

I am using CKEdit

or for Joomla with Joomla! 2.5 (I know I need to update this site) and had a hard time getting sourcer to work. The problem was that CKEditor encoded everything including apostrophes.

If I turned off encoding, nothing would be encoded which is a problem for client admin input.

I used the lower 3 options before coming to this conclusion:

### **The Best Method for Using Sourcer:**

Leave the HTML Entities radio button on Yes:

LINK BROWSER	-	-
Absolute URL in link browser	<input checked="" type="radio"/> Disabled	<input type="radio"/> Enabled
SCAYT autostart	<input checked="" type="radio"/> Disabled	<input type="radio"/> Enabled
HTML entities	<input type="radio"/> Disabled	<input checked="" type="radio"/> Enabled

Add the following into the Custom Configuration:

### Custom Configuration

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options that are not available in this Recommended for advanced users only. Be aware that if you make a mistake here, CKEditor may not load correctly.

Sample configuration:

```
CKEDITOR.config.entities = false;  
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
// The following MUST be double backslashed every change.  
CKEDITOR.config.protectedSource.push( /{source}[\s\S]*?(\/source)/g );
```

### Custom Configuration

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options that are not available in this Recommended for advanced users only. Be aware that if you make a mistake here, CKEditor may not load correctly.

Sample configuration:

```
CKEDITOR.config.entities = false;  
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
// The following MUST be double backslashed every change.  
CKEDITOR.config.protectedSource.push( /{source}[\s\S]*?(\/source)/g );
```

**Below are three other options for entering source code into CKEditor for Joomla!:**

**The first option:**

- PRO
- Easy to implement
- Retains the encoding of basic HTML special chars for lay users and convenience
  
- CON
- Can't use OOP (No > char)
- Can only use apostrophes. No Quotes inside code.

To do this leave the option for encoding entities on:

### LINK BROWSER

Absolute URL in link browser

Disabled

Enabled

SCAYT autostart

Disabled

Enabled

HTML entities

Disabled

Enabled

Then in the Custom Configuration space add this:

### Custom Configuration

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options. **Recommended for advanced users only.** Be aware that if you make a mistake here, CKEditor will not work.

Sample configuration:

```
CKEDITOR.config.entities = false;  
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
CKEDITOR.config.entities_additional = '';
```

The default value of config.entities\_additional is the encoded apostrophe, so this removes that

### The Second Option:

- PRO
  - Can use OOP
  - Can use Quotes and apostrophes
- CON
  - No encoding takes place. Text entry does not encode ampersands, quotes, apostrophes, etc.

To do this, turn off the radio button for encoding, and add this to the box below to still encode ampersands and angle brackets. *NOTE: The first line is probably not necessary as the radio button already did that, but I settled for option 1 and did not test that.*

```
CKEDITOR.config.entities = false;  
CKEDITOR.config.basicEntities = true;
```

### Custom Configuration

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options that are not available by default. **Recommended for advanced users only.** Be aware that if you make a mistake here, CKEditor may not load correctly.

Sample configuration:

```
CKEDITOR.config.entities = false;
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
CKEDITOR.config.entities = false;
CKEDITOR.config.basicEntities = true;
```

### The Third Option:

- PRO
  - Best of both worlds. Encoding takes place on entered text, PHP is preserved (No `[[ ]]` syntax), you can use quotes and apostrophes, and you can use OOP
- CON
  - Must be careful making any further adjustments to CKEditor properties
  - Potential security risk. If you screw up the CKEditor settings (see below), your PHP code may show up in the site. Also, if you are not careful, it could expose the public side to entering PHP code through a WYSIWYG editor (though I think the framework would prevent it unless the public side had access to sourcer as well).

To use the third method, turn encoding back on through the radio button and add the following to the Custom Configuration:

### Custom Configuration

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options that are not available by default. **Recommended for advanced users only.** Be aware that if you make a mistake here, CKEditor may not load correctly.

Sample configuration:

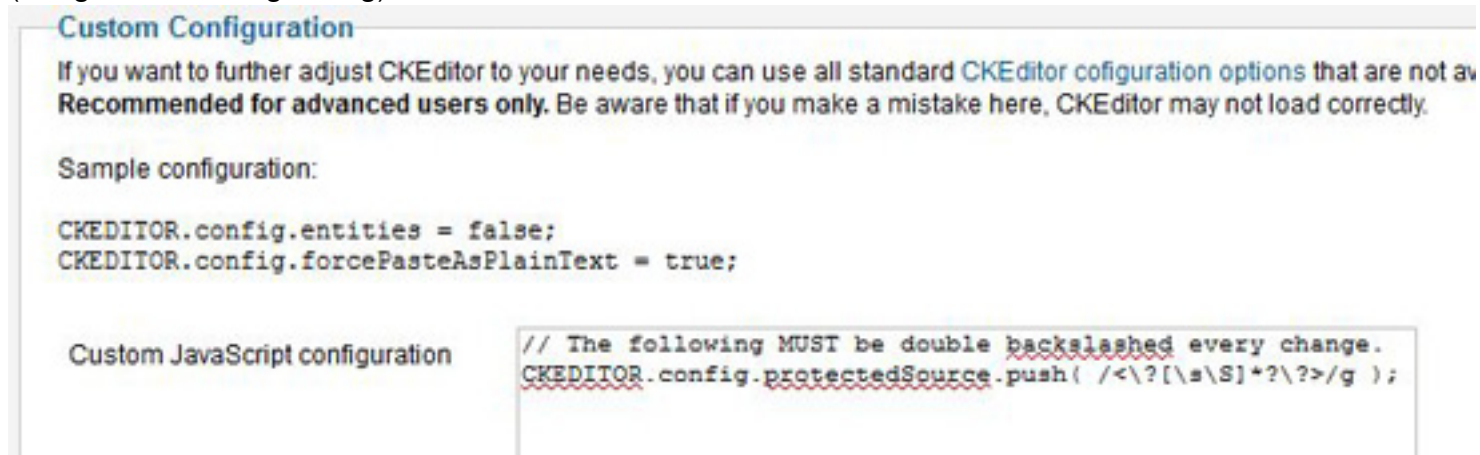
```
CKEDITOR.config.entities = false;
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
// The following MUST be double backslashed every change.
CKEDITOR.config.protectedSource.push( /<\\?[\\s\\S]+?\\>/g );
```

Here is the twist. Notice every backslash is double backslashed. Every time you save the configuration, it strips one backslash, SO ANY FUTURE CHANGES REQUIRE YOU TO REPEAT THE DOUBLE BACKSLASHING OR THE WYSIWYG editor will break.

(Image after saving config)



**Custom Configuration**

If you want to further adjust CKEditor to your needs, you can use all standard CKEditor configuration options that are not a **Recommended for advanced users only**. Be aware that if you make a mistake here, CKEditor may not load correctly.

Sample configuration:

```
CKEDITOR.config.entities = false;
CKEDITOR.config.forcePasteAsPlainText = true;
```

Custom JavaScript configuration

```
// The following MUST be double backslashed every change.
CKEDITOR.config.protectedSource.push( /<\?[\s\S]*?\?>/g );
```

This code protects PHP so you can have your cake and eat it to as long as you don't break the cake.