

Warning: Trying to access array offset on value of type null in
/var/www/vhosts/thecmsbcookbook.com/httpdocs/recipeDetail3.php on line 44

The CMS Builder Cookbook

[Home](#)

[Read Excerpts](#)

[View the
Table of
Contents](#)

[All
User Submitted
CMS Plugins](#)

[CMSB Resources](#)

[Contact Us](#)

DISPLAY LINKS ON YOUR WEB PAGE WITHOUT ERRORS - Nov 23rd, 2012

REVISED TO ACCOMMODATE HTTPS:// WEBSITE URLs

If a link URL is entered into a textfield without an http://, you'll end up with a broken link when you try to follow that link in your viewer.

Here's how to make sure that your URL's work, even if the person entering the information doesn't enter http://

Note: The solution in this recipe is not case sensitive. This means that if you've entered HTTP:// , or Http:// the code will still work.

In this example the code is designed for use with a single field. The field containing the URL is called web_site_link_url, and the field containing the link text is called web_site_link_text.

Once you understand the concepts involved, to use this for multiple link fields in the same viewer, see USING A FUNCTION TO ACCOMPLISH THE SAME RESULT, later in this recipe.

An "if" statement is used to check if the URL starts with http:// and if not, it adds one to the beginning of the URL.

So, for a multi-record list viewer, you'd put the if statement before the link code in the viewer like this:

```
<?php if(!preg_match("/^https:\V/i", $record['web_site_link_url'])):?>
<?PHP if (!preg_match("/^http:\V/i", $record['web_site_link_url'])) {
    $record['web_site_link_url'] = "http://" . $record['web_site_link_url']; } ?>

<?PHP endif ?>

<a href="<?PHP echo $record['web_site_link_url'] ?>"><?PHP echo $record['web_site_link_text'] ?></a>
```

And for a detail viewer:

```
<?php if(!preg_match("/^https:\V/i", $yourRecord['web_site_link_url'])):?>
<?PHP if (!preg_match("/^http:\V/i", $yourRecord['web_site_link_url'])) {
    $yourRecord['web_site_link_url'] = "http://" . $yourRecord['web_site_link_url']; } ?>
<?PHP endif ?>
<a href="<?php echo $yourRecord['web_site_link_url'] ?>"><?php echo $yourRecord['web_site_link_text'] ?></a>
```

Or if you're using one of the info fields in an upload to contain the URL:

```
<?php if(!preg_match("/^https:\V/i", $upload['info1'] )):?>
<?php if (!preg_match("/^http:\V/i", $upload['info1'])) {
$upload['info1'] = "http://" . $upload['info1']; }
?><?php endif ?>
```

```
<a href="<?php echo $upload['info1'] ?>">" height="<?php echo $upload['thumbHeight'] ?>" alt="" /></a>
```

Or for a single record viewer:

```
<?php if(!preg_match("/^https:\V/i", $your_tableRecord['web_site_link_url'] )):?>
<?PHP if (!preg_match("/^http:\V/i", $your_tableRecord['web_site_link_url'])) {
$your_tableRecord['web_site_link_url'] = "http://" . $your_tableRecord['web_site_link_url']; } ?>
<?php endif ?>
<a href="<?php echo $your_tableRecord['web_site_link_url'] ?>"><?php echo $your_tableRecord['web_site_link_text'] ?>
</a>
```

PLACEMENT ON THE PAGE

As long as the preg_match PHP code is before the link code, this technique will work. So, for consistency, you might want to insert all your preg_match PHP code blocks at the top of your <body> code.

HOW IT WORKS

The first line is an if statement that surrounds the rest of the regex operations

```
<?php if(!preg_match("/^https:\V/i", $record['web_site_link_url'] )):?>
```

This statement looks for https:// at the beginning of a field called "web_site_link_url".

If https:// is not found then the next line is executed.

(The "i" make the expression case insensitive. If you want to make it case sensitive, just remove the "i")

```
!preg_match("/^http:\V/i", $record['web_site']
```

This code looks for http:// at the beginning of a field called "web_site_link_url".

(Again, the "i" make the expression case insensitive. If you want to make it case sensitive, just remove the "i")

If http:// is not found (! = NOT) then the next line adds it to the beginning of 'web_site_link_url':

```
$record['web_site_link_url'] = "http://" . $record['web_site_link_url']
```

This line uses the concatenation operator, which is just a dot (.) The above expression can be thought of like $2 = 1 + 1$ or in our case maybe `http://www.cnn.com = (http://) + (www.cnn.com)`

The end if statement <?PHP endif ?> ends the conditions that must be met.

The last line would go where you want the link to appear, as long as it's below the if statement.

```
<a href="<?PHP echo $record['web_site_link_url'] ?>"><?PHP echo $record['web_site_link_text'] ?></a>
```

This displays the \$record[web_site_link_ur] as the link and the \$record['web_site_link_text'] as the link text.

USING A FUNCTION TO ACCOMPLISH THE SAME RESULT

Instead of having to duplicate the code and change the field name, you can use a function that can easily be called for each occurrence.

Just remember to insert the function before and outside of any foreach loops that require it.

```
<?php function ensure_url($url)
{ if(!preg_match("/^https:\V/i", $url ))
{
if (!preg_match("/^http:\V/i", $url)) {
$url = "http://" . $url;
}}
return $url;
}
?>
```

Then, each time you need to check the format of a link, call the function like this:

```
<?php echo ensure_url($record['your_link_field']) ?>
```

or

```
<?php echo ensure_url($your_tableRecord['your_link_field']) ?>
```

Here's a Regex quick reference to help you to customize your regular expressions to meet your needs

```
[abc]    A single character: a, b or c
[^abc]   Any single character but a, b, or c
[a-z]    Any single character in the range a-z
[a-zA-Z] Any single character in the range a-z or A-Z
^        Start of line
$        End of line
\A       Start of string
\z       End of string
.        Any single character
\s       Any whitespace character
\S       Any non-whitespace character
\d       Any digit
\D       Any non-digit
\w       Any word character (letter, number, underscore)
\W       Any non-word character
\b       Any word boundary character
(...)    Capture everything enclosed
(a|b)    a or b
a?       Zero or one of a
a*       Zero or more of a
a+       One or more of a
a{3}     Exactly 3 of a
a{3,}    3 or more of a
a{3,6}   Between 3 and 6 of a
```

options: i case insensitive m make dot match newlines x ignore whitespace in regex o perform #{...} substitutions only once

The materials on this web site have been created for use with CMS Builder content management software. CMS Builder software is published and licensed for use by InteractiveTools.com. Please contact [Interactive Tools](#) for information on the downloading of the software or the purchasing of licenses.

Terms of Service

