

The CMS Builder Cookbook

[Home](#)

[Read Excerpts](#)

[View the Table of Contents](#)

[All User Submitted CMS Plugins](#)

[CMSB Resources](#)

[Contact Us](#)

THE COMPLETE CMSB COOKBOOK

[CLICK TO VIEW THE RECIPES AS A "TABLE OF CONTENTS"](#)

We now have 559 recipes!

Section 1 (Recipes 001 - 100) - [Download PDF](#)
Section 2 (Recipes 101 - 200) - [Download PDF](#)
Section 3 (Recipes 201 - 300) - [Download PDF](#)
Section 4 (Recipes 301 - 400) - [Download PDF](#)
Section 5 (Recipes 401 - 500) - [Download PDF](#)
Section 6 (Recipes 501 - 559) - [Download PDF](#)

LAST UPDATE - Aug 8th, 2023

Click  To View The Recipe

[Click Here To Skip Down To The Complete Cookbook](#)

CHOOSING WHICH RECORDS TO DISPLAY USING OFFSET AND LIMIT COMMANDS - Sep 26th, 2020

Nigel Gordijk from Common Sense Design needed to be able to differentiate between groups of records so that some of their associated images could be used in an image carousel and others could be presented in a list format.

It seemed complicated until CMSB user KennyH offered this simple suggestion using a combination of the limit and offset command in his list records calls.

Here' his elegant solution:

Here we load the first 5 articles

```
list($article_postsRecords, $article_postsMetaData) = getRecords(array(
```

```
'tableName' => 'article_posts',
'limit' => '5',
'loadUploads' => true,
'allowSearch' => false,
));
```

Now, we can create add this again for the next five records, but with a few modifications:

```
list($article_postsSideListRecords, $article_postsSideListMetaData) = getRecords(array(
'tableName' => 'article_posts',
'limit' => '5',
'offset' => '4',
'loadUploads' => true,
'allowSearch' => false,
));
```

\$article_postsRecords, \$article_postsMetaData becomes \$article_postsSideListRecords, \$article_postsSideListMetaData

Add 'offset' => '4', to skip the first 5 records (numbering starts at zero)

Then you would display the first 5 images on the page like this

```
<?php foreach ($article_postsRecords as $record): ?>

<?php foreach ($record['images'] as $index => $upload): ?>
" height="<?php echo
$upload['height'] ?>" alt="">
<?php endforeach ?>

<?php endforeach ?>
```

The next 5 images pull from \$article_postsSideListRecords

```
<?php foreach ($article_postsSideListRecords as $record): ?>

<?php foreach ($record['images'] as $index => $upload): ?>
" height="<?php echo
$upload['height'] ?>" alt="">
<?php endforeach ?>

<?php endforeach ?>
```

USING GROUPBY TO CREATE A LIST OF RECORDS THAT HAVE THE SAME VALUES AND ONLY RETURN ONE RECORD PER GROUP - May 28th, 2019

I had created a service provider directory for a client who wanted to create a list of the countries and states/provinces that had provider listing records (accounts). They also wanted to show a count of how many listings there were for each country and each state/province, eliminating any hidden listings.

With a great deal of help from Daniel Loewe and Greg Thomas, here's what we came up with.

You can see the result at <https://dbtproviders.com/maplist.php>

Daniel first suggested using the `groupBy` function to create a list of records that have the same values and only return one record per group.

```
list($accountsRecords, $accountsMetaData) = getRecords(array(
'tableName' => 'accounts',
'orderBy' => 'practice_country ASC , practice_state ASC ',
'groupBy' => 'practice_country, practice_state',
));
```

Then to display the list, he suggested the following (Greg added the code for counting):

```
<?php $old_group = ""; // init blank var.

foreach ($accountsRecords as $record): ?>
<?php if (!$record['practice_country:label'] == " "):?>
<?php $group = $record['practice_country:label']; // load sub-
```

```

group value from record. ?>
    <?php else : ?>
    <?php $group = $record['other_practice_country']; ?>
    <?php endif ?>
    <?
php $stateCount = mysql_count('accounts', mysql_escapef("`practice_state` = ? AND `hidden` = '0'
", $record['practice_state'])); ?>
    <?
php $countryCount = mysql_count('accounts', mysql_escapef("`practice_country` = ? AND `hidden` = '0'
", $record['practice_country'])); ?>
    <div class="rTableRow">
    <div class="rTableCell text_font">
    <?PHP
if ($group != $old_group) {echo "<h3>$group ($countryCount Listings)</h3>";} ?>
    <?php $state = ($record['practice_state']); ?>
    <?php $country = ($record['practice_country:label']); ?>
    <?php $recnum = ($record['num']); ?>
    <a href="maps.php?state=<?php echo $state ?>&country=<?
php echo $country ?>"><?php echo ucwords($state);
?> (<?php echo $stateCount ?>) </a></div>
    </div>
    <?PHP $old_group = $group; // retain sub-
group name before moving to new record. ?>
    <?php endforeach; ?>

```

THE CODE FOR A 2 COLUMN LISTING PAGE WITH A LINKED IMAGE - Aug 2nd, 2010

Here's an example of a 2 column list page that lists events in rows with a list page image in one column and a short description of the event and the date(s) for the event in another. Note that "if" statements are used to eliminate extra spaces and special text if there is no information in particular fields. The "list_page_image" is a field that holds a specific image that's uploaded for display on the list page.

```

<table width="90%" border="0" cellpadding="5"><?
php foreach ($happeningsRecords as $record): ?>
    <tr>
    <td align="center">&nbsp;  </td>
    <td align="left" valign="top"><hr align="left" width="300" color="#99945e"/></td>
    </tr>
    <tr>
    <td align="center">
    <?php foreach ($record['list_page_image'] as $upload): ?><a href="<?
php echo $record['_link'] ?>">" height="<?
php echo $upload['thumbHeight'] ?>"
alt="" /></a> <div class="body-text-bold">
    <?php if ($upload['info1']): ?><?php echo $upload['info1']; ?><?php endif; ?><br />
    <?php if ($upload['info2']): ?><?php echo $upload['info2']; ?> <?php endif; ?></div>
    <?php endforeach; ?>
    </td><td align="left" valign="top"><a href="<?php echo $record['_link'] ?>">
    <span class="heading-text-bold"><?php echo
$record['title'] ?></span> <?php if ($record['short_description']): ?><br />
    <span class="body-text-bold"><?php echo
$record['short_description'] ?></span><?php endif ?><br />
    <span class="body-text"><?
php echo date("D, M jS, Y g:i", strtotime($record['starting_date'])) ?>
    <?php if ($record['ending_date']): ?>
through <?php echo $record['ending_date'] ?></span>
    <?php endif; ?>
    <br />
    </a><?php endforeach; ?> </td>
    </tr>
</table>

```

chassa2556 was concerned that links on a listings page that were filtered by category were always going to the first record detail page. Here's what he said:

I have my 'see more' link going to:

```
<a href="involvedDetail.php?<?PHP echo $involvedRecord['num'] ?>"> see more </a>
```

I have set it up so it all goes to one section with a category filter. The trouble is it keeps going to the first record. This is even though I am on the second or third record and am expecting it to go to the relating detail page.

This was his header code:

```
list($involvedRecords, $involvedDetails) = getRecords(array(
    'Table name' => 'involved',
    'where' => ' category = "Join Us" ',
));
```

Mr. Jim from WebCamp One was able to shed some light on the problem. He said:

I'm assuming that you have two pages: a list page and then a detail page. The detail page would be the one that pops up when you click on 'read more'. I'm also assuming that you are restricting anything listed on your list page to items that fall within the category "Join Us".

If I'm correct, try this:

1. In your list page, add the following line to your header code:

```
'where' => 'category = "Join Us"',
```

After you add that code, that 'where' statement should be the only 'where' statement in the list page's header code.

2. Remove that same line from the header on your existing detail page and replace it with the following lines:

```
'where' => whereRecordNumberInUrl(1),
'limit' => '1',
```

3. Finally, change the code in your list page to read:

```
<a href="involvedDetail.php?<?PHP echo $involvedRecord['_link'] ?
>"> see more </a>
```

That seemed to do the trick for chassa2556.

ADD A HORIZONTAL LINE TO SEPARATE LETTER GROUPS IN A LIST - Aug 2nd, 2010

Rconring wanted to know how to add an <hr> between the letter groups in an alphabetized list of links. Ross Fairbairn - Product Specialist at Interactive tools had the solution. He said, Just add the indicated code to your page and you're good to go.

```
<?php foreach ($linksRecords as $record): ?>
  <a target="_blank" title="<?php echo $record['screen_tip'] ?>"
  href="<?php echo $record['web_address'] ?>"><?php echo $record['title'] ?></a>
<br />
<!-- Insert This Code -->
<?php
  $letter = substr($record['title'], 0,1);
  if ($letter != @$lastLetter) { print '<hr width="50%" size="1">'; }
  $lastLetter = $letter;
?>
<!-- End of Insert-->
<?php endforeach ?>
```

GROUPING SUBTOPICS IN YOUR LIST VIEW - Aug 2nd, 2010

InHouse offered this simple way of grouping the contents of List Views. I thought it might be useful for someone. Share and enjoy.

In the Editor, create a field to hold the group names. Use the "List" option with either a pull down or check box format to limit user mistakes and make this field mandatory as well. In this example the field is called "type". There are also fields called "title" and "content".

```
<?PHP
$old_group = ""; // init blank var.
foreach ($yourtableRecord as $record):
$group = $record['type']; // load sub-group value from record.
if ($group != $old_group) { // If different from the last sub-group value, print the sub-
group name.
echo "<h5>$group</h5>";
}?>

<a href="<?PHP echo $record['_link'] ?>"><?PHP echo $record['title'] ?></a>
<?PHP echo $record['content'] ?>

<?PHP $old_group = $group; // retain sub-
group name before moving to new record. ?>
<?PHP endforeach ?>
```

If you want to only show a single sub group then use an IF statement within the FOREACH loop:

```
<?PHP
$desired_group = 'yourgroupname'; // init target var.
foreach ($yourtableRecord as $record):

if ($record['type'] == $desired_group ) { // Examine the current Type and see if it matches the desired group type.
?>

<a href="<?PHP echo $record['_link']; ?>"><?PHP echo $record['title']; ?></a>
<?PHP echo $record['content'] ?>

} // End If ?>
<?PHP endforeach ?>
```

Thanks InHouse!!!

Here's how I used this idea on the CMSBuilder Cookbook Links page with a table called "resources" and fields called "category", url, and link_text". The line

```

<table width="800" border="0" cellpadding="0">

<?php $old_group = ""; // init blank var.
foreach ($resourcesRecords as $record):
$group = $record['category']; // load sub-group value from record. ?>

<tr>
<td align="left" class="Medium-Text">
<?PHP if ($group != $old_group) {echo "<h2>$group</h2>";};?>

<a href="<?php echo $record['url'] ?>" target="_blank"><?
php echo $record['link_text'] ?></a>
<br /><?php echo $record['content'] ?><hr align="center" color="#999999" /></td>
</tr>

<?PHP $old_group = $group; // retain sub-
group name before moving to new record. ?>
<?php endforeach; ?>

</table>

```

CREATING A LIST PAGE WITH NESTED INFORMATION FROM 2 MULTI RECORD EDITORS - Aug 2nd, 2010

The goal is to create a list page using information from 2 multi record editors. "Exhibitions", and "Opening Receptions".

The "Exhibitions" editor has all of the permanent event information and the "Opening Receptions" contains only the opening reception information for each exhibition. (I'm using 2 editors so that the "Opening Reception" information can automatically be hidden after the reception date has passed)

Here's the code I thought would work, but it didn't. Then you'll see what did work, and understand why:

Since I was going to access 2 section editors, I put both my getRecords() calls in the head of the document:

```

<?php
require_once "/yourpathto/cmsAdmin/lib/viewer_functions.php";

list($exhibitionsRecords, $exhibitions metadata) = getRecords(array(
'Table name' => 'exhibitions',
));

list($opening_receptionsRecords, $opening_receptionsMetaData) = getRecords(array(
'Table name' => 'opening_receptions',
'where' => "exhibition LIKE '%" .mysql_real_escape_string(@$_REQUEST['sort_exhibition']). "%'",
));

?>

```

And then in the body:

```

<table>
<?php foreach ($exhibitionsRecords as $record): ?>
<tr>
<td><a href="<?php echo $record['_link'] ?>"><?php echo $record['exhibition'] ?>"
</a>

Show Dates: <?php echo date("D, M jS", strtotime($record['opening_date'])) ?> - <?
php echo date("D, M jS, Y ",
strtotime($record['closing_date'])) ?>
<hr color="#C70000" width="300" />

Opening Reception: <?php foreach ($opening_receptionsRecords as $record): ?><?
php echo date("D, M jS",
strtotime($record['reception_date'])) ?><?php endforeach; ?>
</td>
</tr>
<?php endforeach; ?>
</table>

```

Using this approach, no matter how I adjusted the code, all of the "Opening Reception" dates were showing in each

Using this approach, no matter how I adjusted the code, all of the "Opening Reception" dates were showing in each event listing.

I was out of ideas, when Dave suggested that my initial approach didn't work because you can't look up all the opening dates without knowing all the exhibition titles in advance. So one way to do it is to load the opening date as you display each exhibition.

Instead of putting both `getRecords()` calls in the head of the document, set up one `getRecords()` in the document head that loads the exhibitions records. Then create the `foreach` loop to display them.

Then, inside that `foreach` loop, insert the other `getRecords()` call that loads the opening dates for each particular exhibition and a `foreach` loop displays those. Like this:

In the head:

```
<?php
require_once "/yourpath/to/cmsAdmin/lib/viewer_functions.php";

list($exhibitionsRecords, $exhibitionsMetadata) = getRecords(array(
    'Table name' => 'exhibitions',
));

?>
```

Then in the Body:

```
<table>
<?php foreach ($exhibitionsRecords as $record): ?>
<tr>
<td>
<a href="<?php echo $record['_link'] ?>" <class="style2">"<?
php echo $record['exhibition'] ?>"</a>
<br />
Show Dates: <?php echo date("D, M jS", strtotime($record['opening_date'])) ?> - <?
php echo date("D, M jS, Y ",
strtotime($record['closing_date'])) ?><br />
<?php echo $record['days_and_times'] ?></p><hr color="#C70000" width="300" />

Opening Reception: <?php
list($opening_receptionsRecords, $opening_receptionsMetadata) = getRecords(array(
    'Table name' => 'opening_receptions',
    'where' => "exhibition = '" .mysql_real_escape_string(@$record['exhibition']). "'",
));
?>

<?php foreach ($opening_receptionsRecords as $reception): ?>
<?php echo date("D, M jS", strtotime($reception['reception_date'])) ?>
<?php endforeach; ?>

</td>
</tr>
<?php endforeach; ?>
</table>
```

Dave's approach worked like a charm, however, he pointed out that this adds MySQL queries, so if you're displaying lots and lots of records this might not be a very efficient approach.

USING DIFFERENT STYLES TO DISPLAY GROUPS OF RECORDS ON A LIST PAGE - Aug 2nd, 2010

NigelGordijk had a number of articles that he wanted to display on his list page in the following format:

- The latest article with a large photo and a intro description
- The next two articles with smaller photos and intro descriptions
- The next five articles as links only, without photos or intro descriptions

Damon from Interactive Tools offered this solution.

He said, "The code consists of three IF statements:

The first: Top Story - if the counter is 1 then output top story code.

The second: Next Two Stories - if the stories are number 2 or number 3, use the code to output the small images etc.

The third: Last Five Articles - if the story number is greater than 3, use the code that outputs only the title (with link) and date, no images."

You can see how this works by setting up a multi-record editor called "News" with at least a "Title" field, a "Content" field, and one Upload field called "Uploads" and then adding some test records.

Here are the settings Damon used for the thumbnails for this example:

Thumb 1 - 235px wide x 999px high
Thumb 2 - 480px wide x 999px high

Now create a list viewer with the following code before the <head>

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/your_path_to/lib/viewer_functions.php";

list($newsRecords, $newsMetaData) = getRecords(array(
    'tableName' => 'news',
));

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

And in the body of your list viewer insert the following code:

```
<?php $counter = 1; ?>
<?php foreach ($newsRecords as $record): ?>

<?php if(($counter == 1) || ($newsMetaData['page'] > 1)) : ?>
<!-- TOP STORY -->
<?php foreach ($record['uploads'] as $upload): ?>
<?php if ($upload['hasThumbnail']): ?>
" height="<?php echo $upload['thumbHeight2'] ?>" alt="" /><br />
<?php endif ?>
<strong><a href="<?php echo $record['_link'] ?>"><?php echo $record['title'] ?></a>
</strong><br />
<?php echo date("D, M jS, Y", strtotime($record['date'])) ?><br />
<?php echo $record['content'] ?><br />
<?php endforeach ?>
<hr />
<!-- END TOP STORY -->

<?php elseif(($counter > 1) && ($counter < 4)) : ?>
<!-- NEXT TWO STORIES -->
<a href="<?php echo $record['_link'] ?>"><?php echo $record['title'] ?></a><br />
<?php foreach ($record['uploads'] as $upload): ?>
<?php if ($upload['hasThumbnail']): ?>
" height="<?php echo $upload['thumbHeight'] ?>" alt="" align="left" /><br />
<?php endif ?>
<?php endforeach ?>
Date: <?php echo date("D, M jS, Y", strtotime($record['date'])) ?><br />
<?php echo $record['content'] ?>
<br clear="left" />
<hr />
<br clear="left" />
<!-- END NEXT TWO STORIES -->

<?php elseif($counter > 3): ?>
<!-- REMAINING ARTICLES -->
<a href="<?php echo $record['_link'] ?>"><?php echo $record['title'] ?></a><br />
Date: <?php echo date("D, M jS, Y", strtotime($record['date'])) ?><br />
<!-- END REMAINING ARTICLES -->

<?php endif; ?>
<?php $counter++; ?>
<?php endforeach; ?>
```

If you want to limit the number of words displayed for the “Content” and link to a detail page, you can place the following code before the code in the <body> of your page to define a “maxWords function:

```
<?PHP
function maxWords($textOrHtml, $maxWords) {
    $text = strip_tags($textOrHtml);
    $words = preg_split("/\s+/", $text, $maxWords+1);
    if (count($words) > $maxWords) { unset($words[$maxWords]); }
    $output = join(' ', $words);

    return $output;
}
?>
```

And replace the

```
<?php echo $record['content'] ?>
```

code, with something like this to utilize the “maxWords” function you inserted:

```
<?PHP echo maxWords($record['content'], 25);
?>...<a href="<?php echo $record['_link']; ?>">Read More</a>
```

The 25 can be any number and will define the number of words you want to display. Dragging the record to new positions on the record list will change the way they are displayed on your list page. The record at the top of the list is #1, and the second is # 2, etc. Note: the <br clear="left"/> makes sure that no floating elements are allowed on the left side of the text.

PASSING VARIABLES AT THE END OF A URL FROM A DETAIL PAGE TO A LISTING PAGE - Mar 12th, 2017

Creating a link on a detail page that would show all off the records created by the author of that record on a new list page.

ON THE DETAIL PAGE

First, create a variable from the createdByUserNum value on the detail page

```
<?php $member1 = $my_tableRecord['createdByUserNum'] ?>
```

Then append that variable's value, and the new variable to be passed along to the end of the Link URL

```
<a href="show-all-records-test1.php?member=<?php echo $member1 ?>">Show All records</a>
```

ON THE LISTING PAGE

Add this code before the list records call to define the value of the passed \$member variable

```
<?php $member = $_REQUEST['member']; ?>
```

Then in the list records call, add a 'where' to filter the results to only those records that were created by that author:

```
<?php // load records from 'my_table'  
  
list($my_tableRecords, $my_tableMetaData) = getRecords(array(  
    'tableName' => 'my_table',  
    'loadUploads' => true,  
    'allowSearch' => true,  
    'orderBy' => 'createdDate DESC',  
    'where' => 'createdByUserNum = "'. $member ."' ,  
));  
?>
```

The 'orderBy' => 'createdDate DESC', sorts the results with the newest at the top

UPDATING A TABLE VALUE WITHOUT RELOADING A WEB PAGE USING AN AJAXURL - Dec 12th, 2018

I needed to accept or reject submitted entries into an art exhibition so that exhibition managers could see which entries were going to be in the show and which were not. I also needed to send acceptance/rejection emails to the entrants automatically.

The problem was that there were a great many submissions and I didn't want to re-load the page each time, the way it seemed to work when submitting a form to update the entrant's record. This would have made it extremely hard for a manager to keep track of which submission they were working on.

One more piece of background and then on to the code.

I'm using info5 to indicate which submission images are accepted/rejected. Since I'm also using info5 for price values, there's some code that strips out any existing alphabetical characters from the existing info5 value, and appends either 'keep' or 'remove' to the remaining numbers in the field.

To follow this recipe you'll need a multi-record editor (exhibition_submissions) with at least one multi-image upload field (submission_images) with all 5 info fields.

Your application will probably be different but the concepts should translate easily.

First, on the page that displays all of the submitted images there are 2 small Ajax scripts that pass 2 variables 'recNum' and 'newValue'. One to mark a submission as 'keep' and the other to mark it as 'remove':

```
<script type = "text/javascript">  
    function keepSubmission(recNum, newValue ){  
        ajaxUrl = "keep2017.php?  
submit=1&recNum=" + escape(recNum)+"&newValue=" + escape(newValue);  
  
        $.ajax({  
            url: ajaxUrl,  
        }).done(function() {  
            //add code here if anything needs to happen after the ajax call  
            alert("Submission accepted");  
        });  
    }  
</script>  
<script type = "text/javascript">  
    function removeSubmission(recNum, newValueR){
```

```

    ajaxUrl = "remove2017.php?
submit=1&recNum=" + escape(recNum)+"&newValueR=" + escape(newValueR);

$.ajax({
    url: ajaxUrl,
}).done(function() {
    //add code here if anything needs to happen after the ajax call
    alert("Submission removed");
});
}
</script>

```

Then, in the foreach loop that displays each submitted image, the code to remove any existing text in the info5 field:

```

<?php foreach($exhibition_submissionsRecords as $record) : ?>
<?php foreach ($record['submission_images'] as $upload): ?>
<?php // keep the price but replace any text with 'keep' ?>
    <?php $keep = 'keep' ?>
    <?php $oldValue = $upload['info5'] ?>
    <?php $newValue = preg_replace('/[^\0-9 .]"/, $oldValue) ?>
    <?php $newValue = ($newValue. $keep) ?>
<?php // keep the price but replace any text with 'remove' ?>
    <?php $remove = 'remove' ?>
    <?php $oldValue = $upload['info5'] ?>
    <?php $newValueR = preg_replace('/[^\0-9 .]"/, $oldValue) ?>
    <?php $newValueR = ($newValueR. $remove) ?>
    <?php $recNum = $upload['num'] ?>

<a href="<?php echo $upload['urlPath'] ?>">" height="<?php echo ($upload['thumbHeight2']/2) ?
">" alt="" /></a>

        <div id='ajaxlink' onclick="keepSubmission(<?
php echo "$recNum', '$newValue'" ?>)">CLICK TO ACCEPT THIS
SUBMISSION</div>
        <div id='ajaxlink' onclick="removeSubmission(<?
php echo "$recNum', '$newValueR'" ?>)">CLICK TO REMOVE
THIS SUBMISSION</div>

    <?php if($upload['info1']):?>Title: <?php echo $upload['info1']?><?
php else :?><span
style="color:#C30">No Title</span><?php endif ?>
    <?php if($upload['info2']):?>Medium: <?php echo $upload['info2']?><?
php else :?><span
style="color:#C30">No Medium</span><?php endif ?>
    <?php if($upload['info3']):?>Dimensions: <?php echo $upload['info3']?><?
php else :?><span
style="color:#C30">No Dimensions</span><?php endif ?>
    <?php if($upload['info4']):?>Price: <?php echo $upload['info4']?><?
php else :?><span
style="color:#C30">No Price</span><?php endif ?>

<?php endforeach ?>
<?php endforeach ?>

```

You may have noticed that the Ajax scripts have a url attached to each. These pages are where the actual database updating occurs. Here's the code for those.

A page with the update code for 'keep' called keep2017.php:

```

<?php
// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/your_servre_path/', './', '../', './', './', './');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

$newValue = mysql_escape($_REQUEST['newValue']);
mysqlStrictMode(false);
$query = "UPDATE `{TABLE_PREFIX}uploads` SET
info5 = '$newValue'
WHERE num = '".mysql_escape($_REQUEST['recNum'])."'";
mysql_query($query) or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
$userNum = mysql_insert_id();
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

```

```
"http://www.w3.org/1999/xhtml/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta name="robots" content="noindex,nofollow" />
  </head>
</body>
</html>
```

And a page with the update code for 'remove' called remove2017.php:

```
<?php
// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/your_servre_path/','../','../../','../../../');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

$newValueR = mysql_escape($_REQUEST['newValueR']);
mysqlStrictMode(false);
$query = "UPDATE `{$TABLE_PREFIX}uploads` SET
  info5 = '$newValueR'
  WHERE num = '".mysql_escape($_REQUEST['recNum']).'";
mysql_query($query) or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
$userNum = mysql_insert_id();

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta name="robots" content="noindex,nofollow" />
  </head>
</body>
</html>
```

To allow the exhibition manager to see just those images that were accepted I set up keepviewer.php

With the following active code:

In the head:

```
<script type = "text/javascript">
  function removeSubmission(recNum, newValue) {
    ajaxUrl = "remove2017.php?
    submit=1&recNum=" + escape(recNum)+"&newValue=" + escape(newValue);

    $.ajax({
      url: ajaxUrl,
    }).done(function() {
      //add code here if anything needs to happen after the ajax call
      alert("Submission Removed");
    });
  }
</script>
```

And in the body:

```
<?php foreach($exhibition_submissionsRecords as $record) : ?>
  <?php foreach ($record['submission images'] as $upload) : ?>
    <?php if (strpos($upload['info5'], 'eep') : ?>
  <?php // keep the price but replace any text with 'remove' ?>
    <?php $remove = 'remove' ?>
    <?php $oldValue = $upload['info5'] ?>
    <?php $newValueR = preg_replace('/[^0-9 .]"/, "", $oldValue) ?>
    <?php $newValueR = ($newValueR. $remove) ?>
    <?php $recNum = $upload['num'] ?>
```

followed by the code to display the images and info values

I also included ajax scripting on that page to allow for removal of the submission.

```
<div id='ajaxlink' onclick="removeSubmission(<?php echo "'$recNum', '$newValue'" ?
>)">CLICK TO REMOVE SUBMISSION</div>
```

A few small changes will convert that to a remove viewer with scripting to allow the submission to be re-accepted

Just for completeness, here's the code for remove.php with only one variable 'recNum' (I was not using info5 for anything so I could just change it from '0' to '1' and back again.)

```
<?php mysqlStrictMode(false);
$query = "UPDATE `{$TABLE_PREFIX}uploads` SET
    info5 = '1'
    WHERE num = '".mysql_escape( $_REQUEST['recNum'] )."'";
mysql_query($query) or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
$userNum = mysql_insert_id();

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta name="robots" content="noindex,nofollow" />
</head>
</body>
</html>
```

And the changed code in the ajax script:

```
<script type = "text/javascript">
    function removeSubmission( recordNum ) {
        ajaxUrl = "remove.php?submit=1&recNum=" + escape(recordNum);

        $.ajax({
            url: ajaxUrl,
        }).done(function() {
            //add code here if anything needs to happen after the ajax call
            alert("Submission Removed");
        });
    }
</script>
```

Finally, to allow displaying the alert on the page, add the following to the head:

```
<script type="text/javascript">
<!--
function MM_nbGroup(event, grpName) { //v6.0
var i,img,nbArr,args=MM_nbGroup.arguments;
if (event == "init" && args.length > 2) {
    if ((img = MM_findObj(args[2])) != null && !img.MM_init) {
        img.MM_init = true; img.MM_up = args[3]; img.MM_dn = img.src;
        if ((nbArr = document[grpName]) == null) nbArr = document[grpName] = new Array();
        nbArr[nbArr.length] = img;
        for (i=4; i < args.length-1; i+=2) if ((img = MM_findObj(args[i])) != null) {
            if (!img.MM_up) img.MM_up = img.src;
            img.src = img.MM_dn = args[i+1];
            nbArr[nbArr.length] = img;
        }
    }
} else if (event == "over") {
    document.MM_nbOver = nbArr = new Array();
    for (i=1; i < args.length-1; i+=3) if ((img = MM_findObj(args[i])) != null) {
        if (!img.MM_up) img.MM_up = img.src;
        img.src = (img.MM_dn && args[i+2]) ? args[i+2] : ((args[i+1]) ? args[i+1] : img.MM_up);
        nbArr[nbArr.length] = img;
    }
} else if (event == "out") {
    for (i=0; i < document.MM_nbOver.length; i++) {
        img = document.MM_nbOver[i]; img.src = (img.MM_dn) ? img.MM_dn : img.MM_up;
    }
} else if (event == "down") {
    nbArr = document[grpName];
    if (nbArr)
        for (i=0; i < nbArr.length; i++) { img=nbArr[i]; img.src = img.MM_up; img.MM_dn = 0; }
    document[grpName] = nbArr = new Array();
    for (i=2; i < args.length-1; i+=2) if ((img = MM_findObj(args[i])) != null) {
        if (!img.MM_up) img.MM_up = img.src;
        img.src = img.MM_dn = (args[i+1]) ? args[i+1] : img.MM_up;
        nbArr[nbArr.length] = img;
    }
}
}
//-->
</script>
```

PRE-CHECKING ALL ENTRIES MARKED AS 'DEFAULT' IN A MULTI VALUE CHECK BOX LIST - Dec 11th, 2015

With a lot of help from Ross Fairbairn, Interactive Tools' consulting Guru, we came up with this method, based on the following assumptions.

1. You have a multi-record section with one text field for each check box entry. For this example I've called the section "entries". It has at least one text field. For this example the text field is called "list_entry".

2. You have a another multi-record section with a multi-value check box list field. For this example I've called the section "listings" and the list field "my_list".

3) the "my_list" field gets it's options and values from the "entries" section using the Get Options From Database (advanced) method, with:
Section Table Name > entries
Option Values > num
Option Labels > list_entry

The first step is to go to your "entries" section and make note of the records numbers for the list_entry records you want to set as default in the multi-value "my_list" check box field when you create a new "listings" section record.

Let's say you pick list_entry 1, 7 and 16.

The second step is to go to section editors under the admin menu and modify the "listings" Section Editor.

Choose the "my_list" field and enter the record numbers you chose (1,7, and 16) into the "Default Values" field, separated by tab characters.

The only method that worked for me was to open my notepad text editor, entering a tab character in a blank text document by pressing the tab key on my keyboard, then highlighting and copying (CTL-C) the tab character to my clipboard. I could then paste (CTL-V) that character after entering each record number.

The result was:

```
1(tab)7(tab)16(tab)
NOTE:(you won't actually see the (tab), but there should be a multi-character long space between the record numbers.
```

Save this field and the section details.

When you create a new record in the "Listings" editor, the "my_list" entries that you entered as defaults will be pre-checked. (You can still uncheck those entries and/or check other entries in the list as required.)

Note that this will only affect new records, it will not change existing records.

BLOGS

CREATING A BLOG (UPDATED FOR MYSQLI) - Jul 22nd, 2019

I recently put together a blog on one of my client's sites and I thought I'd put the recipe up just in case someone could benefit from it.

BACKGROUND:

The blog editor (called 'blog') has a title, date, contents and an image upload field. Image thumbnail2 is set for a max width of 200 px and a max height of 300 px

The client didn't want to allow comments from others so that's not included in the code. If they had, I would have used either the Website Comments plugin or a 3rd party solution like <http://www.gentlesource.com/comment-script/> ,

<http://www.phpjammers.com/post-comment/> ,
http://www.adriantnt.com/products/comments_script/ , OR
<http://js-kit.com/comments/>

The images are padded and wrapped in the contents by the following css:

```
.pad { border-right: 20px solid transparent; border-
bottom: 10px solid transparent; border-top: 5px solid transparent;}
img { border:hidden;}
img.floatLeft {float: left; margin: 10px;}
img.floatRight {float: right; margin: 4px; }
```

I've allowed the visitor to decide if they want to view the earliest or most current posts first and retaining their choice for subsequent searches.

The use of the letters a and b in the form for added security are explained in the post called: ALLOWING VISITORS TO SET VIEWER ORDERBY OPTIONS

The number of records shown on the main viewer page is set in a single record editor called 'common_information'.

A major key to creating the blog is the code that produces a listing of unique months and years for existing articles to act as an archive of blog posts.

Another was being able to display the archive month and year once at the top of the page, next to the word Archives

Here's the code for the main blog page (blog1.php):

```
<span class="heading_font">Have It Your Way</span>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">

<select name="order">
  <option value="b">earliest date at the top</option>
  <option value="a">most recent date at the top</option>
</select>
<input type="submit" name="submit" value="Click/Tap To Choose Display Order">
</form>
<?php $orderBy = "date";
if (@$_FORM['order'] == 'b') { $orderBy = "date"; }
if (@$_FORM['order'] == 'a') { $orderBy = "date DESC"; }

?>
<?php $record_limit = $common_informationRecord['record_limit'] ?>
<?php if (@!$record_limit) { $record_limit1 = ""; } ?>
<?
php if (@$record_limit >= 1) { $record_limit1 = $common_informationRecord['record_limit']; } ?
>

<?php // load records from 'blog'
list($blogRecords, $blogMetaData) = getRecords(array(
  'tableName' => 'blog',
  'loadUploads' => true,
  'allowSearch' => false,
  'orderBy' => $orderBy,

  'limit' => $record_limit1,
));

?>
<table width="50%" border="0" align="center" cellpadding="10">
<tr>
<td width="75%" align="left" valign="top">
<table width="100%" border="0" align="left" cellpadding="2">
  <?php foreach ($blogRecords as $record): ?>
  <tr>
    <td align="left" ><hr />
    <span class="text_font_bold"><?
php echo date("D, M jS, Y ", strtotime($record['date'])) ?>
    <?php echo strtoupper($record['title']) ?></span>
    <div class="pad" style="float:left">
    <div style="float:none">
      <?php foreach ($record['image'] as $upload): ?>
      <a target="_blank" href="<?php echo $upload['urlPath'] ?>">" height="
<?php echo $upload['thumbHeight2'] ?>"
alt="" /></a>
      <?php endforeach ?>
    </div>
    </div>
    <span class="text_font"><?php echo $record['content'] ?></span></td>
  </tr>
  <?php endforeach ?>
</table></td>
<td width="25%" align="left" valign="top"><p><span class="page_title_font">
Or Select
An Archive</span>
<?php
// get list of unique months and years with articles
$query = "SELECT DATE_FORMAT(date, '%M %Y') as dateAndYear, YEAR(date) as year, MONTH(date) as month FROM cms_blog GROUP
BY dateAndYear ORDER BY date";
$result = mysqli()->query($query) or die("MySQL Error: ". htmlspecialchars(mysqli()-
>error) . "\n");
while ($record = $result->fetch_assoc()):
?>
```

```

    <a href="blog2.php?date_year=<?php echo $record['year'] ?>&date_month=<?
php echo $record['month']
?>&orderBy=<?php echo $orderBy ?>"><?php echo $record['dateAndYear']; ?> </a>
    <?php endwhile ?>
</td>
</tr>
</table>

```

And here's the code for the archive page (blog2.php)

```

<span class="heading_font">Have It Your Way</span>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<input type="hidden" name="date_year" value="<?
php echo @$_REQUEST['date_year']; ?>">
<input type="hidden" name="date_month" value="<?
php echo @$_REQUEST['date_month']; ?>">
<input type="hidden" name="orderBy" value="<?
php echo @$_REQUEST['orderBy']; ?>">
<select name="order">
    <option value="b">earliest date at the top</option>
    <option value="a">most recent date at the top</option>
</select>
<input type="submit" name="submit" value="Click/Tap To Choose Display Order">
</form>
<?php $orderBy = @$_REQUEST['orderBy']; ?>
<?php
if (@$FORM['order'] == 'b') { $orderBy = "date"; }
if (@$FORM['order'] == 'a') { $orderBy = "date DESC"; }

?>

<?php // load records from 'blog'
list($blogRecords, $blogMetaData) = getRecords(array(
    'tableName' => 'blog',
    'loadUploads' => true,
    'allowSearch' => false,
    'orderBy' => $orderBy,

));

?>
<table width="50%" border="0" align="center" cellpadding="10">
<tr>
    <td width="75%" align="left" valign="top">
<table width="100%" border="0" align="left" cellpadding="2">
    <tr>
        <td align="center"><?php $date_year = $_REQUEST['date_year']; ?>
        <?php $date_month = $_REQUEST['date_month']; ?>
        <?php
$date = @$_REQUEST['date_year']."-".$_REQUEST['date_month']."-28";
$formattedDate = date("F Y", strtotime($date));
?>
        <span class="page_title_font"><?php echo $formattedDate ?
> Blog Posts</span>
        </td>
    </tr>
</table>
<?php foreach ($blogRecords as $record): ?>
<?php $record_year = date("Y", strtotime($record['date'])) ?>
<?php $record_month = date("n", strtotime($record['date'])) ?>
<?php if (($date_year == $record_year) && ($date_month == $record_month)): ?>
>
    <tr>
        <td align="left" ><hr />
        <span class="text_font_bold"><?
php echo date("D, M jS, Y ", strtotime($record['date'])) ?>
        <?php echo strtoupper($record['title']) ?></span>
        <?php if($record['image']):?>
        <div class="pad" style="float:left">
        <div style="float:none">
            <?php foreach ($record['image'] as $upload): ?>
            <a target="_blank" href="<?php echo $upload['urlPath'] ?>"> " height="
<?php echo $upload['thumbHeight2'] ?>"
alt="" /></a>
            <?php endforeach ?>
        </div>
        </div>
        <?php endif ?>
        <span class="text_font"><?php echo $record['content'] ?></span></td>
    </tr>
    <tr>
        <td align="right">

```

```

<?php endforeach ?>
</table></td>
<td width="25%" align="left" valign="top"><span class="page_title_font">
Or Select
An Archive</span>
<?php
// get list of unique months and years with articles
$query = "SELECT DATE_FORMAT(date, '%M %Y') as dateAndYear, YEAR(date) as year, MONTH(date) as month FROM cms_blog GROUP
BY dateAndYear ORDER BY date";
$result = mysql_query($query) or die("MySQL Error: ". htmlspecialchars(mysql_error()) . "\n");

while ($record = $result->fetch_assoc()):
?>
<a href="blog3.php?date_year=<?php echo $record['year'] ?>&date_month=<?
php echo $record['month'] ?>&orderBy=<?php
echo $orderBy ?>"><?php echo $record['dateAndYear']; ?></a>
<?php endwhile ?></td>
</tr>
</table>

```

USING A PULL DOWN FORM FIELD TO SHOW AVAILABLE DATES - Jun 7th, 2014

Showing available dates issue got a bit more complicated when I wanted to use a pull down form field instead of a list to show the available date options.

I've made this a separate example to keep the code as simple as possible.

For this example you'll need to create a multi-record section with at least a Date field called 'publication_date' and a text field called 'title' and a checkbox field called 'hidden'.

Note: change 'your_table_name' to the name of the table that you've created (in 2 places). Put a viewer_functions.php call at the top of both viewers, and a list records call for your table at the top of your available_date_test2.php viewer.

With a bit of help from Chris Waddell from Interactive Tools, I came up with the following code for my available_date_test.php viewer (the one with the form):

```

<?php
// get list of unique months and years with records
$query = "SELECT DATE_FORMAT(publication_date, '%M %Y') as dateAndYear, YEAR(publication_date) as year,
MONTH(publication_date) as month FROM cms_your_table_name WHERE `hidden` = 0 GROUP BY dateAndYear ORDER BY
publication_date DESC";
$result = mysql_query($query) or die("MySQL Error: ". htmlspecialchars(mysql_error()) . "\n");
?>

<form method="POST" action="available_date_test2.php">
<select name="dateAndYear" style="width:350px;" >
<option value="">Choose A Date</option>
<?php while ($record = mysql_fetch_assoc($result)):?>
<option class="body-text" value="<?php echo $record['dateAndYear'] ?>"><?
php echo $record['dateAndYear'];
?></option><?php endwhile ?></select>
<input type="submit" name="submit" value="Go To Records" style="width:200px;" >
</form>

```

And this code for the available_date_test2.php page that show records filtered by the date chosen:

```

<?php
@list($date_month, $date_year) = explode(' ', @$_REQUEST['dateAndYear']); // create the variables $date_month and
$date_year split on space
?>
<?php
$date = @$_date_year."-".@date_month."-28";

// According to Jason Sauchuk - This creates a "date" variable based on the 2 variables in the string (we use 28 for
the "day" since all months will have at least 28 days. This number isn't important, but it helps php decipher which
variables are months, days, and years).;

$formattedDate = date("F Y", strtotime($date));

// This formats that date into the form Month Year. You can then output $formattedDate where ever you want to.
?>
Records with a date field of <?php echo $formattedDate ?>
<?php foreach ($your_tableRecords as $record): ?>
<?
php $record_year = date("Y", strtotime($record['publication_date'])) // format the $record_year variable
to show only the 4 digit year ?>
<?
php $record_month = date("F", strtotime($record['publication_date'])) // format the $record_month
variable to show the full name of the month ?>

```

```

<?
php if (($date_year == $record_year) && ($date_month == $record_month )):?>
  <?php echo $record['title'] ?><hr>
<?php endif ?>
<?php endforeach; ?>

```

SORTING BLOG POSTS BY KEYWORDS IN MULTIPLE FIELDS AND AN AVAILABLE DATE FILTER ON ONE VIEWER - Jan 7th, 2019

The previous recipes addressed a basic approach to creating a Blog or Articles editor (really the same thing unless you are allowing user posts on your blog).

Here's an approach that puts the pieces together.

It uses a single viewer, and allows visitors to search by multiple field keyword criteria, or to filter their results from a list that only contains available dates (Month and Year), and put all of this in a neat set of pull down form fields.

The approach uses 2 forms (one for the keyword searches and one for the available date filters) and sets a variable called \$archive depending on which one is used (I'm sure that there's a more elegant way to do this, and welcome anyone's input).

The 2 fields that are set for keyword searches in this example are 'article_type' and 'publication', and the date filter uses month and year.

You'll need to set up 3 multi-record tables to follow this example:

- A table called article_types with one text field called 'title'
- A table called publication_names with one text field called 'title'
- A table called Articles with the following minimum set of fields

- A date field called publication_date
- A text field called article_title
- 2 list fields, one called 'article_type' and the other called 'publication' (these get their option values and option labels from the other 2 tables)

Here's the code:

At the top of your page

```

<?php
require_once "/path_to_your_server/html/cmsAdmin/lib/viewer_functions.php";

list($article_typesRecords, $article_typesMetaData) = getRecords(array(
  'tableName' => 'article_types',
));

list($publication_namesRecords, $publication_namesMetaData) = getRecords(array(
  'tableName' => 'publication_names',
));
?>

```

And in the body:

```

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
  <input type="hidden" name="archive" value="0">
  <table >
    <tr>
      <td><select name="article_type_keyword" style="width:350px; ">
        <option value="">Choose An Article Type To Display</option>
        <?php foreach ($article_typesRecords as $record): ?>
          <option value="<?php echo $record['num'] ?>"><?php echo $record['type'] ?>
        </option>
        <?php endforeach ?>
      </select></td>
      <td>
        <input type="submit" name="submit" value="Filter Your Page" style="width:200px; ">
      </td>
    </tr>
    <tr>
      <td><select name="publication_keyword" style="width:350px; ">
        <option value="">And/Or Choose A Publication To Display</option>
        <?php foreach ($publication_namesRecords as $record): ?>

```

```

        <option value="<?php echo $record['num'] ?>"><?
php echo $record['publication_name'] ?></option>
    <?php endforeach ?>
</select></td>
<td>
<INPUT TYPE="submit" VALUE="Cancel Search Filters" style="width:200px;"></td>
</tr>
</table>
</FORM>
<br />
<?php // insert block 1 (Form) for displaying only current year's articles here ?>
<?php
list($articles_by_ericaRecords, $articles_by_ericaMetaData) = getRecords(array(
    'tableName' => 'articles_by_erica',
    'loadUploads' => true,
    // 'allowSearch' => false,
    'orderBy' => 'publication_date DESC',
));

?>
<?php // get list of unique months and years with articles ?>
<?
php // remove this code for years only: %M and , MONTH(publication_date in the $query ?
>
<?
php $query = "SELECT DATE_FORMAT(publication_date, '%M %Y') as dateAndYear, YEAR(publication_date) as year,
MONTH(publication_date) as month FROM cms_articles_by_erica WHERE `hidden` = 0 GROUP BY dateAndYear ORDER BY
publication_date DESC";
$result = mysql_query($query) or die("MySQL Error: ". htmlspecialchars(mysql_error()) . "\n");
?>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<table >
<tr>
<td ><input type="hidden" name="archive" value="1">
<select name="dateAndYear" style="width:350px;" >
<option value="">>Or, Display Articles By Publication Date</option>
<?php
while ($record = mysql_fetch_assoc($result)):?>
<option value="<?php echo $record['dateAndYear'] ?>"><?
php echo $record['dateAndYear']; ?></option>
<?php endwhile ?>
</select></td>
<td>
<input type="submit" name="submit" value="Filter By Date" style="width:200px;">
</td>
</tr>
</table>
</form>

<?php @$archive = $_REQUEST['archive']; ?>
<?php if(@$archive== 0 || @$archive==""):?>
<?
php // beginning of replaced code block 2 for displaying only current year's articles ?>
<br />
<table >
<?php foreach ($articles_by_ericaRecords as $record): ?>
<tr>
<td ><?php echo $record['article_title'] ?>
<?php if (($record['publication'] && !$record['publication'] == "") ||
($record['publication_date'] &&
!$record['publication_date'] == "") ||
($record['article_type'] && !$record['article_type'] == "")):?>
<br />
(
<?php endif ?>
<?php // end of replaced code block 2 for displaying only current year's articles ?>
<?php if ($record['publication'] && !$record['publication'] == ""):?>
<?php echo $record['publication:label'] ?> -
<?php endif ?>
<?php if ($record['publication_date'] && !$record['publication_date'] == ""):?>
<?php echo date("F Y ", strtotime($record['publication_date'])) ?> -
<?php endif ?>
<?php if ($record['article_type'] && !$record['article_type'] == ""):?>
<?php echo $record['article_type:label'] ?>
<?php endif?>
<?php if (($record['publication'] && !$record['publication'] == "") ||
($record['publication_date'] &&
!$record['publication_date'] == "") ||
($record['article_type'] && !$record['article_type'] == "")):?>
)
<?php endif ?>
<br /><hr /></td>
</tr>

<?php endforeach; ?>
</table>

```

```

<?php endif ?>
<?php if(@$sarchive== 1):?>
<br />
<h3>Articles by Publication Date</h3>
<?php // remove this code for years only: $date_month, ?>
<?
php @list($date_month, $date_year) = explode(' ', @$_REQUEST['dateAndYear']); // split on space
?>
<hr />
<table >
  <?php foreach ($articles_by_ericarecords as $record): ?>
  <?php $record_year = date("Y", strtotime($record['publication_date'])) ?>
  <?php // remove this code for years only: <?
php $record_month = date("F", strtotime($record['publication_date'])) ?>

  <?php $record_month = date("F", strtotime($record['publication_date'])) ?>
  <?php // remove this code for years only: && ($date_month == $record_month) ?>
  <?
php if ((@$_REQUEST && $sarchive == 1) && ($date_year == $record_year) && ($date_month == $record_month )):?
  >
  <tr>
    <td ><?php echo $record['article_title'] ?>
      <?php if (($record['publication'] && !$record['publication'] == "") ||
($record['publication_date'] &&
!$record['publication_date'] == "") ||
($record['article_type'] && !$record['article_type'] == "")):?>
        <br />
        (
        <?php endif ?>
        <?php if ($record['publication'] && !$record['publication'] == ""):?>
        <?php echo $record['publication:label'] ?> -

        <?php endif ?>
        <?php if ($record['publication_date'] && !$record['publication_date'] == ""):?>
        <?php echo date("F Y ", strtotime($record['publication_date'])) ?> -
        <?php endif ?>
        <?php if ($record['article_type'] && !$record['article_type'] == ""):?>
        <?php echo $record['article_type:label'] ?>
        <?php endif?>
        <?php if (($record['publication'] && !$record['publication'] == "") ||
($record['publication_date'] &&
!$record['publication_date'] == "") ||
($record['article_type'] && !$record['article_type'] == "")):?>
        )
        <?php endif ?>
        <br />
      <hr /></td>
    </tr>
  <?php endif?>
  <?php endforeach; ?>
</table>
<?php endif?>

```

Two permutations of this approach is to show only the current years articles as default and to list 'sort by date' archives by year instead of month and year.

A) To show only the current year's articles as default:

1)

Search for <?php // insert block 1 (Form) for displaying only current year's articles here ?

> above and insert the

following code:

```

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
  <input type="hidden" name="submitted" value="0">
  <table align="center" width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td align="left">
<INPUT TYPE="submit" VALUE="Clear Other Filters And Show All <?php
echo date("Y")?> Articles" style="width:92%; background-color:#7F0000;"></td>
      </tr>
    </table>
  </FORM>
<br />

```

2)

Search for <?

php // beginning of replaced code block 2 for displaying only current year's articles ?

> and <?php // end

of replaced code block 2 for displaying only current year's articles ?

> above and replace that code with the code below:

```

<?php // begin show this year's articles only code ?>

```

```

<?php // begin show this year's articles only code :>
    <?php if(@$submitted == " " || @$submitted == "0"):?>
    <br />
    <h3>All <?php echo date("Y")?> Articles Only</h3>
    <table align="center" width="100%">
    <?php $curYearCount = '0' ?>
    <?php foreach ($articles_by_ericarecords as $record): ?>
    <?
    php $pubYear1 = date("Y", strtotime($record['publication_date']))?>
    <?php $curYear1 = date("Y")?>
    <?php if ($pubYear1 == $curYear1):?>
    <?php $curYearCount++; ?>
    <?php endif ?>
    <?php endforeach ?>
    <?php foreach ($articles_by_ericarecords as $record): ?>
    <?
    php $pubYear = date("Y", strtotime($record['publication_date'])) ?>
    <?php $curYear = date("Y")?>
    <?php if (($pubYear == $curYear) && $curYearCount >= 1 ):?>
    <tr>
    <td colspan="2"><span class="Medium-Text-Bold"><?
    php echo $record['article_title']
    ?></span><span class="Medium-Text">
    <?php if (($record['publication'] && !$record['publication'] == "")
    ||($record['publication_date'] && !$record['publication_date'] == "") ||
    ($record['article_type'] &&
    !$record['article_type'] == "")):?>
    <br />
    (
    <?php endif ?>
    <?php if ($record['publication'] && !$record['publication'] == ""):?>
    <?php echo $record['publication:label'] ?> -
    <?php endif ?>
    <?
    php if ($record['publication_date'] && !$record['publication_date'] == ""):?>
    <?php echo date("F Y ", strtotime($record['publication_date'])) ?
    > -
    <?php endif ?>
    <?php if ($record['article_type'] && !$record['article_type'] == ""):?
    >
    <?php echo $record['article_type:label'] ?>
    <?php endif?>
    <?php if (($record['publication'] && !$record['publication'] == "")
    ||($record['publication_date'] && !$record['publication_date'] == "") ||
    ($record['article_type'] &&
    !$record['article_type'] == "")):?>
    )
    <?php endif ?>
    </span><br /></td>
    <tr>
    <td>
    <?php if ($record['article_url']): ?>
    <?php if (!preg_match("/^https:\V/i", $record['article_url'])):?>
    <?PHP if (!preg_match("/^http:\V/i", $record['article_url'])) {
    $record['article_url'] = "http://" . $record['article_url']; } ?>
    <?php endif ?>
    <?php endif ?>
    <?php if ($record['image']):?>
    <td style="padding-left:10px; padding-top:10px; text-
    align:left;" align="center" ><?php
    $article_url = $record['article_url'] ?>
    <?php foreach ($record['image'] as $index => $upload): ?>
    <a target="_blank" href="<?php echo $record['article_url'] ?>">
    " height="<?php echo $upload['thumbHeight2'] ?>" alt="" />
    </a><br />
    <?php if($upload['info1']):?>
    <span class="Small-Text">Photo by:<br />
    <?php echo htmlspecialchars($upload['info1']) ?></span>
    <?php endif ?>
    <?php endforeach ?></td>
    <?php else :?>
    <td>&nbsp;</td>
    <?php endif ?>
    <td <?php if(!$record['image']):?>colspan="2" <?php endif?
    > valign="top"
    class="Medium-Text"><?php echo $record['article_description'] ?><br />
    <a target="_blank" href="<?php echo $record['article_url'] ?>"><?
    php echo
    $record['article_link_text'] ?>... (read more)</a>
    <?php $the_article = $record['article_url'] ?>
    <br />
    <?

```

```

php foreach ($on_line_article_commentsRecords as $record): ?>
    <?php if ($record['link_url'] == $the_article ) :?>
    <br />
    <?php if(!preg_match("/^https:\V/i", $record['link_url'] )):?>
    <?PHP if (!preg_match("/^http:\V/i", $record['link_url'])) {
$record['link_url'] = "http://" . $record['link_url']; } ?>
    <?php endif ?>
    <a href="http://www.ericaminer.com/comments.php?match_url=

<?php echo
$record['link_url'] ?>"><?
php echo $common_informationRecord['comments_link_text'] ?></a>
    <?php break ?>
    <?php endif ?>
    <?php endforeach ?>
</td>
</tr>
<tr>
    <td colspan="2"><hr /></td>
</tr>
<?php endif ?>
<?php endforeach; ?>
</table>

    <?php endif?>
    <?
php if (@$curYearCount == 0 && (@$submitted == " " || @$submitted == "0") ): ?>
    <h3>So far, there are no articles published for <?
php echo ($curYear) ?>.<br /> <br />
    Please use the pulldown above to select <?
php echo ($curYear-1) ?> and earlier
articles.</h3><?php endif ?>
    <?php // end show this year's articles only code?>

```

B) To modify the 'Search by Date' archive to display by year instead of month and date you'll need to remove all references to 'month' in the code above. (There are 4)
 Search for the 4 occurrences of remove this code for years only: in the code above and remove the indicated code.
 You can also change any notes to reflect your changes.

IF STATEMENTS

UNDERSTANDING AND USING IF STATEMENTS - Jun 19th, 2011

The if statement is one of the most powerful and valuable operators that you can use when designing your pages. You can test for many different scenarios and have your page output (or not output) specific code, depending on the outcome of your tests.

The problem is that many times, if you've made a mistake in your coding, you'll get a blank page displayed in your browser and have no idea what your mistake was.

So here's a short primer on the correct way to use this powerful tool.

The first rule to remember is that every (series of) if statement(s) must be closed with an <?php endif ?> in order for them to work correctly.

The second rule is that if you're using a series of if statements, then the first one is always an "if", the last is always "else" and any in the middle are "elseif".

So, if you're using only one if statement to test for a particular condition. Say to test whether the field "your_field" is empty, then you'd use:

In a single record viewer:

```

<?php if ($your_tableRecord['your_field']): ?> ...your code...
<?php endif ?>

```

Or in a multi record viewer:

```

<?php foreach ($your_tableRecords as $record): ?>
<?php if ($record['your_field']): ?> ...your code...
<?php endif ?>
<?php endforeach; ?>

```

YOU CAN ALSO PUT AN IF STATEMENT AROUND A BLOCK OF TEXT TO KEEP IT FROM BEING RUN IF THERE ARE NO RECORDS AVAILABLE.

```
<?php if ($your_tableRecord['your_field']): ?>
<?php foreach ($your_tableRecord['your_field'] as $upload): ?>

... code to be run...

<?php endforeach ?>
<?php endif ?>
```

IF YOU WANTED TO TEST FOR PARTICULAR CONTENTS IN "YOUR_FIELD".

Say, if the contents of "your_field" equaled "apples" then, you'd use the exactly equal to "==" operator:

In a single record viewer:

```
<?php if ($your_tableRecord['your_field'] == 'apples'): ?> ...your code...
<?php endif ?>
```

Or in a multi record viewer:

```
<?php foreach ($your_tableRecords as $record): ?>
<?php if ($record['your_field'] == 'apples'): ?> ...your code...
<?php endif ?>
<?php endforeach; ?>
```

IF YOU WANTED TO TEST FOR THE ABSENCE OF PARTICULAR CONTENTS IN "YOUR_FIELD".

Say, if the contents of "your_field" was anything other than "apples" then, you can use the not operator "!" :

In a single record viewer:

```
<?php if (!$your_tableRecord['your_field'] == 'apples'): ?> ...your code...
<?php endif ?>
```

Or in a multi record viewer:

```
<?php foreach ($your_tableRecords as $record): ?>
<?php if (!$record['your_field'] == 'apples'): ?> ...your code...
<?php endif ?>
<?php endforeach; ?>
```

TO USE A SERIES OF IF STATEMENTS TO TEST FOR A SERIES OF CONDITION, THEN YOU'D USE:

in a detail viewer:

```
<?php if ($your_tableRecord['your_field'] == 'apples'): ?
> ...your field contains apples...
<?php elseif ($your_tableRecord['your_field'] == 'bananas'): ?
> ...your field contains bananas...
<?php else: ?> ...your code...
<?php endif ?>
```

And in a list viewer:

```
<?php foreach ($your_tableRecords as $record): ?>
<?php if ($record['your_field'] == 'apples'): ?> ...your field contains apples...
<?php elseif ($record['your_field'] == 'bananas'): ?> ...your field contains bananas...
<?php else: ?> ...your code...
<?php endif ?>
<?php endforeach; ?>
```

YOU'RE NOT LIMITED TO A SINGLE OR A SINGLE FIELD CONDITION.

To test for multiple conditions, you can use:

(Note: The "!" works in this too)

```
<?
php if ($your_tableRecord['field_a'] == 'apples' && $your_tableRecord['field_b'] == 'bananas' &&
$your_tableRecord['field_c'] == 'pears'): ?> ...Your record has all three...
<?php endif ?>
```

Or

```
<?php foreach ($testRecords as $record): ?>
<?
php if ($record['field_a'] == 'apples' && $record['field_b'] == 'bananas' && $record['field_c'] == 'pears'): ?
>
...Your record has all three...
<?php endif ?>
<?php endforeach; ?>
```

IF YOU WANTED TO TEST FOR THREE CONDITIONS IN A LIST VIEWER RECORD.

Say, "field_a" empty, "field_b" empty and "field_c" not empty, you could use:

```
<?php if (!$record['field_a'] && !$record['field_b'] && $record['field_c']): ?
> ...your code...
<?php endif ?>
```

IF YOU WANTED TO TEST FOR ANY OF A SERIES OF CONDITIONS.

Say a or b or c, you could use:

```
<?
php if ($your_tableRecord['your_field'] == "apples" || $your_tableRecord['your_field'] == "bananas" ||
$your_tableRecord['your_other_field'] == "pears"): ?>
Your code...
<?php else: ?>
Your code...
<?php endif ?>
```

Or

```
<?
php if ($record['your_field'] == "apples" || $record['your_field'] == "bananas" || $record['your_other_field'] ==
"pears"): ?>
Your code...
<?php else: ?>
Your code...
<?php endif ?>
```

IF YOU WANT TO TEST TO SEE IF A FIELD CONTAINS A PARTICULAR ALPHANUMERIC PATTERN

Like seeing if apples is included in a multi item list, you can use the strpos function. Theoretically the strpos function returns the numeric position of the first occurrence of the pattern that you're searching for, however, before it can do that it has to decide if the pattern exists in the string.

To test for the pattern "apples" in the string "bananas, apples, pears, raisins" you could use something like:

```
<?php if (strpos($your_table[your_field], 'apples')): ?>
Your code...
<?php endif ?>
```

This is by far not a comprehensive list of the possibilities for using if statements, but it should be enough to get you started.

Before leaving this topic, I was curious as to the rules for requiring an <?php endif; ?> to close an "if" statement.

Jason Sauchuk, a programmer with Interactive Tools cleared up the question:

"You only need <?php endif ?> if your "if" statement covers more than 1 <?php ?> block.

Example:

```
<?php if($x==1)?>
---some code---
<?php endif ?>
```

This could also be put into 1 <?php ?> block with no <?php endif ?>, as in:

```
<?php if($x==1){
---some code--
}
?>
```

IF STATEMENTS THAT MEET MORE THAN ONE CONDITION - Aug 3rd, 2010

If you've got a table called "people" and two fields "price" and "description" and you need a specific result depending on whether there's information in one, the other or both fields, you can approach it this way. Remember that the operator "!" means not so !\$record['price'] means that there's no information in the price field.

```
<!-- If there's no price and no description, just show the title. -->
<?PHP foreach ($peopleRecords as $record): ?>
<?PHP if (!$record['price'] && !$record['description']): ?><?PHP echo $record['title'] ?
><?PHP endif ?>
```

```
<!-- If there's no price, just a description, offer information only. -->
<?PHP if (!$record['price'] && $record['description']): ?><a href="#"?
PHP echo $record['_link'] ?>><?PHP
echo $record['title'] ?></a>Click the title for more information.<?PHP endif ?>>
```

```
<!-- If there's a price and no description, ask for the sale. -->
<?PHP if ($record['price'] && !$record['description']): ?><a href="#"?
PHP echo $record['_link'] ?>><?PHP echo
$record['title'] ?></a>Click the title to buy.<?PHP endif ?>
```

```
<!-- If there's a price and a description, offer information and ask for the sale. -->
<?PHP if ($record['price'] && $record['description']): ?><a href="#"?
PHP echo $record['_link'] ?>><?PHP echo
$record['title'] ?></a>Click the title the for more information or to buy.<?PHP endif ?>
<?PHP endforeach ?>
```

```
<?php endforeach ?>
```

USING IF STATEMENTS TO DEFINE OUTPUT BASED ON A FILE EXTENSION - Aug 16th, 2011

If you need to publish specific code that's based on the type of file that you've uploaded into an upload field, like a .swf flash file, here's the trick. Just use:

```
<?php if ($upload['extension'] == 'swf'): ?>
```

or for the second through next to last if parameter in the list (the last one gets an <?php else: ?> instead of a <?php elseif...)

```
<?php elseif ($upload['extension'] == 'swf'): ?>
```

You can use this for non upload fields as well:

```
<?php if ($your_fieldRecord['extension'] == 'doc'): ?>
```

or

```
<?php elseif ($your_fieldRecord['extension'] == 'doc'): ?>
```

Don't forget that every if statement must be closed with an <?php endif ?>

SETTING UP "IF" STATEMENTS FOR NO RECORDS MATCHING A CRITERIA - Apr 1st, 2012

This one plagued me for a while until Jason Sauchuk, a programmer with Interactive Tools came to the rescue.

I had a checkbox field in a multi-record editor and I wanted to be able to output a "no records found with the checkbox checked" message if there were no records with the checkbox checked.

Here's what he offered:

```
<?php $ischeck= 0; ?>
<?php foreach ($e_blast_events_noticeRecords as $record): ?>
  <?php if($record['apbc_event']==1) $ischeck=1; ?>
<?php endforeach ?>

<?php if ($ischeck==0) echo "no records found."; ?>
```

I could have also added:

```
<?php if ($ischeck==1) echo "records were found."; ?>
```

In the previous recipe, I was using a "where" statement to restrict which dates are displayed in a group. The code I used to show only events that had a start date or a reception date in the next 7 days was:

```
<?php
list($my_tableRecords, $my_tableMetaData) = getRecords(array(
    'Table name' => 'my_table',
    'where' => '(NOW() + INTERVAL 7 DAY) >= event_reception_date AND event_reception_date >= TIMESTAMP(CURDATE(),
"00:00:00") OR (NOW() + INTERVAL 7 DAY) >= event_start_date AND event_start_date >= TIMESTAMP(CURDATE(), "00:00:00");
));

?>
```

If there were no records that met the criteria, I wanted to output the same kind of "no records found" notice.

This was one I should have been able to figure out but I just couldn't come up with a solution. Again, Jason came to the rescue. He suggested a simple solution:

```
<?
php if(!$e_blast_events_noticeRecords) echo "No events are opening or have a reception during the next 7 days.";
?>
```

And

```
<?
php if($e_blast_events_noticeRecords) echo "These events are opening or have a reception during the next 7 days."; ?
>
```

USING IF STATEMENTS AND RECORD COUNTS TO CHANGE PAGE CONTENTS - Nov 25th, 2010

I recently had an artist client who's portfolios were set up to show a list page of portfolio categories, a thumbnail page showing all of the images that belonged to a particular category, and a detail page that showed details about specific images.

They wanted to be able to link visitors directly to a thumbnail page if there was only one category in their portfolio, and to show the list page only if there was more than one category.

I was able to easily accomplish their goal by counting the visible records in the portfolio category section. (If records were "hidden" they were not counted.)

The portfolio categories were listed in a section called, "portfolio_name_details", and a thumbnail page was called, "portfolio_album.php"

I inserted the following code in the "Portfolio" link of the navigation menu:

```
<a href="<?php if ($portfolio_name_detailsMetaData["totalRecords"] == 1): ?
>portfolio_album.php<?php else:
?>portfolio.php<?php endif ?>">Portfolio</a>
```

You can use the same approach to change the page content and perform other tasks.

NOTE: Don't forget to include the record count table in on each viewer's "get records" call. Mine looked like this:

NOTE: Don't forget to include the record count table in on each view's 'getRecords' call. This looked like this:

```
list($portfolio_name_detailsRecords, $portfolio_name_detailsMetaData) = getRecords(array(
    'tableName' => 'portfolio_name_details',
));
```

USING THE NUMBER OF RECORDS MEETING A MYSQL QUERY TO CHANGE WHAT A VIEWER SHOWS - Feb 1st, 2011

I had a page with a number of "Where" statements setting the criteria for which records would appear on groups on a page.

What my client wanted to do was hide all reference to any group of records where there were no records that met those criteria.

Example. The heading HAPPENING THIS WEEK if there were no records with start dates falling within the next 7 days.

Here's the simple if statement code that worked for me.

```
<?php
list($eventsRecords, $eventsMetaData) = getRecords(array(
    'tableName' => 'events',
    'where' => 'start_date > TIMESTAMP(NOW())+ INTERVAL 7 DAY',

    'orderBy'=> 'end_date ASC',
));
?>

<?php if (!$eventsMetaData['totalRecords'] == 0): ?>

Show this code....

<?php else: ?>

Show some other code...

<?php endif ?>
```

ALLOWING VISITORS TO SET VIEWER ORDERBY OPTIONS IN A VIEWER - Sep 5th, 2022

A while back user rjbathgate wanted to enable visitors to determine the order that results are displayed in a viewer from a series of options.

Dave Edis from Interactive Tools offered:

First create a search form on the viewer, something like:

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<select name="order">
<option value="a">price (highest first)</option>
<option value="b">price (lowest first)</option>
...etc...
</select>

<input type="submit" name="submit" value="Search">
</form>
```

Then add some code to the list records call at the top of the viewer code

```
$orderBy = "";
if (@$FORM['order'] == 'a') { $orderBy = "price DESC"; }
if (@$FORM['order'] == 'b') { $orderBy = "price"; }
# ... etc ...

list($your_tableRecords, $your_tableMetaData) = getRecords(array(
'tableName' => 'your_table',
'orderBy' => $orderBy,
));
```

NOTE: According to Dave Edis from Interactive Tools, the reason to do it that way by passing a letter (or word or code, it doesn't matter) and testing for that instead of just specifying the order by in the option value directly is because you don't want users to be able to pass MySQL directly into your program or it's a security risk.

You can expand this idea to create as complex a set of criteria as required.

ALLOWING VISITOR TO SET WHERE VALUES IN A VIEWER - Sep 5th, 2022

In this example you're trying to limit the records shown to those that match the value of a field called project_title, so first you'll need to create a list field called project_title in your table.

Then, we'll assume that in any record the values for that field can be either Test event 1 or Test event 2.

At the top of your page in the load records calls use the code:

```

$where = "";
if (@$FORM['where'] == 'a') { $where = 'Test Event 1'; }
if (@$FORM['where'] == 'b') { $where = 'Test Event 2'; }

list($$your_tableRecords, $$your_tableMetaData) = getRecords(array(
    'tableName' => '$your_table',
    'where'      => " project_title = '$where' ",
));

```

And in the body, the form code would be:

```

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<select name="where">
<option value="">Select</option>
<option value="a">Event 1</option>
<option value="b">Event 2</option>
</select>

<input type="submit" name="submit" value="Choose an exhibition to View">
</form>

```

And Here's Another example of limiting the records shown on a viewer page. This one has three possible options and incorporates check boxes in a single record editor to determine which options are available to the visitor:

At the top of the viewer page in the list records calls

```

$where = "";
if (@$FORM['where'] == 'a') { $where = "want_this_book = '1'"; }
if (@$FORM['where'] == 'b') { $where = "get_better_copy = '1'"; }
if (@$FORM['where'] == 'c') { $where = "get_better_copy = '1' OR want_this_book = '1'"; }

list($booksRecords, $booksMetaData) = getRecords(array(
    'tableName' => 'books',
    'loadUploads' => true,
    'allowSearch' => true,
    'where'      => $where,

    // 'debugSql' => true,
));

```

?>

In The Body:

```

<form method="POST" action="">
<select name="where">
<option value="">Make a selection from this pulldown menu </option>
<?php if ($common_informationRecord['sort_by_want_these_books'] == '1'):?>
<option value="a">Search For Books I Want But
Don't Have</option><?php endif ?>
<?
php if ($common_informationRecord['sort_by_better_copy_of_these_books'] == '1'):?
><option value="b">Search For Books I
Want A Better Copy Of</option><?php endif ?>
<?php if ($common_informationRecord['sort_by_want_these_books'] == '1' &&
$common_informationRecord['sort_by_better_copy_of_these_books'] == '1'):?>
<option value="c">Search For All The Books I
Want To Add To My Collection</option><?php endif ?>
</select>

<input type="submit" name="submit" value="Sort By Your Selection">
</form>

```

NOTE: According to Dave Edis from Interactive Tools, the reason to do it that way by passing a letter (or word or code, it doesn't matter) and testing for that instead of just specifying the order by in the option value directly is because you don't want users to be able to pass MySQL directly into your program or it's a security risk.

You can expand this idea to create as complex a set of criteria s required.

ANOTHER NOTE: Jason Sauchuk offered this mini tutorial on the use of single and double quotes. He said:

In PHP, a string that is set with double quotes can have variables inserted directly into it without concatenation.

example:

<?php

```
<?php
$myName = "Jason";
$greeting = "Hello, my name is $myName";
?>
```

The value of \$greeting would be:
Hello, my name is Jason

If we took this same piece of code and used single quotes:

```
<?php
$myName = "Jason";
$greeting = 'Hello, my name is $myName';
?>
```

The value of \$greeting would be:
Hello, my name is \$myName

PHP will not put the value of \$myName into the string.

ALLOWING VISITOR TO SET WHERE VALUES FROM MASTER VALUES LIST - Feb 15th, 2013

For this example I wanted my client to be able to filter the Exhibition records displayed by choosing possible project_title 'where' values from a pre-determined list of Exhibition titles. This would insure that all the requests were consistent when filtering the records to be shown, and would keep all the possible values in a CMSB table, instead of needing to hard code them into the viewer code.

I also wanted to make sure that I was not creating a security risk as described by Dave Edis from Interactive Tools, who said:

"By passing a letter (or word or code, it doesn't matter) and testing for that instead of just specifying the order by in the option value directly, you won't allow users to pass MySQL directly into your program and create a security risk."

To accomplish this task, I created a multi-record editor called master_exhibition_list which has only one text field called title. This way each allowed Title was in it's own record.

Then in the head of my viewer, (with a lot of help from Jason Sauchuck from Interactive Tools), I inserted the following code:

```
<?php
list($master_exhibition_listRecords, $master_exhibition_listMetaData) = getRecords(array(
'tableName' => 'master_exhibition_list',

));
?>

<?php
$numToName = array();
foreach ($master_exhibition_listRecords as $record){
    $numToName[$record['num']] = $record['title'];
}
?>

<?php
$where = "";
?>

<?php foreach ($master_exhibition_listRecords as $record): ?>

<?php if (@$_FORM['where'] == $record['num']) { $where = $record['title'];}?>
<?php endforeach; ?>

<?php
list($client_uploadsRecords, $client_uploadsMetaData) = getRecords(array(
'tableName' => 'client_uploads',
'where' => " project_title = '$where'",

));
?>
```

Then for the form that selects the values (again with a lot of help from Jason) I used:

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<select name="where">
<option value=""><span class="body-text-bold">Select</span></option>
<?php foreach($numToName as $num => $name): ?>
<option value="<?php echo $num;?>"><?php echo $name;?></option>
<?php endforeach?>
</select>

<input type="submit" name="submit" value="Select An Exhibition And Click To View">
</form>
```

ALLOWING ADMINS TO SET SORT ORDERBY VALUES BASED ON A LIST FIELD SELECTION - Sep 5th, 2022

I needed to let an site admin decide whether a viewer would show the data on a list page with the newest record in the table 'studio_art_images' first, or show the oldest record first, show the records in a random order, or in the order they appeared in the record list.

With a bit of help from Interactive Tools Guru Daryl Maximo, here's what we came up with.

1) set up a single value list field in a single record editor (in this example it's 'sort_order' and the section is named 'common_information')

2) enter the possible list values as:

Oldest First
Newest First
Show in Random Order
None

At the top of the viewer use the code:

```
<?php
list($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName' => 'common_information',
    //debugSql => true,
    'limit' => '1',
));
$common_informationRecord = @$common_informationRecords[0]; // get first record
?>
<?php $orderBy = ""; ?>
<?php
if (strpos($common_informationRecord['sort_order'], 'Oldest') !== FALSE) { $orderBy = 'createdDate ASC'; };
if (strpos($common_informationRecord['sort_order'], 'Newest') !== FALSE) { $orderBy = 'createdDate DESC'; };
if (strpos($common_informationRecord['sort_order'], 'Rand') !== FALSE) { $orderBy = 'RAND()'; };
if (strpos($common_informationRecord['sort_order'], 'None') !== FALSE) { $orderBy = ""; };

?>
<?
php list($studio_art_imagesRecords, $studio_art_imagesMetaData) = getRecords(array(
    'tableName' => 'studio_art_images',
    //debugSql => true,
    'limit' => '1',
));
$studio_art_imagesRecord = @$studio_art_imagesRecords[0]; // get first record
?>
```

```
'orderBy' => $orderedBy,  
    ));  
?>
```

Then display the record data in your viewer using standard foreach loop code.

According to Daryl,

Without the `!== FALSE` the `strpos()` function returns the position of the first occurrence of a substring in a string so it could be because it's returning position "0", which is a boolean false, that causes none of your if statements to be true.

You can use the `!==` or the `===` operator to check for "0" as a value and not a boolean false.

VIEWERS - DETAIL PAGES

SPLITTING DETAIL PAGES - Aug 3rd, 2010

Instead of a single detail page for a record I needed to have some of the detail information on one detail page and other information on another detail page and link the 2 of them so I could navigate back and forth between them.

I was lost until Dave from Interactive Tools showed just how easy that was. He said:
You just need to pass the record number along when you link back and forth between detail pages.

So if you're on `detailpage1.PHP` and you've loaded record 23, then you'd like to link to `detailpage2.PHP`, use this:

```
<a href="detailpage2.php"?<?PHP echo $record['num'] ?  
>">link text to detail page 2</a>
```

Another way to do it, if you just want to pass along whatever is after the `?` in the url is like this:

```
<a href="detailpage2.php"?<?PHP echo @$_SERVER['QUERY_STRING'] ?  
>">link text to detail page 2</a>
```

USING MORE THAN ONE DETAILS PAGE DEPENDING ON CONDITIONS - Aug 3rd, 2010

Let's say that you have a series of fields in a multi record editor and you'd like to be able to show certain information for one type of record and other information for other types of records.

I was working with an author who needed to display one subset of information for books that had been published and another subset for those that had not yet been published.

Based on some direction from Dave at Interactive Tools, I was able to:

- 1) Create a multi record editor with a special checkbox field to indicate if the book was published or unpublished.
- 2) Set up 2 details pages, one for the published books and another for those that were not yet published.
- 3) Use the following if statements in my list page to determine which details page would be shown and the record number that would be used: (styling information deleted for clarity).

```
<table>
<tr>
<?PHP foreach ($booksRecords as $record): ?>

<?PHP if ($record['published'] == '1'): ?>
  <td>
  <a href="bookdetail1.php?<?PHP echo $record['num'] ?>"> <?
  PHP foreach ($record['list_image'] as $upload): ?> " height="<?PHP echo
  $upload['thumbHeight'] ?>" alt="" /></a>

</td>
<td>
<a href="bookdetail1.php?<?PHP echo $record['num'] ?>"><?
  PHP echo $record['title'] ?></a>
<?PHP endforeach ?>
<?PHP endif ?>

  <?PHP if ($record['published'] == '0'): ?>
  <td">
  <a href="bookdetail2.php?<?PHP echo $record['num'] ?>"> <?
  PHP foreach ($record['list_image'] as $upload): ?> " height="<?PHP echo
  $upload['thumbHeight'] ?>" alt="" /></a>
</td>
<td>
<a href="bookdetail2.php?<?PHP echo $record['num'] ?>"><?
  PHP echo $record['title'] ?></a>

<?PHP endforeach ?>
<?PHP endif ?></div></td>

<?PHP $maxCols=2; if (@++$count % $maxCols == 0): ?></tr><tr><?PHP endif; ?>

  <?PHP endforeach ?>
</tr>
</table>
```

ITEMS FROM A SINGLE RECORD EDITOR NOT SHOWING ON MULTI RECORD DETAIL PAGES - Aug 3rd, 2010

I had an image that came from a single record that I wanted to appear on my details pages from a multi record viewer. It showed up on the first record in the list but gave me an error saying that the argument supplied for foreach() was invalid for other records.

Dave had an insight, He said:

Try removing this line from the single page call in the details page header :

```
'where' => whereRecordNumberInUrl(1),
```

It worked like a charm.

SUGGESTING "ARTICLES YOU MIGHT ALSO LIKE" BASED ON KEYWORDS - May 8th, 2011

User Illume Magazine wanted to be able to suggest articles on their article detail page that, based on matching keywords, might be of interest to readers of a specific article.

This concept could also be used for showing products that share certain qualities to potential buyers.

Here's what a collaboration between Chris Waddell from Interactive Tools and CMSB user Perchpole came up with:

For this recipe you'll need to add a "your_keyword_field" text field to the multi-record editor that contains your articles.

Here's the code that goes at the top of your detail viewer:

Insert the code between the lines INSERT THIS CODE and END OF INSERTED CODE into your viewer and change the path to your viewer_functions.php and the names of your table and fields.

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/path_to_your_server/', './', '../', './../', './../..');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; } }
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

// load records
list($your_products_tableRecords, $your_products_tableMetaData) = getRecords(array(
    'tableName' => 'your_products_table',
    'where'      => whereRecordNumberInUrl(1),
    'limit'      => '1',
));
$your_products_tableRecord = @$your_products_tableRecords[0]; // get first record

// show error message if no matching record is found
if (!$your_products_tableRecord) {
    header("HTTP/1.0 404 Not Found");
    print "Record not found!";
    exit;
}
//INSERT THIS CODE

// construct a where clause out of this record's keywords (e.g. "apples, oranges" -
> "keywords LIKE '%apples%' OR
keywords LIKE '%oranges'")

$keywords = preg_split('/\s*/', $articlesRecord['your_keyword_field']);

$where = '0';
foreach ($keywords as $keyword) {
    $where .= ' OR ';
    $where .= "your_keyword_field LIKE '%" . mysql_escape($keyword) . "%";
}

// find the most recent records with at least one matching keyword (but not including the current record)
list($similarRecords)= getRecords(array(
    'tableName' => 'your_products_table',
    'where'      => "(" . $where . ") AND num != " . $your_products_tableRecord['num'],
    'limit'      => 5,
));
// END OF INSERTED CODE
?>
```

Then, where you want to display the titles of, and links to similar articles on the detail page, use the following code:

```
<?php foreach ($similarRecords as $record): ?>
<a href="<?php echo $record['_link'] ?>"><?php echo $record['title'] ?></a><br /><?php endforeach ?>
```

The regular expression in the first line of code above will create keywords split on a comma followed by any number of spaces. Here it is again for reference:

```
$keywords = preg_split('/,\s*/', $articlesRecord['keywords']);
```

For example, "a quick, brown fox" would turn into two keywords: "a quick" and "brown fox".

If you want to split on commas and/or spaces, you can use this regular expression instead:

```
$keywords = preg_split('/[, \s]+/', $articlesRecord['keywords']);
```

For example, "a quick, brown fox" would turn into four keywords: "a", "quick", "brown", and "fox".

NOTE: You can also use this one to list keywords in a multi-value list field.

** If you're going to use this in a multi value situation, you'll need to insure that there are no spaces between multi word "keywords. You can use "_" or "-

" to separate multiple words and then if you need to print out a keyword list for some reason you can use another regular expression like the one below to replace each "_" or "-" with the required spaces.

```
<?php foreach ($your_tableRecords as $record): ?>
<!-- Insert This Code -->
<?PHP $record['your_field'] = preg_replace("/[-_]/", "&nbsp;", $record['your_field'] ); ?
>
<!-- End of Insert-->
<?php echo $record['your_field'] ?>
<?php endforeach; ?>
```

You can learn more about the power of regular expressions in the recipe called: "REPLACING ONE CHARACTER WITH ANOTHER USING "REGULAR EXPRESSIONS""

VIEWERS - DISPLAYING

FORCING ONLY TEXT TO BE DISPLAYED ON YOUR PAGES - Aug 3rd, 2010

If your editors are savvy they can force html code to be displayed in what you've set up as a text only field with some potentially negative effects on your page layout.

Here's a hidden gem from Dave at Interactive Tools that can help to guard against that. According to Dave, you can force the entry to be displayed as text (even if they enter HTML) by adding "htmlspecialchars" in the display tag like this:

```
<?PHP echo htmlspecialchars($record['content']); ?>
```

That should dissuade any hot shot editors from bending the rules.

ELIMINATE BLANK LINES OR HEADINGS WHEN THERE'S NO INFORMATION IN A FIELD - Aug 3rd, 2010

Dwelling asked the question and Dave came to the rescue. He suggested to start by experimenting with the PHP "if" tag.

```
<?PHP if ($newsRecord['title']): ?>
  Title: <?PHP echo $newsRecord['title'] ?>
<?PHP endif; ?>
```

Here's one that was used in a multi record editor to eliminate the word "through" if there was no "end date" and the heading "Reception" if that field was blank:

```
<?PHP foreach ($exhibitionsRecords as $record): ?>
  <tr valign="top" align="center">
    <td >
      <span class="heading-text-bold"><?PHP echo $record['title'] ?></span><br />
      <span class="body-text"> <?PHP echo $record['start_date'] ?>&nbsp;</span>
      <span class="body-text"> <?PHP if ($record['end_date']): ?>Through&nbsp;<?
      PHP echo $record['end_date'] ?></span><br />
      <?PHP endif; ?>
      <?PHP if ($record['reception_date_and_time']): ?> <span class="heading-text-
      bold">Reception</span><br /><span
      class="body-text"> <?PHP echo $record['reception_date_and_time'] ?> </span>
      <?PHP endif; ?>
    </td>
  </tr>
  <tr>
    <td align="left" height="125" valign="top"><div class="body-text"><?
    PHP echo $record['content'] ?></div></td>
  </tr> <tr>
    <td align="center">
      <!-- display image --> <?PHP foreach ($record['image'] as $upload): ?>
      <?PHP if ($upload['hasThumbnail']): ?>
      " height="<?PHP echo
      $upload['thumbHeight'] ?>" alt="" />
      <?PHP elseif ($upload['isImage']): ?>
      " height="<?PHP echo $upload['height']
      ?>" alt="" />
      <?PHP else: ?>
      <a class="special" href="<?PHP echo $upload['urlPath'] ?>">DOWNLOAD<?
      PHP echo $upload['filename'] ?></a>
      <?PHP endif ?>
      <div class="body-text-bold">
      <?PHP echo $upload['info1']; ?><br />
      <?PHP echo $upload['info2']; ?> </div></td>
```

```
<?PHP endforeach ?>
```

Basically you just put the field you want to test in between the brackets. Note that this won't display for blank - or - zero.

CHECKING FOR SPACES IN OTHERWISE BLANK FIELDS WHEN USING IF STATEMENTS - Mar 7th, 2023

When using "if" statements to suppress titles and other information in a viewer if a field is empty, it's easy for client's to leave blank spaces in an otherwise empty field and defeat the "if" test.

Now, thanks to a plugin (CMSB version 2.04 or above required) written by Dave Edis of Interactive Tools, it's even easier to guard against this by checking to see if the content is text and not just a space and automatically deleting the space before saving the record so that they can't interfere with any "if" tests. It works in both text and WYSIWYG fields

You can download this plugin from:

<http://www.thecmsbcookbook.com/downloads/removeExtraWhitespace3.zip>

The plugins won't delete errant spaces in otherwise blank fields that already exist in your saved records, but you can still check for them in each if statement used on your page.

To a normal "if" statement:

```
<?php if ($yourRecord['title']): ?>
  Title: <?php echo $yourRecord['title'] ?>
<?php endif ?>
```

Just add this code:

```
&& !ctype_space($yourRecord['title'])
```

And the result:

```
<?php if ($record['title'] && !ctype_space($record['title'])): ?>
  <?php echo $record['title'] ?>
<?php endif ?>
```

(Which means "if title has a value AND it's NOT (!) all spacing characters, then show it" will solve the problem.)

According to Dave Edis, This technique will not work if the content has PHP tags in it, but will for whitespace.

Also note this will not strip leading or trailing spaces from the content. To do that gets a little more complex.

CHECKING FOR BLANK DATE FIELDS - Mar 7th, 2023

The second version of Dave Edis' ExtraWhitespace plugin (CMSB version 2.05 or above required) adds the ability to check for blank date fields to checking for white space in otherwise empty text fields, and eliminate the possibility of returning a Jan 1, 1970 or similar date, if a blank date field is encountered.

You can download this version of the plugin from:

<http://www.thecmsbcookbook.com/downloads/removeExtraWhitespace3.zip>

MATCHING A PATTERN OR REPLACING ONE CHARACTER WITH ANOTHER USING "REGULAR EXPRESSIONS" - Jun 3rd, 2017

REPLACING CHARACTERS

Sometimes you can run into problems when there's a space, a dash or underscore between words in a field value. I ran

Sometimes you can run into problems when there's a space, a dash or underscore between words in a field value. I ran into this recently with an implementation of Lightbox.

My client wanted to use a 2 word value for a field and Lightbox wanted to see a single word with no spaces. If I wasn't displaying this value in other places on the page I could have put a dash or underscore in the value instead of a space.

I chose to use a dash to make Lightbox happy and to replace the dash with a space where it appeared on the page.

Here's how for a multi record editor:
(thanks to Chris Waddell at Interactive Tools for the code syntax)

Just insert the line below into your code.

```
<?php foreach ($your_tableRecords as $record): ?>
  <!-- Insert This Code -->
  <?PHP $record['your_field'] = preg_replace("/[-]/", "&nbsp;", $record['your_field'] ); ?>
  <!-- End of Insert-->
  <?php echo $record['your_field'] ?>
<?php endforeach; ?>
```

Chris explained:

/[-]/ means "match a single character in the set [-]".
preg_replace() will repeat the match as many times as it can.

It's also possible to match more than one character in the set by adding the additional characters as shown below.
Changing the code above to this would replace any occurrence of a dash, an underscore, an r or an x with a space. The replacement will be case sensitive.

```
<?PHP $record['your_field'] = preg_replace("/[-_rx]/", "&nbsp;", $record['your_field'] ); ?>
```

To make the replacement case insensitive, just add an "i" to the end of the expression, like this:

```
<?PHP $record['your_field'] = preg_replace("/[-_rx]/i", "&nbsp;", $record['your_field'] ); ?>
```

If you wanted to replace 2 different characters with 2 different ones, you'd do it by adding a second line of "preg_replace" code like this:

```
<?PHP $record['your_field'] = preg_replace("/[-]/", "&nbsp;", $record['your_field'] ); ?>
<?PHP $record['your_field'] = preg_replace("/[_]/", "*", $record['your_field'] ); ?>
```

The above code will replace any occurrence of a dash ("-") in the string with a space (" "). Then it will replace any occurrence of an underscore ("_") with an asterisk ("*").

For a single record editor, the code syntax would be.

```
<?PHP $your_tableRecord['your_field'] = preg_replace("/[-_rx]/", "&nbsp;", $your_tableRecord['your_field'] ); ?>
```

If you want to replace a complete expression like rx with another character then remove the forward slashes around the expression

So This:

```
<?PHP $your_tableRecord['your_field'] = preg_replace("/[rx]/", "&nbsp;", $your_tableRecord['your_field'] ); ?>
```

Would become this:

```
<?PHP $your_tableRecord['your_field'] = preg_replace("[rx]", "&nbsp;", $your_tableRecord['your_field'] ); ?>
```

When using Lightbox and other utilities, an extra double quote can cause the code to become broken. If you are using an uploads info fields you can't specify characters as allowed, so I use this little trick to replace any inadvertent double quotes with 2 single quotes, which looks (almost) the same.

```
<?PHP $upload['info1'] = preg_replace("[\"]", "'", $upload['info1'] ); ?>
```

In order to replace backslashes they must be escaped. Here's how to replace all backslashes with nothing:

```
<?PHP $your_tableRecord['your_field'] = preg_replace("[\\\\]", "", $your_tableRecord['your_field'] ); ?>
```

or

```
<?PHP $your_variable = preg_replace("[\\\\]", "", $your_variable ); ?>
```

According to Dave Edis, there are many other uses for "regular expressions". He commented: "Basically regular expressions are ways of describing patterns. They're used to matching, and replacing content. So this is useful for validating input (such as determining if an url, credit card number, phone number, or email is in the right format), replacing content (stripping html tags, adding extra tags or attributes, etc, or parsing out data (matching all the email addresses in a web page, parsing a CSV file and importing the data, or extracting some data (such as the current temp from a weather web page to use somewhere else).

In PHP all the regular expression functions start with preg_ and in the CMS Builder code base we use those functions in over 200 places. So they come in quite handy. If your web page editor software has a "Find in Files" or search multiple files feature you can search the CMSB code for "preg_" and see some of the ways we use it."

MATCHING A PATTERN

If you wanted anything other than between 11 to 14 characters that were either numbers or spaces in a phone number field and throw an error message if the criteria was not met, you could use something like:

```
if(!preg_match("/^[0-9]{11,14}$/", @$phone )){
    $errorsAndAlerts .= "Phone numbers must be between 10 and 14 characters long and can not contain anything but numbers and spaces \n";}
```

You can see a list of the PHP functions here:

<http://ca3.php.net/manual/en/ref.pcre.php>

Here's a link to a regex tutorial:

<http://www.regular-expressions.info/tutorialcnt.html>

Here's a link to a regex cheat sheet:

http://www.bitcetera.com/page_attachments/0000/0030/regex_in_a_nutshell.pdf

And a few on-line regex testers/debuggers:

<http://www.regexplanet.com/advanced/java/index.html>

<https://regex101.com/>

REGULAR EXPRESSION QUICK REFERENCE - Oct 15th, 2017

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
/[^0-9./] remove all non numerical characters and .

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

ONLY ECHO THE FIRST WORD OF A STRING - Mar 15th, 2012

I had a situation where I needed to strip all but the first word of a string and use that as part of a mailto:link but I also wanted to use the entire string in the link text.

I found that I could echo a specific word from an array but I was starting with a multi-word text string.

So first I converted the string to an array with:

```
<?php $position = mytableRecord['myfield'] ?>
<?php $myArray = explode(" ", $position);?>
```

Then I echoed the first element (which has an index of [0]) in the array with:

```
<?php echo strtolower($myArray[0]);?>
```

Then I placed all that in my mailto: link

```
<a class="special" href="mailto:<?php $myArray = explode(" ", $position);?><?php echo strtolower($myArray[0]);?>@my_web_site.org">E-MAIL THE <?php echo strtoupper($position); ?></a>
```

SHOW RECORD BASED ON DRAG SORT ORDER - Aug 3rd, 2010

Sometimes you may want to show the information from a record based on drag sort order. User rjbathgate did

Dave Edis says it's really simple.

He suggests that if you add the code indicated, where n = the position of the record that you want to start with. Don't forget that the first position is position "0" Zero, so if you want to start with the 3rd position, you'd enter 2 for the "n" value..

```
list($your_tableRecords, $your_tableMetaData) = getRecords(array(
    'Table name' => 'your_table',
    // Insert This Code
    'offset' => n,
    <!-- End of Insert-->
));
```

SHOWING A LIST VIEWER AND A DETAIL VIEWER ON THE SAME WEB PAGE - Aug

3rd, 2010

My client wanted to create a newsletter detail page that displayed their current newsletter and also displayed a list of links to past newsletters on the same page. The client did not want to have their visitors to have to navigate through a list page.

I knew that I could display the first record in a multi-record editor by leaving out any search modifiers at the end of the detail page link.

```
<a href="newsletterdetail.php">newsletter</a>
```

I also found that I removed

```
'limit' => '1',
```

from the call for that table at the head of my detail page, I could show the first record (the current newsletter) and at the same time insert a foreach loop on the page to display all of the past newsletters.

This may not be the most elegant approach, but it worked and may come in handy for some of you.

Here's the code that I used:

In the head:

```
list($your_newsletter_tableRecords, $your_newsletter_tableMetaData) = getRecords(array(
    'Table name' => 'your_newsletter_table',
    'where' => whereRecordNumberInUrl(1),
));
$your_newsletter_tableRecord = @$your_newsletter_tableRecords[0]; // get first record
```

In the body to display the current newsletter:

```
<?php echo $your_newsletter_tableRecord['your_field_1'] ?>
?php echo $your_newsletter_tableRecord['your_field_2'] ?>
?php echo $your_newsletter_tableRecord['your_field_3'] ?>
etc...
```

And for the past newsletters menu:

```
<?php foreach ($your_newsletter_tableRecords as $record): ?>
    <a href="<?php echo $record['_link'] ?>"><?php echo $record['date'] ?></a>
<?php endforeach ?>
```

PULLING MATCHING RECORD INFORMATION FROM MORE THAN ONE MULTI-RECORD EDITOR - Aug 3rd, 2010

The same client in the above entry upped the ante by then wanting to be able to create as many news articles as desired for any given issue of the newsletter. I had already created a multi-record editor, called "newsletters", for most of the information that appeared in their newsletters (mastheads, segment titles, meeting information etc.) That meant creating another multi-record editor for the "articles" and being able to have only the correct month's articles displayed on the detail page.

The first step was to create a simple Multi-Record editor called Newsletter Date with one text field for the Month and Year of their newsletters. These records would be used to populate a pull down list which assigned articles to their correct issue and to create the "past issues" list, and eliminated the chance of misspelling the entries.

The next step was to add a pull down list field to my existing multi-record editor "newsletters" that gets it's information from the Newsletter Date table, like this.

```
Field type: list
Display as: Pull-down
List Options: Get options from Database (advanced)
Section Table name : newsletter_date
Use this field for option values: month_and_year
Use this field for option labels: month_and_year
```

Then I created a multi record editor for the articles called "newsletter articles", with the same pull-down list field as above, plus any other fields required, like article headline, article contents, author.

After those editors were created, I created a few test records for an assortment of months, and entered some dummy information into them.

Then it was time to set up a detail page that would display the correct sets of records. After rummaging around in the forum, I found that inserting this where statement in the require once at the top of my page,

```
'where' => "newsletter_date = '{$newsletter_templateRecord['newsletter_date']}'"
```

I could restrict the article records displayed to only those that matched the "newsletter_date" field in the original "newsletters" editor.

So this is what the top of my page looked like.

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/my_path/cmsAdmin/lib/viewer_functions.php";

list($newsletterRecords, $newsletterMetaData) = getRecords(array(
    'tableName' => 'newsletter',
    'where' => whereRecordNumberInUrl(1),

));
$newsletterRecord = @$newsletterRecords[0]; // get first record

list($newsletter_articlesRecords, $newsletter_articlesMetaData) = getRecords(array(
    'Table name' => 'newsletter_articles',
    'where' => "newsletter_date = '{$newsletterRecord['newsletter_date']}'"

));
$newsletter_articlesRecord = @$newsletter_articlesRecords[0]; // get first record

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

The rest of the page was then pretty straight forward.

Since I was always displaying the first record in the "newsletter" table, where I wanted the information from the "newsletter" table to appear I used:

```
<?php echo $newsletterRecord['my_field_name'] ?>
```

Where I wanted the Articles to appear I used:

```
<?php foreach ($newsletter_articlesRecords as $record): ?>
<?php echo $record['article_headline'] ?><br />
<?php echo $record['article_contents'] ?><br />
<?php echo $record['author'] ?><br />
<?php endforeach; ?>
```

And the code for the list of past issues looked like this:

```
<?php foreach ($newsletterRecords as $record): ?>
<a href="<?php echo $record['_link'] ?>"><?php echo $record['newsletter_date'] ?>
</a>
<?php endforeach ?>
```

I could have set up other multi record editors, and as long as they followed the same pattern, they too would appear correctly.

Hope this approach comes in handy.

SHOWING IMAGES FROM ANOTHER TABLE ON YOUR PAGES - Aug 3rd, 2010

Aquaman had this interesting solution that he shared with us on the forum.

He had a series of banner images and he wanted to be able to specify which ones to display on a series of detail pages without the need to upload the images each time a page was added.

Here's how you can do the same:

First, create a section editor (called "My Images" for this recipe) with two fields to act as your image "library":

- Title
- Image (upload)

The upload field is set for a maximum of 1 upload and the extra info field labels are removed. I'd suggest setting this as a required field. You can set the other values and thumbnail sizes to suite your needs.

In the section where you want to show the images, (Which I called "My Pages" for this recipe), you'd use the following setup for a field (which I called Banner Image):

- Field Type: list
- Display As: pull-down
- Get options from database
- Section Table name: my_images
- Use this field for option values: num
- Use this field for option labels: title

Before the <head> of the detail page where you want's to show the images you'd used the following code:

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/hsphere/local/home/gkornblu/thecmsbcookbook.com/cmsAdmin/lib/viewer_functions.php";

list($my_pagesRecords, $my_pagesMetaData) = getRecords(array(
    'Table name' => 'my_pages',
    'where'      => whereRecordNumberInUrl(1),
    'limit'     => '1',
));
$my_pagesRecord = @$my_pagesRecords[0]; // get first record

// show error message if no matching record is found
if (!$my_pagesRecord) {
    header("HTTP/1.0 404 Not Found");
    print "Record not found!";
    exit;
}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Then in the body of the page where you want's to show the images you'd used the following code:

```
<?php
list($my_imagesRecords, $my_imagesMetaData) = getRecords(array(
    'Table name' => 'my_images',
    'where'      => "num='{$my_pagesRecord['banner_image']}'",
    'limit'     => '1',
));
$my_imagesRecord = @$my_imagesRecords[0]; // get first record
?>
```

And finally where the image is displayed (must be below the code above on your page)

```
<?php foreach ($my_imagesRecord['image'] as $upload): ?>
" height="<?php echo
$upload['thumbHeight'] ?>" alt="" />
<?php endforeach ?>
```

If for some reason you wanted to display the images from the "banner library" on a list page, the code before the <head> tag would change to:

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/your_path_to/cmsAdmin/lib/viewer_functions.php";

list($my_pagesRecords, $my_pagesMetaData) = getRecords(array(
    'Table name' => 'my_pages',
));

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

And you'd need to include the code from the detail page in your foreach loop, and change the:

```
where' => "num ='{ $my_pagesRecord['banner_image']}'",
```

to

```
'where' => "num ='{ $record['banner_image']}'",
```

Your code would then look like this:

```
<?php foreach ($my_pagesRecords as $record): ?>

<?php
list($my_imagesRecords, $my_imagesMetaData) = getRecords(array(
    'Table name' => 'my_images',
    'where' => "num ='{ $record['banner_image']}'",
    'limit' => '1',
));
$my_imagesRecord = @$my_imagesRecords[0]; // get first record
?>

<?php foreach ($my_imagesRecord['image'] as $upload): ?>
    " height="<?php echo
$upload['thumbHeight'] ?>" alt="" />
<?php endforeach ?> <hr/>
<?php endforeach ?>
```

FORMATTING TEXT TO AUTOMATICALLY FLOW EVENLY INTO MULTIPLE COLUMNS - Aug 3rd, 2010

One of the things that I've wanted to be able to do was to be able to make long blocks of text automatically flow evenly into multiple columns.

Here's one approach, that uses a combination of CSS and Javascript to handle the discrepancies in the current implementation of the "proposed" CSS3 multi column rules.

It's based on the examples shown in these articles:

<http://www.alistapart.com/articles/css3multicolumn/>

and

<http://www.cvwdesign.com/txp/article/360/>

So that I could follow the examples in the articles, I created:

So that I could follow the examples in the articles, I created:

- A .css style sheet called multi-column.css
- A Javascript file called css3-multi-column.js
- And a sample page called column-sample.php

You can download the CSS, Javascript and sample files from

<http://www.thecmsbcookbook.com/downloads/css3-multi-column.zip>

Every implementation will be slightly different so I'd recommend playing with the different possibilities listed in the article and in the proposed spec to get your page looking like you want it to.

Because of some conflict issues with the Javascript, you may get css conflicts if you use more than one set of column classes in the same external css file, EVEN IF YOU NAME THE CLASSES DIFFERENTLY.

So either put the column formatting in the head your web page or create a totally different copy of the external .css file for each pages that uses different parameters.

<http://www.w3.org/TR/css3-multicol>

Note that in this example, the Javascript is called only for IE, and the css style sheet contains -moz and -webkit- prefixes for Mozilla and Safari.

That said, Here's the pretty simple code that I used:

In the head section of my 2 column viewer (column sample.php), I called the javascript and CSS file with:

```
<link href="multi-column.css" rel="stylesheet" type="text/css" />
<!--[if IE]>
<script type="text/javascript" src="css3-multi-column.js"></script>
<![endif]-->
```

Then in the body, where I wanted to display the 2 column text, I enclosed it in a <div> tag. You can display multiple fields in the same <div> and they'll all flow into neat columns.

```
<div class="column" align="left">
<?php echo $yourRecord[your_field] ?><br /><br /><hr width="200">
</div>
```

Since first writing this, I've been able to use the technique to create automatic flow in some pretty complex viewers, so experiment and enjoy.

WHO STOLE MY ZEROS? - Aug 3rd, 2010

User Perchpole was concerned by how zeros were displayed on a viewer page. He said: "I'm using a record to perform a calculation that generates a simple product shipping cost. The calculation is the product weight multiplied by a fixed number. Something like this:

```
<?php echo $record[weight] * 5 ?>
```

Although this works fine in practice (and can be used in systems like Paypal, etc.) the figures don't look right.

For example, instead of £2.50 what you see is £2.5!

Instead of £7.00 all you get is £7.

Simple question: How do I rework the code to display the missing zeros?!

Dave Edis from Interactive tools came to the rescue. He answered:

"Try this:

```
<?php echo number_format($record[weight] * 5, 2) ?>
```

The ", 2" tells it to always show 2 decimals, even then they're 00. This will also add thousands separators. So one million will display like this: 1,000,000.00

So simple when you know how...

ADDING UP THE TOTAL VALUE OF ALL 'COST' FIELDS - Oct 13th, 2011

User s2smedia had a field called 'cost' and wanted to add up all of the 'cost' fields and give a TOTAL

Jason Sauchuk from Interactive Tools offered the following for CMSB Version 2.08 and above:

```
<?php
$totalCost = 0.00;
$listings = mysql_select('listings');

foreach ($listings as $listing) {
    $totalCost += floatval($listing['cost']);
}

?>
```

And this for previous CMSB versions:

```
<?php
$totalCost = 0.00;

$tableName = "listings";
$listings = mysql_query_fetch_all_assoc("SELECT * FROM `{$TABLE_PREFIX}$tableName`");

foreach ($listings as $listing) {
    $totalCost += floatval($listing['cost']);
}

?>
```

REMOVING DOLLAR SIGNS FROM NUMERICAL VALUES - Oct 26th, 2012

The easiest way to remove dollar signs from your data is not to allow them in the first place. But if you have to include them, Greg Thomas, a developer at Interactive Tools offered this advice.

If you want to strip the dollar sign and comma from a number you could use these functions together:

```
$value = preg_replace("/[\$,]/", "", $value);
$value = floatval($value);
```

The preg_replace function will strip out the dollar sign and any commas. The floatval function will convert the number from a string to a floating value, ensuring it will behave like a number if you want to use it for addition, multiplication, etc.

To convert the number back to a currency format I would use this:

```
echo '$ '.number_format($number, 2, ',', ',');
```

The number_format function will take a number and convert it into string format it. In this case it makes the number have two decimal places and adds a comma for every 3 digits.

CALCULATOR USING PHP - Jan 5th, 2012

There are many approaches to creating calculators using JavaScript and other approaches. Here's one that I put together for a Gicleé printing company that uses fairly simple PHP programming.

Dimensional values in the pull down lists are taken from a multi-record list of standard print dimensions using the getListOptions function where "dimensions" is the name of the table and "value2" is the name of the field holding the numerical value.

Pricing values are taken from a multi-record list of information about each of the papers that the company utilizes. Each record contains a field for manufacturer, media family, media weight and price.

The form displays pull down list of \$label values made up of the values of the record's fields: manufacturer, media family and media weight. The value of the record's price field is used for the option value.

The formula used is (standard length in inches x standard height in inches)/144) * price per square foot = \$value.

A reset form is included at the end.

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>" >
<select name = "price" >
<option value="">Choose A Paper Type From This List</option>
<?php foreach ($print_mediaRecords as $record): ?>
<?
php $label = @$record['manufacturer:label'] . " ". @$record['media_family'] . " ". @$record['media_weight'] . " lbs"
?>
<option value = "<?php echo(@$record['price']);?>" label="<?php echo($label) ?>">
<?php echo $label; ?></option>
<?php endforeach ?>
</select>
<br /> <br />
<span class="body-text-9">The dimensions listed are our standard printing dimensions. Your actual print dimensions can't be larger than the ones you choose.</span>
<br /> <br />
<select name="input1">
<option value="">Choose Your First Dimension (Length Or Width)</option>
<?php foreach (getListOptions('dimensions','value2') as $value => $label): ?>
<option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['value2']);?>>
<?php echo $label; ?></option>
<?php endforeach ?>
</select>
<br /> <br />
<select name="input2">
<option value="">Choose Your Second Dimension</option>
<?php foreach (getListOptions('dimensions','value2') as $value => $label): ?>
<option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['value2']);?>>
<?php echo $label; ?></option>
<?php endforeach ?>
</select>
<br /> <br />
<input type="submit" name="submit" value="Click To Calculate The Cost Of Your Print" >
</form>
<br />
<?PHP
```

```

//The calculation
$value = (((@$_POST['input1'] * @$_POST['input2'])/144) * @$_POST['price'] ;
?>
<span class="body-text-bold-9">Cost For Each Print:</span>
<span class="body-text-9">$
<?PHP
//Output result
echo number_format($value ,2) ?>
<br />
<span class=" body-text-bold-red-9">*</span> <span class="body-text-
9">Print Prices are approximate, if adjustments of
more than 10% are required, we will notify you prior to proceeding.</span>
<br /> <br />
<FORM ACTION="">
<INPUT TYPE="submit" VALUE="Reset Form and Calculate Another Print Cost">
</FORM>

```

USING PHP VARIABLES IN JAVASCRIPT - Jan 22nd, 2012

I had a hard coded Javascript calculator (validate.js) that looks something like the code below, and I needed to be able to pull the values for the calculation from a CMSB database called "Prices".

Here's where I started (relevant code only):

```

[code]
// JavaScript Document

if ($('#size1').is(':checked')) total = total + 349;
if ($('#size2').is(':checked')) total = total + 249;
if ($('#size3').is(':checked')) total = total + 299;
if ($('#size4').is(':checked')) total = total + 399;

[/code]

```

Here's what worked for me:

First, I created a single record editor called Pricing" with text fields for Size2, Size2, Size3 and Size 4. (I used descriptive text for the field labels and the size1,size2,size3 and size4 for the field names.)

After I entered in some sample prices, I changed my Javascript document validate.js to validate.js.php.

Then I put load viewer and load records code at the top of the Javascript file.

Then I added: <?php header("Content-type: text/css"); ?

> to keep the code rendering happy.

Right below that I defined variables for each of the field names.

```

IE:
<?php $size1= $pricingRecord[size1] ?>
<?php $size2= $pricingRecord[size2] ?>
<?php $size3= $pricingRecord[size3] ?>
<?php $size4= $pricingRecord[size4] ?>

```

Here's how the relevant code in my final Javascript document looked:

```

<?php

// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/your_server_path/', './', '../', '././', './././');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }}
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

// load records
list($pricingRecords, $pricingMetaData) = getRecords(array(
    'tableName' => 'pricing',
    'allowSearch' => '0',
    'limit' => '1',
));
$pricingRecord = @$pricingRecords[0]; // get first record

?>
<?php
header("Content-type: text/css");
?>

<?php $size1= $pricingRecord[size1] ?>
<?php $size2= $pricingRecord[size2] ?>
<?php $size3= $pricingRecord[size3] ?>
<?php $size4= $pricingRecord[size4] ?>

// JavaScript Document

if ($('#size1').is(':checked')) total = total + <?=$size1 ?>;
if ($('#size2').is(':checked')) total = total + <?=$size2 ?>;

```

```
if ($('#size2').is(':checked')) total = total + <?= $size2 ?>;
if ($('#size3').is(':checked')) total = total + <?= $size3 ?>;
if ($('#size4').is(':checked')) total = total + <?= $size4 ?>;
```

FORMATTING ROMAN NUMERALS AND TITLE CAPITALIZATIONS - Jul 9th, 2014

Formatting of Roman Numerals and title capitalization can be tricky, especially when you want most of the words in your title to be capitalized.

This RemoveShouting function will capitalize all roman numerals between 1 and 10 in a string (which I called \$instruction) and also force all the words in the \$lower_exception array to appear as lower case in that string. All other words in the string will have their first letter capitalized. The code is based on an example that I found in the PHP manual at:

<http://www.php.net/manual/en/function.ucfirst.php>

Just insert the function code in the body of your viewer before you need to call it, and outside of any foreach loops (you can only declare a function once on a viewer).

Here's the function code:

```
<?php function RemoveShouting($instruction)
{
    $lower_exceptions = array(
        "to" => "1", "a" => "1", "the" => "1", "and" => "1", "but" => "1", "or" => "1", "for" => "1", "nor" => "1", "of"
        => "1"
```

```

);

$higher_exceptions = array(
    "I" => "1", "II" => "1", "III" => "1", "IV" => "1",
    "V" => "1", "VI" => "1", "VII" => "1", "VIII" => "1",
    "IX" => "1", "X" => "1"
);

$words = explode(" ", $instruction);
$newwords = array();

foreach ($words as $word)
{
    $firstElement = ($word == reset($words));
    $lastElement = ($word == end($words));

    if (!$higher_exceptions[$word]) {$word = strtolower($word);}
    if ((!$lower_exceptions[$word]) || ($word == $firstElement) || ($word == $lastElement) ) {$word =
ucfirst($word);}
    array_push($newwords, $word);
}

return join(" ", $newwords);
}
?>

```

Then where you want to echo the re-formatted \$instruction variable, insert:

```
<?php echo RemoveShouting($instruction) ?>
```

You can add any words to the \$lower_exceptions array to format your titles, but the rules for capitalization vary between sources. Just pick one and be consistent. You can also add Roman Numerals to the \$higher_exceptions array.

If you need to implement RemoveShouting more than once, rename a copy of the function and insert your new variable.

WORKING WITH DATES

DISPLAYING A LIST OF MONTHS WITH STRTOTIME - CAVEAT - Nov 19th, 2012

When a user had issues with displaying months in a pull down menu in a form using:

```

<select name="month">
    <?php foreach(range(1,12) as $month): ?>
        <option value="<?php echo $month;?>" <?
php selectedIf($month,@$_REQUEST['month']);?>"><?php echo
date("F",strtotime("0000-$month"));?></option>
    <?php endforeach ?>
</select>

```

Greg Thomas, a programmer at Interactive Tools had this observation:

"In certain versions of PHP you can't leave any parts of a date string as 0 when using the strtotime function. Try changing the code to look something like this:

```

<select name="month">
    <?php foreach(range(1,12) as $month): ?>
        <option value="<?php echo $month;?>" <?
php selectedIf($month,@$_REQUEST['month']);?>"><?php echo
date("F",strtotime("1985-$month-01 00:00:00"));?></option>
    <?php endforeach ?>
</select>"

```

DISPLAYING ONLY TODAY'S RECORD - Aug 3rd, 2010

User jtedescojam set up a number of records in a multi record editor and wanted to be able to display a record's information only when the date in it's date field matched the actual date.

Dave Edis from Interactive Tools had a simple suggestion.

In the get records call for that editor at the top of the page add this code:

```
'where' => "YEAR(date) = YEAR(NOW()) AND DAYOFYEAR(date) = DAYOFYEAR(NOW())",
```

he noted that there's a lot more information about date functions at:

<http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html>

Thanks Dave.

COMPARING DATES - Apr 1st, 2018

COMPARING DATES

When combined with the "if" statement, this extremely useful concept can be used to show or hide fields based on a particular date, automatically archive records after a specified time period and perform many other functions.

Here's the basic idea of how to compare a date field to today's date on a list page. This example uses a date field called "opening_reception" in a multi record editor called "exhibitions". The goal was to automatically hide the opening reception date after the date had passed.

First you'll need a "foreach" statement to display each record.

```
<?php foreach ($exhibitionsRecords as $record): ?>
```

Since dates are normally expressed as months, days and years are hard to compare mathematically, the dates are converted to the number of seconds since the Unix Epoch (January 1 1970 00:00:00 GMT)

Note that dates are referenced to local time as set in the CMSB "General Settings" information and not server time.

Dave suggested to first define a few variables and to use simple, descriptive names so that the process is easier to follow. Then set some rules for the comparisons. You can use mathematical operators like <, >, <= or >= between values to compare them in different ways.

```
<?php
$eventUnixTime = strtotime( $record['reception_date'] ); // seconds since 1970
$eventDateString = date("l, F jS", $eventUnixTime); // example format: Monday, June 1st
$currentUnixTime = time();
$currentDateString = date("l, F jS", $currentUnixTime);

$isEventToday = ($eventDateString == $currentDateString); // first comparison
$isEventOver = !$isEventToday && ($eventUnixTime < $currentUnixTime); // second comparison
$isFutureEvent = !$isEventOver && !$isEventToday; // third comparison
?>
```

Then it becomes a simple matter of comparing the variables

```
<?php if ($isFutureEvent): ?>
Opening Reception: <?php echo $eventDateString ?>

<?php elseif ($isEventToday): ?>
The Opening reception is today. Don't miss it!!

<?php else: ?>
Sorry, you missed the Opening Reception.

<?php endif; ?>
```

and don't forget the endforeach statement to close your loop.

```
<?php endforeach ?>
```

Or, Let's say you want to group a set of meetings by future and past dates

You could use:

```
<?php foreach ($general_meetingsRecords as $record): ?>
<?php
$eventUnixTime = strtotime( $record['date'] ); // seconds since 1970
$eventDateString = date("l, F jS", $eventUnixTime); // example format: Monday, June 1st
$currentUnixTime = time();
$currentDateString = date("l, F jS", $currentUnixTime);

$isNewEventToday = ($eventDateString == $currentDateString);
$isNewEventOver = !$isNewEventToday && ($eventUnixTime < $currentUnixTime);
$isNewFutureEvent = !$isNewEventOver && !$isNewEventToday;
?>

<?php if ($isNewEventToday): ?> <p>

<div align="center" class="heading-text-bold">TONIGHT'S MEETING:</div>
</p>
<br /> <span class="body-text-bold"><?
php echo date("D, M jS, Y g:i a", strtotime($record['date'])) ?>
</span>
<br />
<div align="left" class="body-text"><?php echo $record['content'] ?></div>
<hr align="left" color="#A29DB2" width="100" /><br />
<?php endif; ?> <?php endforeach ?>
</div>
<p>

<div align="center" class="heading-text-bold">UPCOMING MEETINGS:</div>
</p>
<?php foreach ($general_meetingsRecords as $record): ?>

<?php
$eventUnixTime = strtotime( $record['date'] ); // seconds since 1970
$eventDateString = date("l, F jS", $eventUnixTime); // example format: Monday, June 1st
$currentUnixTime = time();
$currentDateString = date("l, F jS", $currentUnixTime);

$isNewEventToday = ($eventDateString == $currentDateString);
$isNewEventOver = !$isNewEventToday && ($eventUnixTime < $currentUnixTime);
$isNewFutureEvent = !$isNewEventOver && !$isNewEventToday;
?>

<?php if ($isNewFutureEvent): ?>
<br />
<span class="body-text-bold"><?
php echo date("D, M jS, Y g:i a", strtotime($record['date'])) ?></span> <br />

<div align="left" class="body-text"> <?php echo $record['content'] ?></div>
<hr align="left" color="#A29DB2" width="100"
/>
<?php endif; ?> <?php endforeach ?>
<br />
<p><div align="center" class="heading-text-bold">PAST MEETINGS:</div></p>

<?php foreach ($general_meetingsRecords as $record): ?>

<?php
$eventUnixTime = strtotime( $record['date'] ); // seconds since 1970
$eventDateString = date("l, F jS", $eventUnixTime); // example format: Monday, June 1st
$currentUnixTime = time();
$currentDateString = date("l, F jS", $currentUnixTime);

$isNewEventToday = ($eventDateString == $currentDateString);
$isNewEventOver = !$isNewEventToday && ($eventUnixTime < $currentUnixTime);
$isNewFutureEvent = !$isNewEventOver && !$isNewEventToday;
?>

<?php if ($isNewEventOver): ?>
<br />
<span class="body-text-bold"><?
php echo date("D, M jS, Y g:i a", strtotime($record['date'])) ?></span> <br />

<div align="left" class="body-text"><?php echo $record['content'] ?></div>
<hr align="left" color="#A29DB2" width="100"
/>
<?php endif; ?> <?php endforeach ?>
</div>
```

Note that you have to redefine the variables for each "foreach" loop

If you want to compare other field values, just define more variables and compare those as well.

If you want to test for the values of the variable that you're using. Try something like this in your foreach loop:

```
<ul>
  <li>Event date = <?php echo $eventDateString ?></li>
  <li>Current date = <?php echo $currentDateString ?></li>
  <li>$isEventToday = <?php echo $isEventToday ? "true" : "false" ?></li>
  <li>$isEventOver = <?php echo $isEventOver ? "true" : "false" ?></li>
  <li>$isFutureEvent = <?php echo $isFutureEvent ? "true" : "false" ?></li>
  <li>$CurrentUnixTime extended = <?
php echo date("l jS \of F Y h:i:s A", $currentUnixTime); ?></li>
</ul>
```

ADDING DAYS WEEKS OR MONTHS TO DATES - Dec 5th, 2012

When I was creating the on-line version of this cookbook, I wanted to be able to show NEW or REVISED recipes for a period of 30 days.

Fortunately, using PHP it's pretty easy to add (or subtract) days, weeks, or months, to a date. The trick is that you have to convert your dates into UnixTime (seconds since 1970) before you can successfully add or subtract the values.

Here's how I accomplished my goal.

First I created 2 checkboxes fields in my editor. One for New and one for REVISED.

Then I defined the date and time variables that I needed to compare. (Current UnixTime is shown for clarity of this recipe) and placed the code within my foreach loop.

```
<?PHP
$newUnixTime = strtotime($record['createdDate']); // seconds since 1970 to createdDate
$updateUnixTime = strtotime($record['updatedDate']); // seconds since 1970 to updateDate
$currentUnixTime = strtotime(date("Y-m-d"));
$ageUnixTime = $updateUnixTime - $currentUnixTime; // seconds since "00-00-00" to
```

```

$pastUnixTime = strtotime(date("Y-m-d", strtotime("-30 day")));
$newdifference = $newUnixTime - $pastUnixTime;
$updatedifference = $updateUnixTime - $pastUnixTime;
?>

```

Once the variables were determined, it was a matter of comparing them and outputting the correct information based on the result.

```

<!-- Check For New Or Revised Date -->
<?php if ($record['new'] == 1 && $newdifference >= 0): ?><span class="Medium-Text-Bold-Red">NEW</span><?php endif ?>
<?php if ($record['revised'] == 1 && $updatedifference >= 0): ?>
<span class="Medium-Text-Bold-Red">REVISED</span><span class="Medium-Text-Bold"> - </span><?php endif ?>
<!-- End Date Check -->

```

Some other examples of adding to and subtracting from dates.

```

$add_a_week = strtotime(date("Y-m-d", strtotime("+1 week")));
$subtract_2_months = strtotime(date("Y-m-d", strtotime("-2 month")));

```

LISTING RECORDS WITHIN A DATE RANGE - Aug 3rd, 2010

Here's one that came in extremely handy.

I had an organization that had events that spanned a period of time and a special reception somewhere within those dates.

My client wanted to list all of the events on their list page, but wanted to make a special list at the top of the page for those events with receptions that were going to occur within the upcoming week.

Thanks to Dave Edis of Interactive Tools (again) who freely admits that date math makes his head hurt, we came up with the following.

First I created a date field in my 'date_compare_test' multi record editor called 'reception_date'.

Then came the task of creating a list viewer that would list only the events I wanted in the appropriate groups.

Dave came up with the idea of putting 2 viewers on my list viewer page and using 2 different 'where' statements in the get records calls to separate the content that could show in each area.

Here's the 'where' statement that we used to show reception_dates for the upcoming week, as you suggested.

```
'where' => '(NOW() + INTERVAL 7 DAY) >= reception_date AND reception_date >= TIMESTAMP(CURDATE(), "00:00:00")',
```

And here's the 'where' statement that we used to show all the other events (and leave out those already listed)

```
'where' => 'reception_date <= TIMESTAMP(NOW()) OR reception_date >= TIMESTAMP(NOW()+ INTERVAL 7 DAY)',
```

So here's how the complete viewer code looked.

```

<!-- Start of first list viewer code-->
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/hsphere/local/home/apbcweb/artistsofpalmbeachcounty.org/cmsAdmin/lib/viewer_functions.php";

list($date_compare_testRecords, $date_compare_testMetaData) = getRecords(array(
'Table name' => 'date_compare_test',
'where' => '(NOW() + INTERVAL 7 DAY) >= reception_date AND reception_date >= TIMESTAMP(CURDATE(), "00:00:00"',

```

```

));
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<hr />

EVENTS WITH RECEPTIONS TODAY THROUGH 7 DAYS IN THE FUTURE<br />
<br />
<?php foreach ($date_compare_testRecords as $record): ?>
Title: <?php echo $record['title'] ?><br />
The Reception Date is <?
php echo date("D, M jS, Y g:i a", strtotime($record['reception_date'])) ?> <br />
<?php endforeach ?>
<hr />

<!-- Start of second list viewer code-->

<?php

list($date_compare_testRecords, $date_compare_testMetaData) = getRecords(array(
'Table name' => 'date_compare_test',

'where' => 'reception_date < TIMESTAMP(CURDATE(), "00:00:00") OR reception_date > TIMESTAMP(NOW()+ INTERVAL 7 DAY)',
));
?>

EVENTS WITH RECEPTIONS BEFORE TODAY OR MORE THAN 7 DAYS IN THE FUTURE<br />
<br />
<?php foreach ($date_compare_testRecords as $record): ?>

Title: <?php echo $record['title'] ?><br />
The Reception Date is <?
php echo date("D, M jS, Y g:i a", strtotime($record['reception_date'])) ?> <br />
<?php endforeach ?>

</body>
</html>

```

Dave explained that for date queries,

"The first step is to figure out if you need to do it in MySQL or PHP. If you are able to do it in MySQL it's often simpler.

You can find a list of MySQL date/time functions here:

<http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html>

And here's some examples:

NOW() - Gives you the current date and time: 2010-03-01 09:41:50
(NOW() + INTERVAL 7 DAY) - Get date/time 7 days from now: 2010-02-22 09:45:25
CURDATE() - Gives you the current date only: 2010-03-01
TIMESTAMP() - Format value as date/time, or if two values, add them together
TIMESTAMP(CURDATE()) - Get date/time at beginning of day: 2010-03-01 00:00:00
TIMESTAMP(CURDATE(), "23:59:59") - Get date/time at end of day 2010-03-01 23:59:59

And you can test all these with the MySQL Console Plugin by entering SELECT followed by the function. So: SELECT NOW()
in the Mysql Console returns: 2010-03-01 09:41:50

So the first step is to figure out the values you want to compare. My guess is you want:

The date 7 days from now: (NOW() + INTERVAL 7 DAY)
The reception date: reception_date
The date at the end of today: TIMESTAMP(CURDATE(), "23:59:59")

If you write it out in English first it's way easier:
- If the reception_date is 7 days or less from now
- AND the reception_date hasn't already passed

I like to arrange my code so it reads like a time range with the test date in the middle like this:

```
start_date >= test_date AND test_date >= end_date
```

So that would be:

```
(NOW() + INTERVAL 7 DAY) >= reception_date AND reception_date >= TIMESTAMP(CURDATE(), "00:00:00")
```

Now I understand why Dave says date math makes his head hurt.

USING THE NEW "CUSTOM DATE OR STRTOTIME VALUE" DATE FUNCTION - Aug 3rd, 2010

Since we can now specify a custom date or strtotime value in date fields, (V2.01) it's easy to specify a `publishDate` or `removeDate` that is say 4 weeks from the current date. It's just a bit counter intuitive.

When you first choose "specify custom date (or strtotime value) below" from the default value pull-down menu the box is filled with something like "2010-01-01 00:00:00" Which you can manually change to choose a custom date and time.

You can modify this date string to choose a custom date and time in the future.

Or you can add a simple relative expression like "+4 weeks" after the modified date string (without the double quotes) and the date will be set to your modified date + 4 weeks.

To specify a value relative to the current date and time in the field (this is the counterintuitive part). Just delete the 2010-01-01 00:00:00 and replace it with the "+4 weeks" expression (without the double quotes).

You can use any number that you want to and change "weeks" to "years", "days", "hours", "minutes" or even "seconds" to meet your needs. You should see the new date reflected in the "preview" just under the field.

This only works on newly created records so of course it won't update existing records which must be revised manually.

Or Dave Edis from Interactive Tools suggested:

If you wanted to update the old records you could do that with this command in the MySQL Console (a free plugin):

```
UPDATE cms_your_table_name SET removeDate = NOW() + INTERVAL 4 week WHERE removeDate = "0000-00-00 00:00:00"
```

Your table name replaces the text "your_table_name" after the "cms_".

As always, when you're making these kinds of changes to your on-line database, don't forget to back up your original data before you begin.

SHOW DATES ON YOUR PAGE - Jan 15th, 2015

Sometimes you'll want to show the current year on your page, like when showing copyright information. It's easy to show the current year by adding

```
<?php echo date("Y") ?>
```

to your page.

You can use any of the date modifiers to show date information as well, so:

```
<?php echo date("D, M jS, Y") ?>
```

Would show the Day of the week, and the Month, Day and Year.

You can format a date string to display as Day Date, Month Year on a detail page by using:

```
<?php echo date("D jS, M Y ", strtotime($your_tableRecord[your_date_field])) ?>
```

Or, on a list page:

```
<?php echo date("D jS, M Y ", strtotime($record[your_date_field])) ?>
```

If you thought, as I did, that you couldn't format the displaying of the special fields `createdDate` and `updatedDate`, you'd be wrong.

You can format these the same as any other date field.

NOTE: The commas in the string determine where commas show in your display.

This is the complete list of Date formatting modifiers

A = Uppercase 12hr time syntax - Example: AM, PM
a = Lowercase 12hr time syntax - Example: am, pm
B = Swatch Internet time - Example: 000 through 999
c = ISO 8601 date (added in PHP 5) - Example: 2004-02-12T15:19:21+00:00
D = short textual representation of the Day - Example: Fri
d = Day of the month, 2 digits with leading zeros - Example: 03
e = Timezone identifier (added in PHP 5.1.0) - Example: UTC, GMT, Atlantic/Azores
F = Full Textual representation of the month - Example: January
G = Hours in 24 hour format without leading zeros - Example: 17
g = Hours in 12 hour format without leading zeros - Example: 7
H = Hours in 24 hour format with leading zeros - Example: 17
h = Hours in 12 hour format with leading zeros - Example: 07
I (capital i) = Whether or not the date is in daylight saving time - Example: 1 if Daylight Saving Time, 0 otherwise
i = Minutes with leading zeros - Example: 00 through 59
j = Day of the month without leading zeros - Example: 10, 25
L = Whether it's a leap year - Example: 1 if it is a leap year, 0 otherwise
l (lowercase 'L') = full textual representation of the day of the week e.g. Monday, Tuesday
M = A short textual representation of a month, three letters - Example: Jan, Feb
m = Numeric representation of a month, with leading zeros - Example: 01
N = ISO-8601 numeric representation of the day of the week (added in PHP 5.1.0) - Example: 1 (for Monday) through 7 (for Sunday)
n = Numeric representation of a month, without leading zeros - Example: 1 through 12
O = Difference to Greenwich time (GMT) in hours - Example: +0400
o = ISO-8601 year number. This has the same value as Y, except that if the ISO week number (W) belongs to the previous or next year, that year is used instead. (added in PHP 5.1.0) e.g. 2010
P = Difference to Greenwich time (GMT) with colon between hours and minutes (added in PHP 5.1.3) - Example: +02:00
r = RFC 2822 formatted date - Example: Thu, 21 Dec 2000 16:01:07 +0200
S = English ordinal suffix for the day of month - Example: st, nd, rd, or th.
s = Seconds with leading zeros - Example: 28
T = Timezone abbreviation - Example: GMT, PST, EST, etc
t = Number of days in the given month - Example: 28, 31, etc.
U = Seconds since the UNIX epoch (January 1 1970 00:00:00 GMT) - Example: 1041604168
u = Microseconds (added in PHP 5.2.2) - Example: 654321
W = ISO-8601 week number of year, weeks starting on Monday (added in PHP 4.1.0) - Example: 42 (the 42nd week in the year)
w = numeric representation of the day - Example: 0 (for Sunday) through 6 (for Saturday)
Y = 4 digit year - Example: 2003
y = 2 digit year - Example: 03
Z = Timezone offset in seconds. The offset for time zones west of UTC is always negative, and for those east of UTC is always positive. - Example: 43200 through 50400
z = Day of the year (starting from 0) - Example: 0 through 365

And here are some Date String examples and their output

```
date("F j, Y, g:i a"); - March 10, 2010, 5:16 pm  
date("m.d.y"); - 03.10.10  
date("j, n, Y") - 10, 3, 2010  
date("D M j G:i:s T Y"); - Sat Mar 10 15:16:08 PST 2010  
date("D M jS, Y"); - Sat Mar 10th, 2010  
date("H:i:s"); - 17:16:17  
date("\i\t \i\s \t'h'e jS \d'ay."); - It is the 10th day.
```

*** According to Dave Edis of Interactive Tools, in V2.01+ if you have a blank date field, the date stored in your database is actually 0000-00-00 00:00:00, which is the default format MySQL stores dates in.

So, if you try to display the date, the technical reason you get 1969 or 1970 is because the way many servers record time is as epoch time or "seconds since midnight 1970 GMT" so what `strtotime()` does is convert your date to epoch time

time is as epoch time of seconds since midnight 1970 GMT, so what ultimately does to convert your date to epoch time, but since your date is zeroed out you get 0 seconds since 1970, which the date() function interprets as 1970, and then adjusts for your timezone (GMT -8 hours?)

If you try to test for a blank date with a simple:

```
<?php if ($your_tableRecord['your_date_field']): ?>
```

The test won't work as you expect, since there is actually a date value stored in your database, (0000-00-00 00:00:00)

Until this glitch is fixed, and you update to the "fixed" version, you'll have to format your test like this:

```
<?php if ($your_tableRecord['your_date_field'] && $your_tableRecord['your_date_field'] != '0000-00-00 00:00:00'): ?>
```

You can learn more about date formatting at:

<http://www.php.net/date>

RELATIVE DATES - FACEBOOK STYLE - May 8th, 2011

Chris Waddell from Interactive Tools just wrote a function that turns a date into a more human-readable format such as "37 seconds ago", "about an hour ago", "Wednesday at 6:07pm", or "February 17, 2007", depending on how far in the past the date is. (It doesn't handle future-dates at all.)

This code can go anywhere before it's first used:

```
<?php
function pretty_relative_time($time) {
    if ($time !== intval($time)) { $time = strtotime($time); }
    $d = time() - $time;
    if ($time < strtotime(date("Y-m-d 00:00:00")) - 60*60*24*3) {
        $format = 'F j';
        if (date("Y") !== date("Y", $time)) {
            $format .= ", Y";
        }
        return date($format, $time);
    }
    if ($d >= 60*60*24) {
        $day = 'Yesterday';
        if (date("l", time() - 60*60*24) !== date("l", $time)) { $day = date("l", $time); }
        return $day . " at " . date('g:ia', $time);
    }
    if ($d >= 60*60*2) { return intval($d / (60*60)) . " hours ago"; }
    if ($d >= 60*60) { return "about an hour ago"; }
    if ($d >= 60*2) { return intval($d / 60) . " minutes ago"; }
    if ($d >= 60) { return "about a minute ago"; }
    if ($d >= 2) { return intval($d) . " seconds ago"; }
    return "a few seconds ago";
}
?>
```

And here's how to call it in the body of your page:

```
<?php echo pretty_relative_date($record['createdDate']); ?>
```

SHOWING THE DIFFERENCE BETWEEN 2 DATES IN DAYS, HOURS, AND MINUTES - May 15th, 2015

Here's an interesting idea that was posted by CMSB user mifjtech.

```
<?php
$openDate = strtotime($record['createdDate']);
$closeDate = strtotime($record['closedDate']);
$dateDiff = ($closeDate - $openDate);

$d = ($dateDiff/(60*60*24))%365;
$h = ($dateDiff/(60*60))%24;
$m = ($dateDiff/60)%60;

echo $d." days\n";
echo $h." hours\n";
echo $m." minutes\n";
?>
```

VIEWERS - SEARCHING/SORTING/COMPARING

SETTING UP BASIC SEARCHES - Sep 4th, 2022

Here's some basic code that you can use to set up a simple search box on the list page of a multi-record editor to return only those records that match your search criteria. There's also a cancel search form. Style the forms to match your needs.

***If 'allowSearch' => is in your load records call at the top of your viewer, make sure that its set to true ***

```
<form action="" type="post">
  <input type="text" name="your_field_name_keyword" id="search" placeholder="Enter A Name To Search For - Partial Names OK" value="<?php echo htmlspecialchars(@$_REQUEST['your_field_name_keyword']); ?>" size="60" />
  <input type="submit" value="Start Your Search" />
</form>

<form type="post" action="" >
  <input type="submit" value="Cancel Search Filters - Show All Records">
</form>
```

Notice the "name" entry on the sample form above. It starts with the field in which you want to preform the search (your_field_name) and is followed by a modifier that determines what type of search is to be performed (_your_criteria).

CMSB is set up to allow you to search by many criteria by changing / your_

Simple is set up to allow you to search by many criteria by changing (your criteria) in the "name" entry to one of these:

- _match - an exact match
- _keyword - will look for specific words or partial words
- _prefix - starts with keyword (or letter)
- _query - allows google-style query searches such as: +dog - cat "multi word phrase". Only records matching EVERY word or quoted phrase are returned. Words or phrases that start with - mean "must not match". The + means "must match" Its optional and not required.
- _fieldname_empty Matches blank fields. Matches fields that are blank ("").
- _min - A minimum value for numeric searches
- _max - A maximum value for numeric searches
- _year year number for date searches
- _month - month number for date searches
- _day - Day of month for date searches

So, let's say you've set up a field called "fruit" that can contain one or more keywords, like Apple, Banana, Pear, Orange.

Here's the basic code that you would use to search for one of those.

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
  <input type="text" name="fruit_keyword" value="">
  <input type="submit" name="submit" value="Search">
</form>
```

If the visitor entered Apple, Banana, Pear, or Orange, only those records that contained the keyword would be listed.

You could also change the form to include a drop down menu of choices instead of the text box like this:

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">

  <select name="fruit_keyword">
    <option value="">Please Choose a Fruit</option>
    <option value="Apple">Apple</option>
    <option value="Banana">Banana</option>
    <option value="Pear">Pear</option>
    <option value="Orange">Orange</option>
  </select>

  <input type="submit" name="submit" value="Search">
</form>
```

CANCELLING A SEARCH - Aug 27th, 2020

To cancel a search use this code:

```
<form method="post" action="" >
<INPUT TYPE="submit" VALUE="Cancel Search">
</form>
```

BASIC SEARCH FOR A KEYWORD IN MULTIPLE FIELDS IN A MULTI-RECORD EDITOR - Oct 2nd, 2022

At the top of your page:

```
<?php
// load records from 'your_table'
list($your_tableRecords, $your_tableMetaData) = getRecords(array(
    'tableName' => 'your_table',
    'loadUploads' => true,
    'allowSearch' => true,
    'where' => " field1 LIKE '%".mysql_escape($query)."% ' OR field2 LIKE '%".mysql_escape($query)."% ' OR field3 LIKE '%".mysql_escape($query)."% ' " // Add as many fields as you need above, separating each by an OR
));

$query = @$_REQUEST['query']; // this will hold the keyword
?>
```

And in the Body

```
<form action="search.php" method="post">
    <input type="text" name="query">
    <input type="submit" name="submit" value="Search">
</form>
```

You might also want to include a cancel search form:

```
<form method="post" action="" >
<INPUT TYPE="submit" VALUE="Cancel Search - Start a New Search">
</form>
```

MORE COMPLEX SEARCH FOR A KEYWORD IN MULTIPLE FIELDS IN A MULTI-RECORD EDITOR - Oct 2nd, 2022

Searching Multiple fields at once

I needed to be able to search 3 category fields (text fields) at once, all of which got their option values and option names from a master catalog of categories. There were 3 fields because any of three books could be in more than one category. They could be "Art" books that also fit the "animals" category. To complicate the issue, my client wanted all of the Genre Categories to show in the pull down for admins but only a small subset of those would be shown to the public (non-admins).

Step 1 was to set up a master "Genre" catalog multi record editor. This editor was called "Genre Categories" and it had only 2 fields. A text field called 'Genre' and a checkbox called "Include In public search". (If I wanted to show a value to the public, then that box got checked). You should create 4 or 5 records for testing and check some to be in the public list.

Step 2 was to set up the multi record main books editor, called "Books". This editor had many fields, but the 4 that are needed for the multi field search are list fields called, " Genre", Genre 2, Genre 3 and "Public Genre". In addition, I'd add a "title" field for testing.

All list fields get their list options from "MYSQL query (advanced)"

The query for "Genre", "Genre 2", and "Genre 3" are:

```
SELECT num, genre_names
FROM `<?php echo $TABLE_PREFIX ?>genre_categories`
ORDER BY genre_names ASC
```

And the query for "Public Genre" is:

```
SELECT num, genre_names
FROM `<?php echo $TABLE_PREFIX ?>genre_categories`
WHERE include_in_public_search = 1
ORDER BY genre_names ASC
```

Step 3 was to populate the books editor with some records and assign different values to Genre 1, Genre 2 and Genre 3 for testing.

Now for the list page viewer.

At the top of your page you'll need to load your viewer library and load records from the books editor. Make sure that your load records call has "allow search" set to true, like this

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
```

```

<?php header( Content-type: text/html, charset=utf-8 ); //
<?php
/* STEP 1: LOAD RECORDS - Copy this PHP code block near the TOP of your page */

// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = [ '..', '..', '..', '..']; // add if needed: '/your/server/public_html/path/'
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }}
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

// load records from 'books'
list($booksRecords, $booksMetaData) = getRecords(array(
    'tableName' => 'books',
    'loadUploads' => true,
    'allowSearch' => true,
));
?>

```

In the body of your page insert your search forms:

```

<form method="post" action="search.php">
    <input type="hidden" name="save" value="1" />
<table align="left" width="90%" border="0" cellspacing="0" cellpadding="2">
    <tr>
        <td colspan="2">&nbsp;</td>
    </tr>
    <tr>
        <td align="right" class="text_font"><b>Book Search 1:</b></td>
        <td align="left" valign="bottom" colspan="2">
<select name = "genre, genre_2, genre_3_match[]" width="300"
class="text_font" style="width: 300px; max-height: 50px; " multiple>
    <?php foreach (getListOptions('books', 'genre') as $value => $label43): ?>
        <option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['genre, genre_2,
genre_3_match[]]);?>> <?php echo $label43; ?></option>
    <?php endforeach ?>
    </select></td>
</tr>
<tr>
        <td align="right" class="text_font"><b>Book Search 2:</b></td>
        <td align="left" valign="bottom" colspan="2">
<select name = "genre, genre_2, genre_3_match[]" width="300"
class="text_font" style="width: 300px; max-height: 50px; " multiple>
    <?
php foreach (getListOptions('books', 'public_genre') as $value => $label43): ?>
        <option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['genre, genre_2,
genre_3_match[]]);?>> <?php echo $label43; ?></option>
    <?php endforeach ?>
    </select></td>
</tr>
<tr>
        <td align="right" class="text_font"><b>&nbsp;</b></td>
        <td align="left" valign="bottom" colspan="2">
<input type="submit" value="Submit Search Filters" ></td>
    </tr>
</table>
</form>
<table align="left" width="90%" border="0" cellspacing="0" cellpadding="2">
    <tr>
        <!--<td>&nbsp;</td-->
        <td align="center" colspan="2"><form method="post" action="" >
            <INPUT TYPE="submit" VALUE="Cancel Search Filters - Start Another Search">
        </form></td>
    </tr>
</table>

```

Below that you'll show the results of your search starting with a telltale that shows the number of records returned that match your search criteria:

```

<!-- BEGIN RESULTS DISPLAY-->
<table align="left" width="90%" border="0" cellspacing="0" cellpadding="2">
<tr>
    <td colspan="2">
        <?php if(@$_REQUEST['save']):?>
        <?php $count = 0 ?>
        <?php foreach ($booksRecords as $record): ?>
        <?php $count++ ?>

```

```

<?php endforeach?>

    There are <?php echo $count; ?> records that match your search criteria
<?php endif ?></td>
</tr>

```

Followed by:

```

<tr>
    <td span class="text_font" colspan="2">
<hr align="left" style="height:2px; width=90%" />
<?php foreach ($booksRecords as $record): ?>

<b>Title:</b> <?php echo htmlentities($record['title']) ?>

    <?php if($record['genre']):?>
    <b>Genre 1:</b> <?php echo $record['genre:label'] ?>
    <?php endif ?>

    <?php if($record['genre_2']):?>
    <b>Genre 2:</b> <?php echo $record['genre_2:label'] ?>
    <?php endif ?>

    <?php if($record['genre_3']):?>
    <b>Genre 3:</b> <?php echo $record['genre_3:label'] ?>
    <?php endif ?>
<?php // Add any additional fields here to show them in the results ?>
<hr align="left" style="height:2px;" width="90%"/>
<?php endforeach ?>

```

And that's it. After you've tested your code, you can add <?php \$CMS_USER = getCurrentUserFromCMS(); ?> to see if the current user is an admin, and surround the "Book Search 1" code with <?php if (@\$CURRENT_USER['isAdmin']): ?> and <?php endif ?>.

You can use that code to restrict the displaying of any fields you like to admins only, and style the page any way that works for you.

SEARCH FORM THAT GETS LABEL VALUES FROM A LIST FIELD - Aug 17th, 2011

Instead of manually entering your search criteria, you can use this code to retrieve the label values from a list field and automatically insert them into your search form. Possible values for your criteria are listed below.

```

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<select name = "your_field_your-criteria" >
<option value="">Please Choose a Category</option>
<?php foreach (getListOptions('your_table ', 'your_field') as $value => $label): ?>

<option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['your_field']);?>>

<?php echo $label; ?></option>

<?php endforeach ?>

</select>

<input type="submit" name="submit" value="Search" >
</form>

```

Search Criteria (your-criteria)

- _match - an exact match
- _keyword - will look for specific words
- _prefix - starts with keyword (or letter)
- _query - allows google-style query searches such as: +dog - cat "multi word phrase". Only records matching EVERY word or quoted phrase are returned. Words or phrases that start with - mean "must not match". The + is optional and not required.
- _fieldname_empty Matches blank fields Matches fields that are blank (""). Example: email_empty=1
- _min - A minimum value for numeric searches
- _max - A maximum value for numeric searches
- _year year number for date searches


```

function wordCount($textOrHtml) {
    $text = strip_tags($textOrHtml, "<b></b><i></i>");
    $words = preg_split("/\s+/", $text);

    return count($words);
}
?>
<table width="100%" align="center" border="0" cellpadding="10">
    <tr>
        <td><?php foreach ($categorygroup as $group): ?>
            <div class=" heading-text-12"><?
                php foreach ($mobility_aid_records as $record): ?><?php if
                ($record[mobility_aid_category:label] == $group) : ?> <?
                php echo strtoupper($group); ?><?php break ?><?php endif;
                ?><?php endforeach ?>
            </div>
            <?php foreach ($mobility_aid_records as $record): ?>
            <?php if ($record[mobility_aid_category:label] == $group) : ?>
            <?php $name = htmlspecialchars($record[name]); ?>
            <table border="0">
                <tr>
                    <td width="150" height="150" align="center" valign="middle"><?
                    php foreach ($record[image] as $index => $upload):
                    ?>
                        <a href="<?php echo $record[_link]; ?>">" height="<?php echo $upload[thumbHeight] ?>" alt="" />
                        </a>
                        <?php endforeach ?>
                    </td>
                    <td>
                        <a href="<?php echo $record[_link]; ?>"><span class="heading-text-12">
                    <b><?php echo
                    strtoupper($name); ?></b></span></a><br /> <br />
                        <div align="left" class=" body-text-9"><?
                        php echo maxWords($record[description], 25); ?>
                        <?php $words = 25 ?>
                        <?php if (wordCount($record[description]) > $words) : ?>
                        <span class=" body-text-9">...</span><a href="<?
                        php echo $record[_link]; ?>"><span class="
                        body-text-bold-italic-10"> READ MORE</span></a></td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>

```

TAGGING CATEGORIES TO FIND SIMILAR CONTENT - Aug 3rd, 2010

In response to a post from Mickey that asks: Is it possible to add tags or categories to an article and then after the article content have CMSB list those 'tags' in a format that allows a user to click on a tag and then retrieve similarly tagged articles output to say 10 headline (summary contents) items per page?

Jake at Interactive Tools came up with this idea:

To set this up, you'll want to create a textfield for the tags to be entered into - I called mine "Tags". You may also want to specify some instructions for this field that each tag should be separated by a comma. Now create a normal detail viewer page, or open up your existing one. The only thing we're going to do here is add this code:

```
<?PHP $tagList = split(',', $record['tags']); ?>
<?PHP foreach ($tagList as $tagLink): ?>
<a href="/path/to/your/listViewer.php?tags_keyword=<?
PHP echo urlencode(trim($tagLink)); ?>">
<?PHP echo $tagLink ?></a>
<?PHP endforeach ?>
```

That code will split up your tags each time it encounters a comma, and will also remove any spaces entered before or after the comma, if present. It will also URL encode the text should some weird characters make it in there. Finally it will generate a search link for each one of the entered tags.

The only thing you'll need to change in that code is this chunk of text:

```
/path/to/your/listViewer.php
```

This should point to the list viewer page for this section.

If you'd like to limit the results to ten on this page, just enter "10" into the 'limit' option at the start of your list viewer page:

```
'limit' => '10',
```

That should be it!

Great idea, Jake...

PREVIOUS AND NEXT RECORD SEARCH ON DETAILS PAGE - Aug 3rd, 2010

Matrix created an image viewer that displays thumbnail images for a gallery of images as the list viewer page. Each thumbnail is linked to the detail page where the larger image is displayed. The new idea was to add Next & Previous links on each Detail Page for sequential navigation through the individual image pages if someone doesn't want to return to the section list (thumbnail display) each time.

Interactive Tools' Dave Edis came to the rescue once again.
(This works in version 1.18 and above.)

He said: Add this code to the bottom of step 1 in your detail viewer (change the 'Table name' to the name of your table):

```
list($prevRecord, $nextRecord) = getPrevAndNextRecords(array(
    'Table name' => 'news',
    'recordNum' => getNumberFromEndOfUrl(),
));
```

Then add this where you want your prev/next links to go:

```
<?PHP if (@$prevRecord): ?>
<a href="<?PHP echo $prevRecord['_link'] ?>"> &lt;&lt; <?
PHP echo $prevRecord['title'] ?> </a>
```

```

<?PHP echo $nextRecord['title'];</a>
<?PHP endif; ?>

<?PHP if (@$nextRecord): ?>
  <a href="<?PHP echo $nextRecord['_link'] ?>"><?PHP echo $nextRecord['title'] ?
  > &gt;&gt;</a>
<?PHP endif; ?>

```

Voila...

After implementing this, Matrix commented:

"It works like a charm for us. This client is very actively using 10 "albums" contained in a "gallery" all produced through CMS Builder. How we did this was to simply present the thumbnails in a list page, each linked to a detail page with the image and lots of content. With the addition of the next and previous links for each detail page, it's perfect.

So, the interface only requires uploading a new work for each new page, add details and content in a couple of additional text box fields, and publish. It's especially helpful that the detail pages can be re-ordered by dragging. Very nice features."

USING THE MULTI-SELECT DROP DOWN LIST TO CREATE LINKS TO LISTINGS RELATED TO A MAIN CATEGORY - Dec 29th, 2018

Creating links from checked entries in a multi value list field

In a section called "salon_test", I created a multi value list field (salons_presented_in), which gets its option values from the record number (num) field in the table I want to access (salon_eblasts), and its option labels from the title field in that table.

I've created a detail page for the "salon_test" section that gets its record number from the end of the URL.

On that detail page, I needed to create separate links to another detail page for each entry that's checked in the multi value list field (salons_presented_in).

Building on some related products posts in the forum, I came up with this:

At the top of the viewer

```

// load viewer library
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/path_to_your_server/', '..', './', '../', './../');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

// load record from 'salon_test'
list($salon_testRecords, $salon_testMetaData) = getRecords(array(
'tableName' => 'salon_test',
'where'     => whereRecordNumberInUrl(0),
'loadUploads' => true,
'allowSearch' => false,
'limit'     => '1',
));
$salon_testRecord = @$salon_testRecords[0]; // get first record
?>

```

And in the body where I wanted to show the links:

```

<?php
$relatedSalons = join(',', explode("\t", trim($salon_testRecord['salons_presented_in'], "\t")));
$valueFieldName = "num";

if (!$relatedSalons) { $relatedSalons = 0; }

// load records from 'salon_e_blasts'
list($salon_e_blastsRecords, $salon_e_blastsMetaData) = getRecords(array(

```

```
'tableName' => 'salon_e_blasts',
'loadUploads' => true,
'allowSearch' => false,
'where' => "$valueFieldName` IN ($relatedSalons)",
));

?>
<?php foreach ($salon_e_blastsRecords as $record): ?>
<a href="your_second_detail_page.php?<?php echo $record['num']?>"><?
php echo $record['presentation_title'] ?></a>
<?php endforeach; ?>
```

There's also a long thread here, with lots of juicy information:

<http://www.interactivetools.com/forum/gforum.cgi?post=63505>

SEARCHING FOR MULTIPLE WORDS IN MULTIPLE FIELDS - Mar 9th, 2012

In this example, a search would return all records where the field yourfield contains an exact mtch for a word or phrase. If those words appeared in another order, or their was an extra space, or they were found in another field, they would be ignored.

```
<form action="results.php" method="post">
<table>
<tr>
<td >enter search terms:</td>
<td ><input type="text" name="yourfield_keyword" /></td>
</tr>
<tr>
<td>&nbsp;  </td>
<td ><input type="submit" value="Search" /></td>
</tr>
</table>
</form>
```

If you want to search for all the words in a phrase in any order, change your keyword search to a query search.

```
<input type="text" name="yourfield_query" />
```

And if you wanted to search in multiple fields, then you'd have to list all the fields you want it to search, separated by commas, in the input field name.

```
<input type="text" name="yourfield,yourfield2,yourfield3_query" />
```

SEARCHING FOR KEYWORDS SITEWIDE INCLUDING IMAGE INFO FIELDS (MULTISEARCH) - Jul 8th, 2020

Back in 2008 (ancient history...) Dave Edis announced that as of version 1.13 (more ancient history...) a multi search function had been added to CMSB and he offered some example implementation.

I thought that it would be great if I could including upload info fields in the multisearch results.

With a lot of help from Daniel Loewe, a senior programmer at Interactive Tools, I was able to reach my goal.

Here's the code for both the MultiSearch page (called multisearch.php) and the Image Viewer page (called test.php) (no styling):

```
<?php
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/your_server_path/', '..!'; '..!'; '..!'; '..!');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }}
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

$searchOptions = array();
$searchOptions['keywords'] = @$_FORM['q'];
$searchOptions['perPage'] = "100";
$searchOptions['debugSql'] = "0";

$searchTables = array();
$searchTables['family_stories'] = array(
'viewerUrl' => 'story_detail.php',
'titleField' => 'title'
```

```

        'summaryField' => 'sub_title',
        'searchFields' => array('title','sub_title','story'),
    );

    $searchTables['faq'] = array(
        'viewerUrl' => 'faqdetail.php',
        'titleField' => 'question',
        'summaryField' => 'category',
        'searchFields' => array('category','question','answer'),
    );

    $searchTables['uploads'] = array(
        'viewerUrl' => 'test.php',
        'titleField' => 'filePath',
        'summaryField' => 'thumbFilePath',
        'searchFields' => array('filePath', 'info1','info2', 'info3', 'info4', 'info5'),
    );

    list($searchRows, $searchDetails) = searchMultipleTables($searchTables, $searchOptions);
    ?>
    <!-- /STEP1: Load Record List -->
    <?php echo "<?xml version='1.0'?>\n"; ?>
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
    body {
    font-family: arial;
    font-size: 12px;
    }
    </style>
    </head>
    <body>
    <h1>Search Viewer</h1>
    <form method="GET" action="<?php echo $_SERVER['PHP_SELF']; ?>" >
    <input type="text" name="q" value="<?php echo htmlspecialchars(@$FORM['q']); ?>" size="50">
    <input type="submit" name="" value="Search">

    <table border=0 cellpadding=2 cellspacing=0 width=100%>
    <tr>
    <td bgcolor="#EEEEEE"><font size=2 class="maintext">Search results for <b><?php echo htmlspecialchars(@$FORM['q']); ?></b></font> - <font size=2 class="maintext">This page shows results <b><?php echo htmlspecialchars(@$searchDetails['pageResultsStart']); ?></b> to <b><?php echo htmlspecialchars(@$searchDetails['pageResultsEnd']); ?></b> out of <b><?php echo htmlspecialchars(@$searchDetails['totalRecords']); ?></b> found.</font></td>
    </tr>
    </table>

    <!-- STEP3: Display Page Links (Paste anywhere below "Load Record List") -->
    <?php if ($searchDetails['prevPage']): ?>
    <a href="<?php echo $searchDetails['prevPageLink'] ?>">&lt;&lt; previous page</a>
    <?php else: ?>
    &lt;&lt; prev
    <?php endif ?>
    - page <?php echo $searchDetails['page'] ?> of <?php echo $searchDetails['totalPages'] ?> -
    <?php if ($searchDetails['nextPage']): ?>
    <a href="<?php echo $searchDetails['nextPageLink'] ?>">next page &gt;&gt;</a>
    <?php else: ?>
    next &gt;&gt;
    <?php endif ?>
    <!-- /STEP3: Display Page Links -->

    <!-- show errors -->

    <?php if ($searchDetails['invalidPageNum']): ?>
    Results page '<?php echo $searchDetails['page']?>' not found, <a href="<?php echo $searchDetails['firstPageLink'] ?>">start over &gt;&gt;</a>.
    <?php elseif ($searchOptions['keywords'] && $searchDetails['noRecordsFound']): ?>
    No records matched search query!

    <?php elseif ($searchOptions['keywords'] == ""): ?>
    Enter a keyword to search.

    <?php endif ?>

```

```

<!-- STEP2: Display Record List -->
<hr />
<?php foreach ($searchRows as $record): ?>
<a href="<?php echo $record['_link'] ?>"><?php echo $record['_title'] ?></a>
<?php if ($record['_summary']): ?>
<?php echo $record['_summary'] ?>
<?php else: ?>
No description is available for this page.
<?php endif ?>
<a href="<?php echo $record['_link'] ?>" style="color: #008000"><?
php echo $record['_link'] ?></a>
<hr/>
<?php endforeach ?>
<!-- /STEP2: Display Record List -->

</form>
</body>
</html>

```

And for the Image Viewer called test.php (no styling)

```

<?php
$libraryPath = 'cmsAdmin/lib/viewer_functions.php';
$dirsToCheck = array('/your_server_path/', './', '../', './../', './../..');
foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }
if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }

list($uploadsRecords, $uploadsMetaData) = getRecords(array(
'tableName' => 'uploads',
'where' => whereRecordNumberInUrl(0),
'loadUploads' => true,
'allowSearch' => false,
'limit' => '1',
));
$uploadsRecord = @$uploadsRecords[0]; // get first record

?>

<a href="/multisearch.php"><span class=" navigation_font" style="font-
size:24px;">&lt; &lt; BACK TO SEARCH
PAGE</span></a>

<div > 
<?PHP $uploadsRecord['info1'] = preg_replace("/[/\]/", "", $uploadsRecord['info1'] ); ?
>
<?PHP $uploadsRecord['info2'] = preg_replace("/[/\]/", "", $uploadsRecord['info2'] ); ?
>
<?PHP $uploadsRecord['info3'] = preg_replace("/[/\]/", "", $uploadsRecord['info3'] ); ?
>
<?PHP $uploadsRecord['info4'] = preg_replace("/[/\]/", "", $uploadsRecord['info4'] ); ?
>
<?PHP $uploadsRecord['info5'] = preg_replace("/[/\]/", "", $uploadsRecord['info5'] ); ?
>
<div align="center" style="font-size:1.4em; vertical-align:top; text-
align:left;" class="text_font" ><?php
echo($uploadsRecord['info1']) ?> <?php echo($uploadsRecord['info2']) ?> <?
php echo($uploadsRecord['info3']) ?> <?php
echo($uploadsRecord['info4']) ?> <?php echo($uploadsRecord['info4']) ?></div>
The first CMS Builder reference book is now available on-line!

```

SEARCH RESULTS PAGE TELLTALES (THINK BREADCRUMBS) - Jan 3rd, 2014

I needed to create a telltale on a search results page to remind visitors of the search criteria they had entered in a search form.

Everything in the telltale was pretty straight forward (I left the Status (a list field) search, and the all listings search as an example) until I got to the listing age searches.

I wanted to be able to filter the results to only those listings that were created less than either 1, 2, or 3 months ago, and the telltale was returning long date/time indicators instead of just how many months old the listings shown would be.

With a lot of help from Dave Edis, Senior Developer at Interactive Tools, Here's the solution we came up with:

```

<table width="40%" border="0" cellspacing="0" cellpadding="2">
<form method="POST" class="arial_14" action="listings-s.php">
<tr>
<td class="arial_14"><font color="#FFFFFF"><b>Show Listings: </b></font></td>
<td colspan="3">
<select name = "status_match"width="300" class="arial_14" style="width: 300px" >
<option value="">All Listings</option>
<?php foreach (getListOptions('listing' 'status') as $value => $label4): ?>

```

```

<?php foreach ($status_options as $status) {
    <option value = "<?php echo $value;?>" <?
php selectedIf($value, @$_REQUEST['status']);?>> <?php echo
$label4; ?></option>
    <?php endforeach ?>
</select></td>
</tr>

<tr>
    <td class="arial_14"><font color="#FFFFFF"><b>Date Listed: </b></font></td>
    <td colspan="3">
        <select name="months_ago" width="300" class="arial_14" style="width: 300px">
        <option value="">All Dates</option>
        <option value="1">Listed less than 1 Month ago</option>
        <option value="2">Listed less than 2 Months ago</option>
        <option value="3">Listed less than 3 Months ago</option>
        </select>
    </td>
</tr>
<tr>
<td class="arial_14"><b>&nbsp;</b></td>
<td align="left" colspan="3"><input type="submit" name="submit" value="Search" ></td>
</tr>
</form>
</table>

```

And On The Results Page

```

<?php
if (@$_REQUEST['months_ago']) {
    $monthsAgo = $_REQUEST['months_ago'];
    $_REQUEST['createdDate_min'] = mysql_datetime(strtotime("-$monthsAgo month"));
}

?>

<?php
$searchCriteria = "";
if (@$_REQUEST['status_match']) { $searchCriteria .= "Status: {$_REQUEST['status_match']} - "; }

    if (@$_REQUEST['months_ago']) { $searchCriteria .= "Listed Less Than: {$_REQUEST['months_ago']} months ago - "; }
    $searchCriteria = chop($searchCriteria, ', '); // remove trailing , or spaces
    $searchCriteria = chop($searchCriteria, '- '); // remove trailing - or spaces

?>

<div align="center"> <span class="arial_14"> There are <?
php echo $listingMetaData['totalRecords']; ?> listings that
match your search for:
<?php if ( @$searchCriteria ):?>
All Listings
<?php else :?>

<?php echo $searchCriteria ?>
<?php endif ?>
</span>

<a href="search.php"><span class="arial_14">
&lt;&lt; <u>Return to the Advanced Search Page</u></span></a></div>

```

BREAKING E-MAIL LISTS INTO GROUPS OF "N" ADDRESSES EACH - Apr 4th, 2013

To limit abuse on their email server, some ISPs limit the number of recipients that can be included in a single email, or the amount of emails that can be sent at one time.

I encountered this problem when using an email list generated from a multi-record section of publicity email contacts.

Here's the way I broke the email list generated from the email contact records into groups of acceptable numbers.

For this example, you'll need to create a multi-record editor called "publicity" with 2 text fields, "title" and "email"

I also added a max_emails text field in a single record editor called "Common Information" to determine the number of emails allowed in each group.

If statements are used to set the criteria for different email lists. To refine which emails appear in each of your lists, you can use as many criteria as you want to.

In this example there's only one email list displayed on the page, but there are 2 counters used. "counta" to display the total number of records containing email addresses in the list and "countb" to limit the number of emails in each group. Because I wanted the total number to appear above the email list it was associated with, a separate foreach loop

group. Because I wanted the total number to appear above the email list it was associated with, a separate foreach loop, with identical criteria was created to increment counta . You can name your counter variables anything you want, but make sure that each counter variable is unique.

Then, if you're using divs to generate the groups, use the following code in the body of your list viewer

Where you want the Information header to appear:

```
NOTE: To help to conform to ISP recipient limits, email addresses are separated into groups of no more than <?php echo $common_informationRecord['max_emails'] ? > addresses each. System administrators can change this number in the "Common Information" editor.
```

Where you want the list heading to appear:

```
<div align="left"> <?php $counta = 0; ?>
<?php foreach ($publicityRecords as $record ): ?>
  <?php if ($record['email']): ?>
    <?php $counta++; ?>
  <?php endif ?>
<?php endforeach; ?>
<span class="your_class">There are currently <?php echo $counta ? >
> addresses in this list .</span>
</div>
```

And where you want the email list to appear:

```
<div align="left">
<?php $countb = 0; ?>
<?php foreach ($publicityRecords as $record): ?>
<?php if ($record['email']): ?>
<?php $countb++; ?>
<?php echo $record['email'] ?>;&nbsp;
<?php endif ?>
<?PHP $maxemails= $common_informationRecord['max_emails']; ?>
<?php if ($countb >= $maxemails): ?>
<?php $countb = 0; ?>
</div>
<div align="left">
<br />
<hr >
<br />
<?php endif ?>
<?php endforeach ?>
</div>
```

If you're using tables, in place of the code:

```
<?php if ($countb >= $maxemails): ?>
<?php $countb = 0; ?>
</div>
<div align="left">
<br />
<hr >
<br />
<?php endif ?>
<?php endforeach ?>
</div>
```

substitute

```
<?php if ($countb >= $maxemails): ?>
<?php $countb = 0; ?>
</td></tr><tr><td>
<br />
<hr />
<br />
```

```
<?php endif ?>
<?php endforeach ?>
</td>
</tr>
</table>
```

CHAPTER 3 - WORKING WITH UPLOADS

IMAGES

CHANGING AN UPLOAD FIELD'S INFO1 TYPE FROM TEXT FIELD TO TEXT BOX - Oct 25th, 2021

When entering captions on photographs for project, I was continuously running out of room in the info text fields. As an interim solution I used multiple info fields to hold any overflow text and displayed them all on the web page.

A better approach was to automatically convert info1 fields to a text box format, and Carl from Interactive Tools helped me create these 2 plugins to accomplish that goal. He also wrote the code necessary to automatically combine the text from any existing multiple info fields into the new text box so that I didn't have to re-enter all the existing information (use this one with caution).

The resulting set of plugins (you need to activate both) seems to convert both existing upload info1 fields and new ones, but that shouldn't change the way the fields are displayed on your web pages. The plugins also adds a character and word count to all text box fields (which could be used independently).

After the text boxes are created and data is entered in them, deactivating the plugins didn't seem to erase any of the info1 field information from the database, it just changed the info1 field type and how the data is displayed in the back end. Re-activating them restored the new info1 text box format and the information.

There's an executable form accessible in the plugin listing that is used to combine existing info1, 2, 3, 4, and 5 text entries into the new info1 text box.

Please do extensive testing to make sure that there are no adverse effects to your live site when using these plugins.

You can download the plugins from:

<http://www.thecmsbcookbook.com/downloads/convertInfo1.zip>

LIMITING THE AMOUNT OF COLUMNS IN A SINGLE ROW IMAGE DISPLAY - Aug 3rd, 2010

Here's a suggestion from Interactive Tools' Dave Edis:

The CSS way is to have each image in a div that has something like: style="float: left; width: 200px". This will make them flow from left to right and wrap when they run out of space (you can put them inside a bigger fixed width container).

Or with a table you can have one table cell per image and insert a closing and opening TR every X images. Here's a code snippet that does that:

```
<?php $maxCols=2; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>
```

Which is just a way of saying "Every 2 images insert "</tr><tr>"". "

Here's a single row example using the thumbnail2 image as a link to a detail page, separate Title and Subtitle fields, a hidden border for IE and a fixed height for the image cell so that everything lines up nicely.

```
<table width="100%" border="0" cellpadding="5">
<tr>
<?php foreach ($your_sectionRecords as $record): ?><?
php foreach ($record['image'] as $upload): ?>
<td align="center" height="350" valign="bottom">
<a href="<?php echo $record['_link'] ?>">" height="<?php echo $upload['thumbHeight2'] ?
">" alt="" style="border:hidden" /><br /><div
align="center" class="medium-title"><?php echo $record['title'] ?></div>
```

```

<div align="center" class="medium-text"><?php echo $record['sub_title'] ?></div></a>
</td>
<?php $maxCols=2; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php endforeach ?><?php endforeach ?>
</tr>
</table>

```

Similarly, if you're not using images this code will output a </tr>
<tr> after every 2 <td>...your content...</td>
lines:

```

<table border="1">
<tr>

<?PHP foreach ($yourrecords as $record): ?>
<td>
<?PHP echo $record['your_content'] ?><br />
</td>
<?PHP $maxCols=2; if (@++$count % $maxCols == 0): ?></tr><tr><?PHP endif; ?>
<?PHP endforeach; ?>
</tr>
</table>

```

Use your own variable names instead of \$yourrecord, but the inserted code can stay the same.

LIMITING THE AMOUNT OF COLUMNS IN A MULTI ROW IMAGE DISPLAY - Feb 9th, 2013

The table example above is fine for a single row of images or information that you want to wrap every x columns. If you want to set up sets of rows that you want to wrap, you'll have to wrap your table inside a table. Here's a 2 row example using the thumbnail2 image as a link to a detail page, separate Title and Subtitle fields, a hidden border for IE and a fixed height for the image cell so that everything lines up nicely.

```

<table width="100%" border="0" cellpadding="5">
<tr>
<td align="center" >
<tr>
<?php foreach ($your_sectionRecords as $record): ?><?
php foreach ($record['image'] as $upload): ?>
<td align="center" width="50%">
<table>
<tr>
<td height="350" valign="bottom"><a href="<?php echo $record['_link'] ?>">
" height="<?
php echo $upload['thumbHeight2'] ?>" alt=""

```

```

style="border:hidden" /></a>
</td>
</tr>
<tr>
<td align="center" class="medium-title"><?php echo $record['title'] ?></div>
<div align="center" class="medium-text"><?php echo $record['sub_title'] ?></div>
</td></tr></table>
</td>
<?php $maxCols=2; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php endforeach ?> <?php endforeach ?>
</tr>
</table>

```

Again, Use your own variable names instead of \$your_sectionRecord, but the inserted code can stay the same.

*** If you're going to use that little code snippet multiple times on the same page, just use a different variable instead of \$count or it will remember the count from the last section and increment that.

Instead of this:

```

<?php $maxCols=5; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>

```

Use these:

```

<?php $maxCols=5; if (@++$count1 % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php $maxCols=5; if (@++$count2 % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php $maxCols=5; if (@++$count3 % $maxCols == 0): ?></tr><tr><?php endif; ?>

```

LIMITING THE AMOUNT OF COLUMNS IN A MULTI COLUMN IMAGE DISPLAY - Aug 3rd, 2010

If you want to set up sets of columns that you want to wrap, you'll have to wrap your table of columns inside a table also.

Here's a similar example to the above that lists sets of contact information next to images.

```

<table width="100%" border="0" align="left" cellpadding="0">
<tr>
<td align="left" >
<tr>
<?php foreach ($your_sectionRecords as $record): ?><?
php foreach ($record['image'] as $upload): ?>
<td align="center" width="50%">

<table border="0" align="left" cellpadding="5">
<tr>
<td align="left" valign="top">

<table width="100%" border="0" cellpadding="5">
<tr>
<td align="left" valign="top"><a href="<?php echo $record['_link'] ?>">" height="<?
php echo $upload['thumbHeight'] ?>" alt="" style="border:hidden"
/></a></td>
<td align="left" valign="top"><div class="yourclass"><?php echo $record['title'] ?>
</div>
<div class="yourotherclass"><?php echo $record['name'] ?><br />
<?php echo $record['tel_no'] ?><br /><?php echo $record['fax_no'] ?><br /><?
php echo $record['email'] ?></div></td>
</tr>
</table>
</td>
</tr>
</table>
</td>

```

```
</tr>
</table>

</td>
<?php $maxCols=4; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php endforeach ?> <?php endforeach ?>
</tr>
</table>
```

LETTING YOUR CLIENT DECIDE HOW MANY COLUMNS TO DISPLAY - Aug 3rd, 2010

If you want to allow your client to enter the number of columns that they'd like to display, you can do it like this.

In a single record editor (I have one for all of the common information on my site) add a field called `thumbnail_column_limit`

Then create a variable to replace the maxcol number:

```
<?PHP $col = ($common_informationRecord["thumbnail_column_limit"]); ?>
```

And replace the number in your maxcol code with that variable, like this:

```
<?php $maxCols=$col; if (@++$count % $maxCols == 0): ?> </tr> <tr>
<?php endif; ?>
```

SHOWING ONLY THE FIRST IMAGE OR GROUP OF IMAGES ON A PAGE - Oct 13th, 2011

User Perchpole asked: On a List page, how do I limit the number of (image) uploads displayed with each record to one?

Dave answered:

If you want to show just one image you can use `<?PHP break ?>` like this to stop after just one:

```
<?PHP foreach ($record["gallery_images"] as $upload): ?>
" width="<?
PHP echo $upload["thumbWidth"] ?>" height="<?PHP echo
$upload["thumbHeight"] ?>" alt="" />
<?PHP break ?>
<?PHP endforeach ?>
```

Jason Sauchuk from Interactive Tools suggested that another option would be to limit your query to only one result and then put the first record in it's own variable.

For example:

```
// load records
```

```
list($gallery_imagesRecords, $gallery_imagesMetaData) = getRecords(array(
    'tableName' => 'gallery_images',
    'limit' => 1,
));

$gallery_imagesRecord = $gallery_imagesRecords[0]; // get first record.
```

You can then use the variable `$gallery_imageRecord` to show various fields in the first record without needing a `foreach` loop.

A variation on this is the ability to show a specified number of images from a series of records before a `<?PHP break ?>`

Jason Sauchuk offered the following approach:

```
<?php $count = 0; ?>

<?php foreach ($record['gallery_images'] as $upload): ?>

" height="<?PHP echo
$upload['thumbHeight'] ?>" alt="" />

<?php if (++$count > 3) { break; } ?>

<?php endforeach ?>
```

To randomize the images shown:

```
<?php $count = 0; ?>

<?php shuffle($record['gallery_images']); ?>

<?php foreach ($record['gallery_images'] as $upload): ?>

" height="<?PHP echo
$upload['thumbHeight'] ?>" alt="" />

<?php if (++$count > 3) { break; } ?>

<?php endforeach ?>
```

was exceeded. The limit part was easy. In the get records code at the top of the page I entered

```
'limit' => '20',
```

To display the code if the number was exceeded, (in this case a link to a second viewer) I used the following: (I was already using a counter on this page so I called this counter count2)

```
<?php $count2 = 0; ?>
<?php foreach ($portfolio_imagesRecords as $record): ?>
  <?php $count2++; ?>
<?php endforeach; ?>

<?php if ($count2 >19): ?>
<a href="http://www.your_site.com/your_viewer2.php">Click for more images.<?
php endif ?>
```

You could also use this approach to display different messages:

```
<?php $count2 = 0; ?>
<?php foreach ($portfolio_imagesRecords as $record): ?>
  <?php $count2++; ?>
<?php endforeach; ?>

<?
php echo $count2>19 ? "There are over the maximum number of thumbnails" : "You cannot add more thumbnails" ; ?
>
```

COUNTING IMAGES IN A MULTI-IMAGE UPLOAD FIELD - Jan 18th, 2012

I needed to count the number of images that existed in a multi-image upload field that was the only record in my editor, so that if there was only one image I could display it as an image and if there was more than one image I could display them as a slide show.

I started out with another recipe in the Cookbook called: LIMITING THE NUMBER OF IMAGES ON A PAGE AND SHOWING CODE IF EXCEEDED (just above this one)

The basic approach in that recipe is:

```
<?php foreach ($home_page_slidesRecords as $record): ?>
<?php $count = 0; ?>
<?php foreach ($record['image'] as $upload): ?>
  <?php $count++; ?>
<?php endforeach; ?>
<?
php echo $count>1 ? "There is more than one thumbnail" : "There is only one thumbnail" ; ?
```

```
>  
<?php endforeach; ?>
```

I modified the code so that instead of messages, the required code was inserted into the page. I also added a checkbox field to allow the client to decide if they wanted to use this as a slide show regardless of how many images were uploaded. I also added `<?php shuffle($record['images']) ?`
> to the code to randomize the image displayed when there was no slide show.

```
<?php foreach ($home_page_slidesRecords as $record): ?>  
<?php $count = 0; ?>  
<?php foreach ($record['image'] as $upload): ?>  
<?php $count++; ?>  
<?php endforeach; ?>  
<?php if ($count > 1 && $home_page_slidesRecord['slideshow'] == 1): ?>  
<div id="show" class="slideshow"></div>  
<?php else: ?>  
<?php shuffle($record['images']) ?><?php foreach ($record['image'] as $upload): ?>  
" height="<?PHP echo $upload['thumbHeight']  
?>" alt="" />  
<?PHP endforeach ?>  
<?php endif ?>  
<?php endforeach; ?>
```

DISPLAY IMAGES RANDOMLY - Aug 3rd, 2010

If you want to randomize how your images are displayed each time your page is reloaded you can shuffle your array first:

```
<?PHP shuffle($record['gallery_images']) ?>  
<?PHP foreach ($record['gallery_images'] as $upload): ?>  
" height="<?PHP echo $upload['thumbHeight'] ?>" alt="" />  
<!-- Optional Code -->  
<?PHP break ?>  
<!-- End Optional Code -->  
<?PHP endforeach ?>
```

The `<?PHP break ?>` stops the foreach loop so that only a single image is displayed.

Cool huh!!

DISPLAY A BACKGROUND IMAGE USING ONLY THE URL OF THE IMAGE OR THUMBNAIL - Aug 3rd, 2010

If you're using an image as a background image for a page, you can use the code below in the CSS to specify the location of the image:

```
<?PHP foreach ($record['gallery_images'] as $upload): ?>  
<?PHP echo $upload['uriPath'] ?>  
<?PHP endforeach ?>
```

Or

```
<?PHP foreach ($record['gallery_images'] as $upload): ?>
<?PHP echo $upload['thumbUrlPath'] ?>"
<?PHP endforeach ?>
```

If you want to specify a particular thumbnail's location, just put a 2, 3 or 4 after thumbUrlPath

DISPLAYING THUMBNAIL 2, THUMBNAIL3, THUMBNAIL4 - Aug 3rd, 2010

If you've created these thumbnails, you can use them by adding a 2, 3, etc. at the end of the "thumbUrlPath", "thumbWidth" and "thumbHeight" statements.

Dave added:

To test for the existence of the alternate thumbnails, just add the number (2, 3, or 4) on the end and test for thumbUrlPathX instead of "hasThumbnail". If there's no thumbnail, thumbUrlPathX will be blank and test as false.

```
<?php if ($upload['thumbUrlPath2']): ?>
<a href="<?php echo $upload['urlPath2'] ?>" rel="lightbox[model]">
"
height="<?php echo $upload['thumbHeight2'] ?>" alt="" />
<?php endif ?>
```

DISPLAYING A GENERIC IMAGE OR SPECIAL TEXT IF NO IMAGE IS UPLOADED - Aug 3rd, 2010

A number of users have asked how to display a generic image (or specific text) if no image has been uploaded to a record. Well there are a number of ways. One offered by Dave Edis from Interactive Tools is:

```
<?php foreach ($your_tableRecords as $record): ?>

    <?php if (sizeof($record['your_image_field']) >= 1): ?>

        <?php foreach ($record['your_image_field'] as $upload): ?>
        <?php if ($upload['hasThumbnail']): ?>
        <a href="<?php echo $record['_link'] ?>">" height="<?php echo $upload['thumbHeight'] ?
>" align="center" /></a>
        <?php elseif ($upload['isImage']): ?>
        " height="<?php echo
$upload['height'] ?>" align="right" />
        <?php endif ?>
        <?php endforeach ?>

        <?php else: ?>
        
        <?php endif ?>
        <?php endforeach; ?>
```

ROLLOVER IMAGES AND CMSB - Aug 3rd, 2010

Creating Rollover images is pretty simple, here's how Dreamweaver does it.

When a rollover scenario is created, this script gets inserted in the head

```
<script type="text/javascript">
<!--
function MM_swapImgRestore() { //v3.0
    var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&
(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
    var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
    var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
    if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
    var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-
2);i+=3)
        if ((x=MM_findObj(a[i]))!=null)
            {document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
</script>
```

To preload the rollover image(s) this code gets inserted in the opening <body> tag code :

```
<body onload="MM_preloadImages('path_to/your_normal_image.jpg','path_to/your_rollover_image.jpg')">
```

And where the rollover image is to appear, this code is inserted:

```
<a href="http://www.your_site.com" onmouseout="MM_swapImgRestore()"
onmouseover="MM_swapImage('image_name','path_to/your_rollover_image.jpg',1)">
</a>
```

To integrate this into CMSB:

Add the javascript above to the head section of your page, then to preload your images:

For a multi record editor on a list page, you'd insert this into the opening <body> tag

```
<body onload="MM_preloadImages('<?
php foreach ($your_tableRecords as $record): ?><?php foreach
($record['your_roll_over_image_field'] as $upload): ?><?
php echo $upload['thumbUriPath'] ?>', <?php endforeach;
?><?php endforeach; ?>'blank.jpg')">
```

(The call to blank.jpg adjusts the syntax of the commas between entries.)

For a single record editor or a detail page the code would be:

```
<body onload="MM_preloadImages(/>
```

```

<body onload= mm_preloadImages(
php foreach ($your_tableRecord[your_rollover_image_field] as $upload): ?><?
php
echo $upload['thumbUrlPath'] ?>', <?php endforeach; ?>'blank.jpg')">

```

(Note: If you're adding a series of roll over images to some you already have, just add their calls to the end of the existing preload list)

Then in the Body, you'd insert the following where you wanted the rollovers to be displayed.

For a multi record editor on a list page, you'd insert:

```

<?php foreach ($your_tableRecords as $record): ?>

<a href="http://www.your_site.com"

onmouseout="MM_swapImgRestore()"

onmouseover="MM_swapImage('<?php echo $record[num] ?>',", '<?
php foreach ($record[your_roll_over_image_field] as
$upload): ?><?php echo $upload['thumbUrlPath'] ?> <?php endforeach; ?>',1)">

" height="<?
php echo $upload['thumbHeight'] ?>" border="0"
<?php endforeach; ?>id="<?php echo $record[num] ?>" /></a>

<?php endforeach; ?>

```

The names and ids of your images must be unique. I've used the record number to accomplish this, but you can use any unique field value that is associated with the record.

For a single record editor or detail page with multiple single image fields you'd use:

```

<?php foreach ($your_tableRecord[your_normal_image_field] as $upload): ?>

<a href="http://www.your_site.com"
onmouseout="MM_swapImgRestore()"

onmouseover="MM_swapImage('<?php echo $upload[info1] ?>',", '<?
php echo $upload['thumbUrlPath'],1)">

"
width="<?php echo $upload['thumbWidth'] ?>" height="<?
php echo $upload['thumbHeight'] ?> border="0" id="<?php echo
$upload[info1] ?>" /></a>

<?php endforeach; ?>

```

The names and ids of your images must be unique. I've used info2 to accomplish this, but you can use any info field value that is associated with the image.

```

<a href="http://www.your_site.com"

onmouseout="MM_swapImgRestore()"

onmouseover="MM_swapImage('<?php foreach ($your_tableRecord[your_normal_image_field] as $upload): ?
>>?php echo
$upload[info2] ?><?php endforeach ?>',", '<?
php foreach ($your_tableRecord[your_roll_over_image_field] as $upload):
?>

<?php echo $upload['thumbUrlPath'] ?><?php endforeach ?>',1)">

<?
php echo $upload[info2] ?><?php
endforeach ?>"

```

```

name="<?php foreach ($your_tableRecord['your_normal_image_field'] as $upload): ?
><?php echo $upload['info2'] ?><?php
endforeach ?>"
width="<?php foreach ($your_tableRecord['your_normal_image_field'] as $upload): ?>
<?php echo $upload['thumbWidth']
?><?php endforeach ?>"
height="<?php foreach ($your_tableRecord['your_normal_image_field'] as $upload): ?
><?php echo $upload['thumbHeight']
?><?php endforeach ?>"
border="0"
id="<?php foreach ($your_tableRecord['your_normal_image_field'] as $upload): ?><?
php echo $upload['info2'] ?>
<?php endforeach ?>" />
</a>
<?php endforeach ?>

```

There are many ways to slice and dice the rollover scenario, but if you mock it up in HTML first, and then replace the values as needed with your PHP calls, keeping in mind the unique identifier requirement, you should be fine.

CREATING A DETAILS PAGE TO DISPLAY A SINGLE IMAGE FROM THE MULTI IMAGE UPLOAD AND USING INFO FIELDS - Aug 3rd, 2010

The standard code block still goes at the top of the page:

```

<?php

require_once "/path_to_your/cmsAdmin/lib/viewer_functions.php";
list($photographyRecords, $photographyMetaData) = getRecords(array(
    'Table name' => 'photography',
    'where'      => whereRecordNumberInUrl(1),
    'limit'     => '1',
));
$photographyRecord = @$photographyRecords[0]; // get first record?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

Then for example, if you want the 5th image (counting starts with "0") in the series that was the size specified in thumbnail 2 to be displayed on the page you would call it with:

```

" height="<?PHP echo $upload['thumbHeight2']?
>" style="margin-bottom: 5px" />

```

And if you wanted to use the info fields associated with that specific image you would call them with (substituting the info number for "X"):

```
<?PHP echo $photographyRecord['images'][4]['infoX']?>
```

If you want to test for an image first you can use the code:

```
<?PHP if ($photographyRecord['images'][4]): ?>
...
<?PHP endif ?>
```

Pretty simple when you know how isn't it.

LINK THUMBNAILS TO FULL-SIZED VERSION OF IMAGES - Aug 3rd, 2010

By default, this is the code used to generate the image tags for thumbnails is:

```
<?PHP if ($upload['hasThumbnail']): ?>
  " height="<?PHP
echo $upload['thumbHeight'] ?>" alt="" /><br />
```

If you want your thumbnails to link to the full-sized version of the image, you can substitute that code with this:

```
<?PHP if ($upload['hasThumbnail']): ?>
  <a href="<?PHP echo $upload['urlPath'] ?>">" height="<?PHP echo $upload['thumbHeight'] ?
>" alt="" /></a><br />
```

MAKE A TITLE AND CAPTION APPEAR UNDER OR ABOVE AN IMAGE - Aug 3rd, 2010

Uploads can have up to 5 additional "info" fields. By default Title and Caption are defined. If you are displaying your images with a variable called \$upload you can define values for the title and caption by adding fields named info1, info2,...info5 to your section editor and then displaying them on your web page like this:

```
Title: <?PHP echo $upload['info1']; ?>
Caption: <?PHP echo $upload['info2']; ?>
Your Definition: <?PHP echo $upload['info3']; ?>
```

CREATING PHOTO GALLERY DETAIL PAGES - Aug 3rd, 2010

You can create a series of detailed pages that can be found on a listing page by using the approach below.

We'll use the example of a simple image gallery that an organization can use to display images from it's past events.

We'll use galleryList.php as a template for a gallery list page

galleryList.php is set up to show the "title" for all past event records available, with links to that particular event's record.

We'll use galleryDetail.php as a template for the image collections from the individual events held that year

Each Gallery record is in a multi record format with fields for a "title", an "event description", and up to 30 individual "images". The images are set to limit uploads to a maximum size of 1000KB, to resize any images larger than 800 x600 pixels, and to create thumbnails of no more than 275 pixels in their larger dimension.

If you want to keep the events of a single year (or group) on separate list pages, then you'll want to create a separate section editor for each year. You can set up the links to the list pages on a master links page or on a navigation menu.

If you're using this approach for a different application other than a straight ahead photo gallery, you can create your first section editor and then make it a template in the section editor. Don't forget to change the Search URL and field names to match your new template's values. And to change any image limitations as necessary.

Copying a sectioneditor.ini.php file from your on-line cmsAdmin/data/schema/ folder to your cmsAdmin/data/schemaPresets/ folder and it will show up in the "add section" pull down menu.

After creating a section editor for your project, you can make this section editor into a template, by:

Going to your on-line cmsAdmin/data/schema/ folder and downloading the new sectioneditor.ini.php file.

Opening the downloaded file in your web page editor.

To change the name that appears in the pull down list, change the "menuName" value in the first section of the new sectioneditor.ini.php file.

Change any other "label" values to generic values if necessary. You'll be changing these individually to suite when you create your new section editors.

Upload your modified new sectioneditor.ini.php file to your cmsAdmin/data/schemaPresets/ folder and it will show up in the "add section" pull down menu

To add a set of image galleries for a new year:

Create new section editor from the Gallery preset menu
change the Menu Name and Table Name as appropriate.
Modify Viewer URLs as appropriate.

Add a hidden field if appropriate and add neverRemove and hidden to show in the editor menu.

Create a new years list and details pages by replacing all the template year references with the new year.

CONTROLLING THE TYPES OF FILES THAT CAN BE UPLOADED - Aug 3rd, 2010

To modify one of your existing upload fields to accept .wav or mp3 files, go to Admin -> Section Editors and click the "modify" link for the section this field is being used in. Now click the "modify" link for this specific field, and then click on the "Show All" link for the Input Validation section. Here, there's a field called "File extensions allowed:", which you can add "wav" to. This field acts as a comma-separated list. WITHOUT THE PERIODS. Here's an example of the format to follow when adding in file type extensions here:

```
wav,wmv,wma,aif
```

Just save your setting adjustments and you'll be able to upload specific types of files for that field.

ALIGNING IMAGES SO THAT TEXT WRAPS AROUND THEM - Aug 3rd, 2010

Here's a suggestion that was offered by CMSB user equinox69. He said:

If you don't want to put the data into separate table fields then a simple html trick to make it wrap for any browser is the following:

If you want the image to show on the right and have text wrap on the left and below the image automatically adjusting do this: `` text here text here text here text here, etc.

If you want the image to show on the left and have the text wrap on the right and below the image automatically adjusting, then do this:

```
text here, text here text here text here, etc. 
```

It's worked in every version of IE since at least 3.0 and netscape, mozilla, etc. The browser want to put the text on the side the graphic is listed but the graphic is forced to the other side so the browser forces the graphic over and wraps the text appropriately.

DISPLAYING NEWEST IMAGES FIRST - Aug 3rd, 2010

NigelGordijk asked if there was a way to display images as newest at the top of a gallery instead of the oldest images first.

As usual, Dave Edis from Interactive Tools had the perfect solution. He said:

You can reverse the order with `array_reverse()`.

If your current code looks something like this:

```
<?php foreach ($record['uploads'] as $upload): ?>
```

Just add "array_reverse" to your code:

```
<?php foreach (array_reverse($record['uploads']) as $upload): ?>
```

Thanks Dave.

INSERTING RANDOM ROTATING IMAGES ON A WEB PAGE - Aug 3rd, 2010

Inserting images or banners on a web page and having them change each time the page is reloaded is really easy.

For this example we'll display one image at random from a simple multi record section editor. The fields in the editor are Title, an image upload field limited to one upload per record and a link URL that's entered into the first extra information field (info1) in the upload advanced options.

The code that displays the images and links on the web page is identical to any other image display code:

```
<?php foreach ($your_tableRecords as $record): ?>
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>"> " height="<?php echo $upload['thumbHeight2'] ?
```

```

<?php echo $upload['thumbHeight2'] :
>" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>

```

The secret is using various options to determine how the image records in your_table will be displayed. For our situation we want to randomize the order of the images and then display only one image. We'll use the option 'orderBy' => 'RAND()', to randomize the order and 'limit' => '1', to limit the number of images that can be displayed at a time.

Here's the code that would appear at the top of your PHP web page:

```

<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/your_path_to/cmsAdmin/lib/viewer_functions.php";

list($your_tableRecords, $your_tableMetaData) = getRecords(array(
    'Table name' => 'your_table',
    'orderBy' => 'RAND()',
    'limit' => '1',
));

```

If you wanted to have more than one group of images or banners display on your page, just create a second multi record editor (your_second_table) and insert the following at the top of your PHP page:

```

<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/your_path_to/cmsAdmin/lib/viewer_functions.php";

list($your_tableRecords, $your_tableMetaData) = getRecords(array(
    'Table name' => 'your_table',
    'orderBy' => 'RAND()',
    'limit' => '1',
));

list($your_second_tableRecords, $your_second_tableMetaData) = getRecords(array(
    'Table name' => 'your_second_table',
    'orderBy' => 'RAND()',
    'limit' => '1',
));

```

and display the images from that table like this:

```

<?php foreach ($your_second_tableRecords as $record): ?>
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>"> " height="<?php echo $upload['thumbHeight2'] ?
>" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>

```

INSERTING RANDOM ROTATING IMAGES ON A WEB PAGE FROM RANDOM RECORDS - May 19th, 2013

User SkidderChains wanted to provide images for a rotating image header with an interesting twist. The images had to be pulled from a random record in a multi-record editor and then from a random image in that record's image upload field.

Greg Thomas from Interactive Tools offered this solution:

```
// load records from 'gallery_2'
list($gallery, $gallery_2MetaData) = getRecords(array(
    'tableName' => 'gallery_2',
    'orderBy' => 'RAND()',
    'loadUploads' => true,
    'allowSearch' => false,
));

//shuffle all of the images for each record
foreach($gallery as $images){
    shuffle($images['image']);
}

//counter
$n = 0;
//max number of images you want to pull from each section;
$max = 5;
while($n <= $max): ?>

    <?php foreach($gallery as $key => $images): ?>

        <?
php //If there is a picture in the image array with a key of $n, assign it to the $image variable, else return
false
        if($image = @$images['image'][$n]): ?>

            "/>
            <?php endif; ?>

        <?php endforeach; ?>
        <?php $n++; ?>
    <?php endwhile; ?>
```

Greg said:

This is just an example, and you will have to modify the code to work with your setup.

"I'm using the getRecords function to randomly sort the sections, then cycling through them and displaying each individual image using the \$n variable to select a single image from the image array. So first it will display all of the images from each section with a key of 0, then 1, 2 etc.

DISPLAYING IMAGES BLOCKS IN A RANDOM ORDER - Sep 19th, 2010

A client of mine wanted to have 4 groups of images and a navigation menu display in a single row, 5 column table on their home page. They wanted the images to change in a random fashion and the navigation menu to appear in a different cell each time the page was visited.

I was able to help them by combining the random display concept above with a simple javascript program that I found on the vBulletin support forum.

<http://forum.vbulletinsetup.com/f62/display-content-random-order-xhtml-valid-2355.html>

I created 4 section editors like the ones above, and then created a web page with the following code:
The class="group1" is used by the randomizing script to swap the contents of the cells.

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php

require_once "/path_to_your/cmsAdmin/lib/viewer_functions.php";

list($images_1Records, $images_1MetaData) = getRecords(array(
    'Table name' => 'images_1',
    'orderBy' => 'RAND()',
    'limit' => '1',

));

list($images_2Records, $images_2MetaData) = getRecords(array(
    'Table name' => 'images_2',
    'orderBy' => 'RAND()',
    'limit' => '1',

));

list($images_3Records, $images_3MetaData) = getRecords(array(
    'Table name' => 'images_3',
    'orderBy' => 'RAND()',
    'limit' => '1',

));

list($images_4Records, $images_4MetaData) = getRecords(array(
    'Table name' => 'images_4',
    'orderBy' => 'RAND()',
    'limit' => '1',

));

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="scripts/random.js"></script>

<title>Random Blocks</title>
</head>

<body >
<table border="0" align="center" cellpadding="0">

<tr>
<td valign="top">
<?php foreach ($images_1Records as $record): ?>
<div class="group1">
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>"> " height="<?php echo $upload['thumbHeight2'] ?
">" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>
</div></td>
```

```

<td valign="top">
<?php foreach ($images_2Records as $record): ?>
<div class="group1">
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>">" height="<?php echo $upload['thumbHeight2'] ?
>" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>
</div></td>

<td valign="top">
<?php foreach ($images_3Records as $record): ?>
<div class="group1">
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>">" height="<?php echo $upload['thumbHeight2'] ?
>" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>
</div></td>

<td valign="top">
<?php foreach ($images_4Records as $record): ?>
<div class="group1">
<?php foreach ($record['image'] as $upload): ?>
<a href="<?php echo $upload['info1'] ?>">" height="<?php echo $upload['thumbHeight2'] ?
>" style="border:hidden" alt="" /></a>
<?php endforeach ?>
<?php endforeach ?>
</div></td>

<td valign="top"><div class="group1">
<table align="center" width="90%" border="0" cellpadding="0">
<tr>
<td valign="top" ><a class="your-class" href="your_index.php">Home</a></td>
</tr>
<tr>
<td valign="top" ><a class="your-class" href="page_1.php">Menu Item 1</a></td>
</tr>
<tr>
<td ><a class="your-class" href="page_2.php">Menu Item 2</a></td>
</tr>
<tr>
<td ><a class="your-class" href="page_3.php">Menu Item 3</a></td>
</tr>
<tr>
<td ><a class="your-class" href="page_4.php">Menu Item 4</a></td>
</tr>
</table>

</div></td>

</tr>

</table>

<script type="text/javascript">
//randomize order of contents with DIV class="group1"
randomizeContent("group1")
</script>

</body>
</html>

```

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

