

## XoopsEditors Compatibility Issues – Fixed - Hacks

NEWS\_PDF\_AUTHOR: Catzwolf

NEWS\_PDF\_DATE: 2008/4/7 13:40:00

For the longest time, I have been trying to understand the workings of the XoopsEditors. I am not a beginner with PHP or the workings of Xoops, but for the longest time, this issue has bugged me really bad, to the point I wanted to pull my hair out (what is left of it anyway) and I nearly gave up many times. Persistence and brute ignorance are two of my many flaws, but they have seen me through.

Anyway, back to the XoopsEditors. While in the process of revamping WF-Channel, I was asked many times to include a WYSIWYG Editor and after much thought about it I decided to go down the XoopsEditors path. So doing what most other people do... I started to look for examples on how it works and how I could incorporate it into WF-Channel.

Well, I have to say the results were not what I expected. Nearly every developer had 'their' own way of doing this and there wasn't one single example that used the same method or methods and the more I looked, the more I got so confused with it all. I hate to think what someone with little knowledge of Xoops and PHP must be thinking, I really do dread.

The problem stems from all the different editors that are actually available right at this moment and the fact that no one has bothered to consult each other over what would be the best approach (Myself included) to bring some uniformity and compatibility to the whole issue, well expect for DJ that is and kudos to him for doing so too.

For me, XoopsEditors should become the standard de facto for all developers and if users want a specialist editor, seek the developers who do these instead and users should install these themselves.

Anyway, back to the XoopsEditors issue. Originally, this class was written for Xoops 2.2 and not the 2.016 branch, and this is where lays the issues for all developers. If you look at the Xoops 2.2 package, especially the XoopsEditors Class and compare it with the class that DJ has made public, there are certain differences between them; mainly files included in the 2.2 version that are missing from the XoopsEditors package.

In the sampleform.php example it states the following for adding this class to your own module:

```
// options for the editor
//required configs
$options['name'] = 'required_element';
$options['value'] = empty($_REQUEST['message'])? "" : $_REQUEST['message'];
```

```
//optional configs
$options['rows'] = 25; // default value = 5
$options['cols'] = 60; // default value = 50
$options['width'] = '100%'; // default value = 100%
$options['height'] = '400px'; // default value = 400px

// "textarea": if the selected editor with name of $editor can not be created, the editor "textarea"
will be used
// if no $onFailure is set, then the first available editor will be used
// If dohtml is disabled, set $noHtml to true
$sample_form->addElement(new XoopsFormEditor(_MD_MESSAGEC, $editor,
$editor_configs, $nohtml = false, $onfailure = "textarea"), true);
```

Yet if I do this on the on the 2.018+ branch I get the following error:

Quote:

```
Fatal error: Class 'XoopsFormEditor' not found in blah blah
```

Well this is really to be expected. Because no where in the XoopsForm class there is a 'hook' to include this class at all. Sure I could add the following line:

Require\_once XOOPS\_ROOT\_PATH.'/class/xoops\_editor.php' before calling this line:

Quote:

```
$sample_form->addElement(new XoopsFormEditor(_MD_MESSAGEC, $editor,
$editor_configs, $nohtml = false, $onfailure = "textarea"), true);
```

But in reality we shouldn't need to do that. If the XoopsForm class requires XoopsEditor to work correctly, then we should have the mechanics already within the XoopsForm class without the developer having to worry about additional lines of code to make it work. The XoopsForm class should handle this.

So bear in mind, the XoopsEditor class was originally written for Xoops 2.20 and the Xoops editor will work correctly as above, but, I want it to work on Xoops version 2.18+ without having to resort to having to add loads more code that just break compatibility.

Lets look at an example within the news module, the workaround by Herve wouldn't have been required if the class was written for both 2.00 and 2.20 branches.

```
$x22=false;
$xv=str_replace('XOOPS ','',XOOPS_VERSION);
if(substr($xv,2,1)=='2') {
    $x22=true;
}

switch(strtolower(news_getmoduleoption('form_options'))) {
    case 'koivi':
        if(!$x22) {
            if ( is_readable(XOOPS_ROOT_PATH . '/class/wysiwyg/formwysiwygtextarea.php')) {
                include_once(XOOPS_ROOT_PATH . '/class/wysiwyg/formwysiwygtextarea.php');
                $editor = new XoopsFormWysiwygTextArea($caption, $name, $value,
'100%', '450px', "");
            }
        } else {
            $editor = new XoopsFormEditor($caption, 'koivi', $editor_configs);
        }
        break;
    }
    return $editor;
```

As you can see, First there is a check to which version of Xoops is being used, then there is a check to see if there is another Koivi editor installed, then if both fail we use the default. This is not the best solution for a developer anyway. In reality we shouldn't need to have two different Editor Classes installed. This is where all the confusion comes from, what works with what?

I am quite sure I am not alone when I say this from a developers point of view, but I dread when someone asks me to add a WYSIWYG editor to my modules. Why? Because there isn't no standard within the Xoops frame work to do so. Well until now.

Actually the answer is quite simple and it gives all developers one way of including the many different editors within the XoopsEditor framework on 2.00, 2.20 and now 2.30 thus having compatibility across all versions of Xoops and without having to resorting to creating their own hooks to do so.

Here is how:

First we have to modify the following file:

Quote:

```
XOOPS_ROOT_PATH.'/class/xoopsformloader.php'
```

And add the following lines at the end:

```
include_once XOOPS_ROOT_PATH . '/ class/xoopsform/formeditor.php';  
include_once XOOPS_ROOT_PATH . '/ class/xoopsform/formselecteditor.php';
```

Then copy the above files from the 2.20 version over into the 2.00 branch that you use.

Hopefully DJ will include these within his excellent XoopsEditor class and developers start to use the above method of including Editors within their own modules and bring back some uniformity and lessen the impact of compatibility issues within Xoops.

Hope this helps everyone a little.

For the longest time, I have been trying to understand the workings of the XoopsEditors. I am not a beginner with PHP or the workings of Xoops, but for the longest time, this issue has bugged me really bad, to the point I wanted to pull my hair out (what is left of it anyway) and I nearly gave up many times. Persistence and brute ignorance are two of my many flaws, but they have seen me through.

Anyway, back to the XoopsEditors. While in the process of revamping WF-Channel, I was asked many times to include a WYSIWYG Editor and after much thought about it I decided to go down the XoopsEditors path. So doing what most other people do... I started to look for examples on how it works and how I could incorporate it into WF-Channel.

Well, I have to say the results were not what I expected. Nearly every developer had 'their' own way of doing this and there wasn't one single example that used the same method or methods and the more I looked, the more I got so confused with it all. I hate to think what someone with little knowledge of Xoops and PHP must be thinking, I really do dread.

The problem stems from all the different editors that are actually available right at this moment and the fact that no one has bothered to consult each other over what would be the best approach (Myself included) to bring some uniformity and compatibility to the whole issue, well expect for DJ that is and kudos to him for doing so too.

For me, XoopsEditors should become the standard de facto for all developers and if users want a specialist editor, seek the developers who do these instead and users should install these themselves.

Anyway, back to the XoopsEditors issue. Originally, this class was written for Xoops 2.2 and not the 2.016 branch, and this is where lays the issues for all developers. If you look at the Xoops 2.2 package, especially the XoopsEditors Class and compare it with the class that DJ has made public, there are certain differences between them; mainly files included in the 2.2 version that are missing from the XoopsEditors package.

In the sampleform.php example it states the following for adding this class to your own module:

```
// options for the editor
//required configs
$options['name'] = 'required_element';
$options['value'] = empty($_REQUEST['message'])? "" : $_REQUEST['message'];
//optional configs
$options['rows'] = 25; // default value = 5
$options['cols'] = 60; // default value = 50
$options['width'] = '100%'; // default value = 100%
$options['height'] = '400px'; // default value = 400px

// "textarea": if the selected editor with name of $editor can not be created, the editor "textarea"
will be used
```

```
// if no $onFailure is set, then the first available editor will be used
// If dohtml is disabled, set $noHtml to true
$sample_form->addElement(new XoopsFormEditor(_MD_MESSAGEEC, $editor,
$editor_configs, $nohtml = false, $onfailure = "textarea"), true);
```

Yet if I do this on the on the 2.018+ branch I get the following error:

Quote:

```
Fatal error: Class 'XoopsFormEditor' not found in blah blah
```

Well this is really to be expected. Because no where in the XoopsForm class there is a 'hook' to include this class at all. Sure I could add the following line:

Require\_once XOOPS\_ROOT\_PATH.'/class/xoops\_editor.php' before calling this line:

Quote:

```
$sample_form->addElement(new XoopsFormEditor(_MD_MESSAGEEC, $editor,
$editor_configs, $nohtml = false, $onfailure = "textarea"), true);
```

But in reality we shouldn't need to do that. If the XoopsForm class requires XoopsEditor to work correctly, then we should have the mechanics already within the XoopsForm class without the developer having to worry about additional lines of code to make it work. The XoopsForm class should handle this.

So bear in mind, the XoopsEditor class was originally written for Xoops 2.20 and the Xoops editor will work correctly as above, but, I want it to work on Xoops version 2.18+ without having to resort to having to add loads more code that just break compatibility.

Lets look at an example within the news module, the workaround by Herve wouldn't have been required if the class was written for both 2.00 and 2.20 branches.

```
$x22=false;
$xv=str_replace('XOOPS ','',XOOPS_VERSION);
if(substr($xv,2,1)=='2') {
    $x22=true;
}
```

```
switch(strtolower(news_getmoduleoption('form_options')) {
  case 'koivi':
    if (!$x22) {
      if ( is_readable(XOOPS_ROOT_PATH . '/class/wysiwyg/formwysiwygtextarea.php')) {
        include_once(XOOPS_ROOT_PATH . '/class/wysiwyg/formwysiwygtextarea.php');
        $editor = new XoopsFormWysiwygTextArea($caption, $name, $value,
'100%', '450px', "");
      }
    } else {
      $editor = new XoopsFormEditor($caption, 'koivi', $editor_configs);
    }
    break;
}
return $editor;
```

As you can see, First there is a check to which version of Xoops is being used, then there is a check to see if there is another Koivi editor installed, then if both fail we use the default. This is not the best solution for a developer anyway. In reality we shouldn't need to have two different Editor Classes installed. This is where all the confusion comes from, what works with what?

I am quite sure I am not alone when I say this from a developers point of view, but I dread when someone asks me to add a WYSIWYG editor to my modules. Why? Because there isn't no standard within the Xoops frame work to do so. Well until now.

Actually the answer is quite simple and it gives all developers one way of including the many different editors within the XoopsEditor framework on 2.00, 2.20 and now 2.30 thus having compatibility across all versions of Xoops and without having to resorting to creating their own hooks to do so.

Here is how:

First we have to modify the following file:

Quote:

```
XOOPS_ROOT_PATH.'/class/xoopsformloader.php'
```

And add the following lines at the end:

```
include_once XOOPS_ROOT_PATH . '/ class/xoopsform/formeditor.php';
include_once XOOPS_ROOT_PATH . '/ class/xoopsform/formselecteditor.php';
```

Then copy the above files from the 2.20 version over into the 2.00 branch that you use.

Hopefully DJ will include these within his excellent XoopsEditor class and developers start to use the above method of including Editors within their own modules and bring back some uniformity and lessen the impact of compatibility issues within Xoops.

Hope this helps everyone a little.