Ignore the Enabled Filters (not supported at present)
   d. Click **Save Configuration**

c) Return to Configuration / Content Authoring / Text Formats

http://<server>/admin/config/content/formats

and confirm your new Text Format is in the list. The following screenshot shows one named **EditLive!**:

NAME	ROLES	WEIGHT	OPERATIONS	
Filtered HTML	anonymous user, authenticated user, ephox marketing	-10	configure	disable
Full HTML	administrator	-9	configure	disable
Plain text	*All roles may use this format*	-8	configure	
EditLive	administrator	0	configure	disable
PHP code	administrator	-10	configure	disable

d) Navigate to the Configuration/Content Authoring/Wysiwyg Profiles page

http://<server>/admin/config/content/wysiwyg

and confirm your text format is in the list

TEXT FORMAT	EDITOR	OPERATIONS	
Filtered HTML	TinyMCE 3.3.9	Edit	Delete
Full HTML	TinyMCE 3.3.9	Edit	Delete
Plain text	No editor		
EditLive	No editor		
PHP code	No editor		

e) Select Textbox.io or EditLive! from the Editor select list and click **Save** button. The list of profiles should now show either Textbox.io or EditLive! is the editor for the Text Format. The following screenshot shows two different Text Formats created – one that uses Textbox.io and one that uses EditLive!. Note that the name of the Text Format can be whatever you like:

TEXT FORMAT	EDITOR
Full HTML	TinyMCE 3.3.9
EditLive HTML	EditLive 9
TBIO	TextBox.io 1

f) Now you can test to see if things are configured properly. Create a new content item and use your newly created Text Format. If everything is configured properly you will see Textbox.io or EditLive! load.

Note: If you have issues at this point please register on our support site at http://support.ephox.com *and open a support ticket.*

Injecting CSS into the Editor

When using a rich text editor you typically want the wysiwyg view of the editor to match what the author will see when the content is published. In order to do this you need to pass a CSS file to the editor. This CSS file will control how the wysiwyg view of Textbox.io and EditLive! renders content.

We recommend you create a specific CSS file for the editor that includes only those classes needed to render and edit content. This will stop the author from seeing a long list of CSS classes that they may never need to leverage.

*Note: Test that Textbox.io or EditLive! loads **BEFORE** you try to add CSS. If the path to the CSS is wrong the editor may not load. If you test the remainder of the work first you will know that any issues at this point are CSS related!*

Configure CSS to be passed into Textbox.io

For details on how EditLive! leverages CSS please see:

http://docs.ephox.com/display/tbio/Using+your+own+styles

a) Navigate to the **/sites/all/modules/wysiwyg/editors/js** directory and open the **textboxio-1.js** file.

b) In this file there should already be a **defaultConfig** variable near the top. In the JavaScript object defined by **defaultConfig** you should see a **css** section that defines a **stylesheets** property:

```
var defaultConfig = {
    basePath: '/sites/all/libraries/textboxio/resources'
    css: {
        stylesheets: [''],
```

c) You can add any number of CSS files that you like to this stylesheets property. For example:

```
stylesheets : ['http://www.example.com/mycss.css', 'anotherfile.css'],
```

You can use relative or complete paths to the CSS files so long as the files are accessible to the user when using their browser. If there is an error loading the CSS file you will likely

see an error message in the browser's JavaScript console.

d) Now you can test to see if things are configured properly. Create a new content item and use your Textbox.io text format. Test to see that the CSS you provided is coming through and that you see the correct fonts/sizes/colors/etc in Textbox.io.

Configure CSS to be passed into EditLive!

For details on how EditLive! leverages CSS please see:

http://docs.ephox.com/display/EditLive/Using+CSS+in+the+Applet.

a) Place your CSS file in your **DEFAULT** theme at
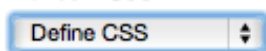   **/sites/all/themes/<themename>/css**

   (we will assume the file is named editor.css for this document)

b) Navigate to the Configuration/Content Authoring/Wysiwyg Profiles page

   http://<server>/admin/config/content/wysiwyg

c) Click the **Edit** link on the EditLive! profile line

d) Click on **CSS** to open the CSS section

e) Select Define CSS in the Editor CSS section

Editor CSS
Define CSS ⬍

f) Enter a path to the CSS file. (Since we placed it in the Default Theme we can use substitution variables to help create the path:

CSS path
%b%t/css/editor.css

g) Click **Save** to save the EditLive! profile.

h) Click **Save** to save the wysiwyg profile page changes.

i) Now you can test to see if things are configured properly. Create a new content item and use your EditLive! text format. Test to see that the CSS you provided is coming through and that you see the correct fonts/sizes/colors/etc in EditLive!.