The CMS Builder Cookbook

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

ADDING LINKS TO THE CMSB INTERFACE USING THE MODIFYHEADERLINKS PLUGIN
***WAS CALLED EXAMPLEHEADERLINKS* - Mar 27th, 2019**

Since the header_links hook has been removed from CMSB starting with the release of version 3.0, Dave Edis from Interactive Tools has shared a new version of the examplHeaderLinks plugin called modifyHeaderLinks.php that allows the addition of custom links in the upper left portion of the main CMSB management interface page.

Here's Dave's plugin code:

```
<?php


// NOTE: RENAME this plugin so when you download updates you don't OVERWRITE your changes!!!

addFilter('menulinks_myAccount', 'modifyHeaderLinks');

//
function modifyHeaderLinks($menuArray) {

  // Example of custom menu:



  // remove license link
  foreach ($menuArray as $index => $menuAttr) {
    if (preg_match("/\b(menu=license)\b/", @$menuAttr['link'])) { unset($menuArray[$index]); }
  }

  // add links to the beginning
  $newMenu = [
    'menuName'        => t('New Menu'),
    'menuType'        => 'custom',
    'link'            => "?menu=new_menu",
    'visibility'      => 'requireLogin',                  // will be displayed in the user is logged in
    'isSelected'      => (@$_REQUEST['menu'] == 'new_menu'), // set to true to show this menu as selected
    'br_after'        => true,                             // optional: add line break after menu item instead of
separator ("|")
  ];
  array_unshift($menuArray, $newMenu);

  // add links to the end
  $newMenu = [
    'menuName'        => 'Google',
    'menuType'         => 'custom',
    'link'             => "https://www.google.com/",
    'isSelected'       => false,                    // set to true to show this menu as selected
    'linkTarget'       => '_blank',                 // optional: set this to open link in a new tab
    'visibility'       => 'showAlways',             // will be displayed always
  ];
  array_push($menuArray, $newMenu);

  return $menuArray;
}

// eof
```

I was a bit confused as to how to implement the plugin until Daniel Louwe, a programmer at Interactive Tools came to my rescue.

Daniel said:

"For adding these external links before the 'My Account' link , you would use something like this:

```
$newMenu = [
  'menuName' => 'The Free Pixlr On-Line Image Editors &gt;&gt;',
  'menuType' => 'custom',
  'link' => "http://pixlr.com/express/",
  'isSelected' => false,
  'linkTarget' => '_blank',
  'visibility' => 'showAlways',
  'br_after' => true,
];
array_unshift($menuArray, $newMenu);

$newMenu = [
  'menuName' => 'How to Use Pixlr Express &gt;&gt;',
  'menuType' => 'custom',
  'link' => "http://www.elleschorrphotography.com/using-pixlr.php",
  'isSelected' => false,
  'linkTarget' => '_blank',
  'visibility' => 'showAlways',
  'br_after' => true,
];
array_unshift($menuArray, $newMenu);
```

And to add external links after the 'View Website' link

```
 $newMenu = [
    'menuName'         => 'COMPLETE IMAGEUPLOADING INSTRUCTIONS >> ',
    'menuType'         => 'custom',
    'link'             => "http://www.yoursite.com/faqdetails.php?21",
    'isSelected'       => false,                    // set to true to show this menu as selected
    'linkTarget'       => '_blank',                 // optional: set this to open link in a new tab
    'visibility'       => 'showAlways',             // will be displayed always
     'br_after'        => true,
  ];
  array_push($menuArray, $newMenu);

   $newMenu = [
    'menuName'         => 'UPLOAD YOUR NEW IMAGES ORREVISE EXISTING SUBMISSIONS >> ',
    'menuType'         => 'custom',
    'link'             => "http://www.yoursite.com/cmsAdmin/admin.php?
menu=exhibition_submissions",
    'isSelected'       => false,                    // set to true to show this menu as selected
    'linkTarget'       => '_blank',                 // optional: set this to open link in a new tab
    'visibility'       => 'showAlways',             // will be displayed always
     'br_after'        => true,
  ];
  array_push($menuArray, $newMenu);
```

As you can see, you can keep using the $newMenu variable name - simply call either array_unshift($menuArray, $newMenu) (add before the 'My Account' link) or array_push($menuArray, $newMenu) (add after the 'View"Website' link) after each option array definition to add it to the menu.

In answer to "how would I set the following line to TRUE?

```
 'isSelected' => (@$_REQUEST['menu'] == 'new_menu'), // set to true to show this menu as selected
```

"

Daniel offered the following:

This is only necessary for links that point to pages within CMSB and is mostly done automatically. You would just need to make sure that the value in that line matches the "menu=" value in the link option. For example:

'link' => "?menu=example",
'isSelected' => (@$_REQUEST['menu'] == 'example'),

AUTOMATICALLY POPULATING FIELDS IN A TARGET TABLE USING THE CREATEDBY FUNCTIONALITY IN CMSB - Dec 9th, 2018

The createdBy functionality built into CMSB allows you to echo field values from the account record of the person who created a record in another table.

I was trying to use that createdBy functionality to sort records by last name in a multi-record table that does not contain a last_name field field.

When I tried to sort the records in the table using 'orderBy' => 'createdBy.last_name ASC', in the load records call, that didn't work.

Thankfully, Dave Edis came to the rescue once again.

He pointed out that although you can't use the createdBy functionality for sorting directly, you can use a plugin to populate real fields in your target table from those createBy values and then use the actual field data to sort your records.

Based on what he suggested, here's the plugin that we came up with:

```
<?php


addAction('record_postsave', function($tableName) {

   $sourceField   = "first_name"; // from accounts table
   $fieldToUpdate = "createdBy_first_name";
   $sourceField2   = "last_name"; // from accounts table
   $fieldToUpdate2 = "createdBy_last_name";
 // Error Checking
 if (!isset($GLOBALS['schema']
[$fieldToUpdate])) { return; } // skip if field doesn't exist in current table
 if (!isset($GLOBALS['schema']
[$fieldToUpdate2])) { return; } // skip if field doesn't exist in current table

//Update Values
 $tablePrefix   = $GLOBALS['TABLE_PREFIX'];

  $query = "UPDATE {$tablePrefix}$tableName t
         LEFT JOIN {$tablePrefix}accounts a ON t.createdByUserNum = a.num
               SET t.`$fieldToUpdate` = a.`$sourceField`,
              `$fieldToUpdate2` = a.`$sourceField2`";

  mysql_do($query);
});
```

USING THE EMAILONAPPROVED PLUGIN WITH ANY MULTI-RECORD SECTION - Oct 26th, 2017

Because I didn't want to allow non-members to upload images (or other files) directly to the database, I wanted to send an email to a person that posted a listing request once it was approved and listed. That email would ask them to reply to the email with an image attached.

I was at a loss until I got some help from Jeff Shields a long time CMSB user.

Here's how it works:
A person submits an event for listing.
The submission creates an email to the admin with all the information about the event, and a preformatted link that will create the listing record. (see the recipe: USING AN EMAIL TEMPLATE TO CREATE A NEW ACCOUNT RECORD)
The admin reviews the information, and if it’s appropriate, clicks the link to create a record.
When they save the record (with an 'approved' check box automatically checked) the modified plugin automatically sends an email to the person who listed the event that they should email an event image to the admin.

Here's the original plugin code:

```
<?php


addAction('record_postsave',  'emailOnApproved_sendPassword', null, 4);

//
function emailOnApproved_sendPassword($tableName, $isNewRecord, $oldRecord, $recordNum) {
  global $CURRENT_USER, $SETTINGS;
  $fieldname = 'approved';

  // error checking
    if ($tableName != 'accounts') { return; }
  if (!array_key_exists($fieldname, $CURRENT_USER)) {
    die(__FUNCTION__ .": You must create an accounts fields called '$fieldname'!");
  }

  // send email
  $wasChecked   = intval(!$oldRecord[$fieldname] && $_REQUEST[$fieldname]);
  $wasUnchecked = intval($oldRecord[$fieldname]  && !$_REQUEST[$fieldname]);

$message=<<< __TEXT__
    Welcome!

Your subscription has been  processed successfully and you now have access to the Members Only area of our web site.
Your user name is: {$_REQUEST['username']}
and your temporary password is: {$_REQUEST['password']}
Once you have successfully logged in, you can change your password and update your profile information.


<a href="http://www.your_website_URL.com{$GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL']}">Click here to login</a>

Best,

The Subscription Committee
__TEXT__;


$headers  = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
$headers .="FROM:". $SETTINGS['adminEmail'];

   if ($wasChecked) {
    $errors = mail($_REQUEST['email'],"Your membership has been successfully processed!",$message,$headers);
    if ($errors!=1) { die("Mail Error: $php_errormsg"); }
  }

}

?>
```

and here's the code after Jeff's suggestions:
It assumes a multi-
record table called: 'your_table_name', a check box field called 'approved', a first name field called 'first_name', and an email field called 'email'.
note that all references to $CURRENT_USER have been removed since they are related to the website membership plugin.

```
<?php


addAction('record_postsave',  'emailOnApproved_sendPassword', null, 4);


//

function emailOnApproved_sendPassword($tableName, $isNewRecord, $oldRecord, $recordNum) {
  $fieldname = 'approved';

  // error checking
    if ($tableName != 'your_table_name') { return; }

  // send email
  $wasChecked   = intval(!$oldRecord[$fieldname] && $_REQUEST[$fieldname]);
  $wasUnchecked = intval($oldRecord[$fieldname]  && !$_REQUEST[$fieldname]);

$message=<<< __TEXT__
    Hello {$_REQUEST['first_name']},

Your event has been posted, but a compelling picture would make your listing really stand out.
You can send us an image by replying to this email with your image attached.

Please note, images should be in .jpg format and no larger than 1mb in size.

Best,

The Listing Team
__TEXT__;

$the_from = 'your_from_email@your_site.com';
$the_to = $_REQUEST['email'];
$the_subject = 'Your Subject Line';
$headers  = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=uft-8' . "\r\n";
$headers .="From:". $the_from;

   if ($wasChecked) {
    $errors = mail($the_to,$the_subject,$message,$headers);
    if ($errors!=1) { die("Mail Error: $php_errormsg"); }
  }

}

?>
```

CAVEATS FOR PLUGIN MODIFICATION - Aug 29th, 2011

Although it's possible and sometimes very handy to modify your plugins to pull data from other editors, It's important that there are no spaces, blank lines, or other code before the code:

```
<?php
```

If you want to add any load records calls, add:

```
?>
<?php
```

after the code above, and insert the calls in a new set of php tags.

So your code might look like:

```
<?php


?>
<?php

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

  // load records
  list(your_tableRecords, $your_tableMetaData) = getRecords(array(
    'tableName'   => 'your_table',
    'where'       => whereRecordNumberInUrl(1),
    'limit'       => '1',
  ));
  $your_tableRecord = @$your_tableRecords[0]; // get first record

?>
<?php

the rest of your plugin code with modifications...
```

You can learn more about how to modify a plugin in the recipe about the SHOW HIDE FIELDS FOR USERS PLUGIN

RSS FEED FOR PLUGIN AND ADD ON UPDATES - Dec 29th, 2018

Use this RSS feed to be kept up to date on new information about both Official and User submitted plugins and add ons.

http://www.interactivetools.com/add-ons/rss.php

THE AUTO BACKUP PLUG IN - Aug 6th, 2010

GIVING YOUR CLIENTS A SECURE FEELING WITH AUTOMATIC BACKUPS
With the Auto Backup plug-in from the Interactive Tools App Store

http://www.interactivetools.com/add-ons/

there’s now a simple way for you or your clients to schedule automatic daily, weekly, and monthly database backups for CMS Builder.

New in Version 1.01
- Automatic hourly backups are now supported (up to 24 per day)
- Email Backups: Enter your email address to receive automatic backups via email
- Email Backups: Email backup files will automatically be compressed if supported by your server

Backup timers will automatically be set/reset each time the Admin or section editors are accessed or someone accesses your pages on the web. By default, backups will be kept, for 3 time periods. Daily backup for 3 days, Weekly backup for 3 weeks,. and monthly backups for three months. The hourly default is 24 hours

You can also specify an e-mail address to receive backups and the frequency that those e-mails will be sent.

You can modify the code to change these default values. Backups are automatically saved on your server where you can download them to your local computer for safe keeping.

Auto Backup is extremely easy to set up. All you need to do is upload the plug-in file to your server and activate the plug-in through your CMS Builder.

CAUTION: Auto Backup only backs up your MySQL data and creates snapshots of the database that can only be restored using THE SAME VERSION of CMS Builder with which they were created.

FTP BACKUP - The programming gurus at Interactive Tools are wrestling with the complexities of adding automatic FTP backups to the plug-in so you can have a copy of these files on a local computer for safe keeping, but the hurdles are high and it may be a while before a flexible enough system can be developed. Until then you might consider offering a backup service to your clients. Both WS-FTP Pro and CuteFTP offer automatic backup scheduling and you could use it create a new profit center for your business. Just don’t forget to add a legal disclaimer to the contract with your clients if you haven’t already done that. The one that Interactive Tools uses reads:

LIMITED WARRANTY
Although Vendor has tested this Software and reviewed the documentation, Vendor makes no warranty or representation, either expressed or implied, with respect to this Software, its quality, performance, merchantability, or fitness for a particular purpose. As a result, Software is licensed AS-IS, and you are assuming the entire risk as to its quality and performance.

Vendor shall not be liable for any claim or right to recover damages, including, but not limited to, loss of profit, data, or use of the software or special, incidental, or consequential damages, or other similar claims, even if Vendor has been specifically advised of the possibility of such damages.

I asked Dave Edis at Interactive Tools for clarification on a few issues:

Me: If I haven’t accessed my CMS interface in months and the auto delete is set for 3 weeks, will I find any backup file to restore?

Dave: Yes, assuming you had one page view per day you'd have snapshots for the last 3 days, last 3 weeks, and last 3 months.

Me: If I manually create a backup in admin > general settings, will that be automatically deleted as well?

Dave: No, the auto-backup files use a different naming scheme and won't erase any of your manually generated backups.

Me: If I check my database and find that it’s missing will the restore function attempt to recreate it for me using the info for IP address, password and user name in General Settings

Dave: Yes, assuming everything is corrupted you may need to do a new install. But autobackup creates the same type of backup file as clicking backup would.

Observation: Seems like it’s a good idea to create a manual backup when you set up your plug-in. That way you’ll have a baseline backup that you can restore if necessary.

MAKING THE AUTOBACKUP PLUGIN RUN AUTOMATICALLY - Dec 29th, 2018

It seems that the auto backup plugin doesn’t run in the background on a server. Which means that no backups will be run unless, according to Chris Waddell at Interactive Tools, “...something hits admin.php periodically.”

You can read the entire thread at:

http://www.interactivetools.com/iforum/Products_C2/CMS_Builder%3A_Plugins_%26_Add-ons_F40/P84078

Chris offered a few solutions.

1) Set up one of the many Web Cron services to hit your admin.php page, allowing Auto Backup to do its thing. You won't need to provide a username/password or anything. Also, Web Crons tend to be very easy to configure!

User kblm suggested that to keep your backups from getting a prefix of “no-hostname” when running a cron, you can enter a file prefix on the line:

```
$GLOBALS['AUTOBACKUP_DEFAULT_PREFIX'] = ''; // Backup file prefix, leave blank to use current domain name
```

2) Changing the line in the plugin below will cause a normal site visit to trigger automatic backups:

```
addAction('init_complete', 'autoBackup_updateBackups',  null, 0);
```

to

```
autoBackup_updateBackups();
```

Chris noted that this might cause some extra strain on your server if you get a lot of visitors.

User aev suggested:

“If we have 1000 pageviews/hour wouldn't that require only a simple 'if' check or something similar for 999 of those? And then pageview #1000 triggers the backup?“

THE WEBSITE SAVED SEARCHES PLUGIN - Dec 9th, 2010

This handy plugin works in conjunction with the Website Membership plugin and lets your visitors name and save links to their favorite pages to a list, and to manage that list. The plugin is designed to work only if the visitor has a membership account and is logged in, and will only show that visitors list entries.

It’s pretty easy to implement by following the instructions in the included readme file, but as always, there are some details worth noting.

The first thing that I found was that the path listed in the code to be inserted in the head of your viewer for the “websiteSavedSearches.js” assumed that it was to be found in a “plugins/websiteSavedSearches” sub-directory. If this is not your situation, you’ll need to adjust the path.

The next thing was that although the jQuery script called was version 1.3.2, the jQuery script included with CMSB V2.06 is version 1.4.1. For now you’ll have to manually adjust the code so that the versions match.

Note that this <head> code block must be included on every page that uses the “Saved Searches” plugin.

Here’s how I implemented the plugin for the CMSB Cookbook recipe detail page:

Since the “/lib/viewer_functions.php” file was already loaded on that page, all I had to do was insert the previously modified <script> code block just after the <head> tag, and above any other <script> tags:

```
 <!-- saved searches -->
  <script type="text/javascript" src="/cmsAdmin/plugins/websiteSavedSearches.js">
</script>
```

```
  <script type="text/javascript" src="/cmsAdmin/3rdParty/jquery/jquery1.4.1.js"></script>
  <script type="text/javascript">jQuery.noConflict();</script>
  <?php $GLOBALS['WSS_SCRIPTS_LOADED'] = true; ?>
  <!-- /saved searches -->
```

Then in the body where I wanted to display the “Save”, “Remove” and “View My Saved Searches” page links, I used this code:

NOTES:
1) My “View My Saved Searches” page is called saved_searches.php
2) I added “AndReload” to both of the “onclick” commands as suggested in the readme file, so that the page would automatically reload to show the “Remove This Recipe From My Saved Searches” link after a page was saved.
3) You can remove the class="<?php wss_cssClassFor('add'); ?
>" code from the div tags, but don’t remove the
style="<?php wss_displayStyleFor('add'); ?>" code, or both links will show on your page.

```
<div class="<?php wss_cssClassFor('add'); ?>"
       style="<?php wss_displayStyleFor('add'); ?>">
    <a class="special" href="#" onclick="<?php wss_onClickFor('addAndReload'); ?
>">Save This Recipe To My "Saved
Searches" List</a>
  </div>

  <div class="<?php wss_cssClassFor('remove'); ?>"
       style="<?php wss_displayStyleFor('remove'); ?>">
    <a class="special" href="#" onclick="<?php wss_onClickFor('removeAndReload'); ?
>">Remove This Recipe From My "Saved
Searches" List</a>
  </div>
  <div ><a class="special"href="saved_searches.php" target="_blank"><br />
    View My "Saved Searches" List</a></div>
```

On the saved_searches.php page, since the “/lib/viewer_functions.php” file was loaded, I inserted the following code into the code at the top pf the viewer page:

```
 // load saved searches
    list($savedSearches, $savedSearchesMetaData) = getRecords(array(
      'tableName'     => '_website_saved_searches',
      'where'         => "createdByUserNum = '" .mysql_escape( @$CURRENT_USER['num'] ). "'",
      'loadCreatedBy' => false,
      'allowSearch'   => false,
    ));
```

In the <head> section, I added the <script> calls as before:

```
 <!-- saved searches -->
  <script type="text/javascript" src="/cmsAdmin/plugins/websiteSavedSearches.js">
</script>
  <script type="text/javascript" src="/cmsAdmin/3rdParty/jquery/jquery1.4.1.js"></script>
  <script type="text/javascript">jQuery.noConflict();</script>
  <?php $GLOBALS['WSS_SCRIPTS_LOADED'] = true; ?>
  <!-- /saved searches -->
```

Then in the body of the page, I inserted the following to show the “Saved Searches” list:

```
 <?php if ($savedSearches): ?>
    <table width="737" border="0" cellspacing="0" cellpadding="0"><tr>
      <td width="79%" class="Medium-Text-Bold">Click On An Entry To View</td>
      <td width="21%"  class="Medium-Text-Bold">Delete From List?</td></tr>
 <?php foreach ($savedSearches as $record): ?>
 <tr>
   <td colspan="2"><hr /></td>
   </tr>
 <tr>
    <td > <a class="saved" href="<?php echo $record['url']; ?>"><?
php echo htmlspecialchars($record['name']);
?></a></td>
    <td ><a class="Small-Text-Bold" href="#" onclick="<?
php wss_onClickFor('removeAndReload', $record['url']);
?>">Delete Search</a></td>
  </tr> <?php endforeach ?>
</table>

  <?php else: ?>
    <div align="center"><span class="Medium-Text
    <div align="center"><span class="Medium-Text-
Bold">You haven't saved any searches so far.<br />But when you do,
they'll be listed here.</span></div>
  <?php endif; ?>
```

I've posted a note on the forum about the jQuery version issue, and will update this when I find out more.

RESET THE VALUE OF A FIELD IN ALL RECORDS OR IN FIELDS IN MULTIPLE TABLES (CHECK BOXES ONLY) - Nov 11th, 2011

And here's another useful plugin from Carl Crowder at Interactive Tools called fieldResetter. This one does exactly that. It resets the all the values of one or more fields in one or more tables globally. I'm using it to reset a field in a membership database called paid_for_current_year, which needs to be reset at the end of each year.

After implementing the plugin, you'll find the option to execute the global change in the "Advanced Commands" pull down menu at the bottom of the record list for that section editor.

You can download this plugin from:

http://www.thecmsbcookbook.com/downloads/fieldResetter.zip

Implementation is really simple.

If you want to globally reset the value of a check box field called "paid_for_current_year" to "0" in a table called "members", you'd adjust the array on line 15 to:

```
$GLOBALS['FIELD_RESET_FIELDS'] = array(
  array('Table name' => 'members', 'fieldName' => 'paid_for_current_year', 'resetValue' => 0 ) );
```

If you have 2 fields to be reset, you'd adjust the array to:

```
$GLOBALS['FIELD_RESET_FIELDS'] = array(
  array('Table name' => 'members', 'fieldName' => 'paid_for_current_year', 'resetValue' => 0 ),
  array('Table name' => 'members', 'fieldName' => 'other_field', 'resetValue' => 0 ) );
);
```

For 2 tables:

```
$GLOBALS['FIELD_RESET_FIELDS'] = array(
  array('Table name' => 'members', 'fieldName' => 'paid_for_current_year', 'resetValue' => 0 ),
  array('Table name' => 'other_table', 'fieldName' => 'other_field', 'resetValue' => 0 ) );
```

Let’s say you wanted to give permission to accomplish this to a “Super Admin only, you could add a check box field to your Accounts section called 'super_admin' and change the check on line 27 to

```
if ( !$CURRENT_USER['super_admin'] ) return;
```

If you make the “super_admin” field a system field, it will be harder to inadvertently change it’s value.

RESET THE VALUE OF A FIELD IN A MULTI RECORD TABLE (DATE FIELDS ONLY) - Aug 29th, 2014

My client needed to update the value of a date field in all of the records in a multi-record table, where the hidden
field value was “0" (not hidden) with a specific date.

Here’s the approach that we used. The date must be in the format YYYY-MM-DD

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

    // load viewer library
  $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
   $dirsToCheck = array('/path_to_your_root_directory/','','../','../../','../../../');
  foreach ($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 'your_table'
  list($your_tableRecords, $your_tableMetaData) = getRecords(array(
    'tableName'   => 'your_table',
    'loadUploads' => true,

  ));

?><!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" />
<title>access-update2</title>
</head>

<body>
<?php foreach ($your_tableRecords as $record): ?>
<?php
    mysql_query("UPDATE `{$TABLE_PREFIX}your_table`
    SET  expiresDate = '2015-04-01'
    WHERE hidden = '" . 0 . "'"
       )
       or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
        $userNum = mysql_insert_id();

?>
<?php endforeach ?>
  </body></html>
```

REMOVE EXTRA WHITE SPACE AND BLANK DATE PLUGIN - 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

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.

UPDATED VERSION
The latest 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. It is compatible with CMSB version 2.53 and above

You can download this version of the plugin from:

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

THE CREATEPDF PLUGIN - Nov 16th, 2019

The create PDF can be extremely useful to generate PDF documents from web pages or URLs, and display them as in line content or as attachments.

Here’s the step by step recipe on how to install this plugin.

2) Upload the complete createPDF folder including the 2 subfolders included in the plugin (“wkhtmltopdf” and “examples”) to the cmsAdmin/plugins folder on your server.

3) Activate the plugin and click on "Test Server Requirements". If you encounter errors, there are some required server libraries listed in the readme file that came with the plugin, and you may have to work with your server admin to get them up and running. If you still have issues, post the specifics on the plugins forum and IT will help to resolve them.

NOTE: I had to adjust the server path to my viewer_functions.php by removing one of the ../ from the path. I also encountered a “No input file specified” error in the PDF output test, but that was resolved by Dave’s first suggestion below.

4) After you’ve eliminated all the errors, test some of the examples in the examples folder

5) You can run the example code from viewers in another directory, but you’ll have to adjust the path to your viewer_functions.php file

When I ran the example code, the URL as inline opened as a web page in FF(3), IE(8), Safari (4), or Chrome (7) and look really good. If I ran the URL as attachment code, it downloaded a PDF that also looked good.

No matter what I tried, I could not display an html or output PDF, either as inline or as attachment. The results all showed "no input file specified" as the only text in the resulting PDFs.

Dave Edis from Interactive Tools came to the rescue. He said,

“The plugin creates temp .html files and your server was redirecting *.html to *.php so the plugin couldn't find it's temp .html files. Add this line to your .htaccess so requests will only be rewritten if the .html file doesn't exist:”

```
RewriteEngine on
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)\.html$ $1.php [nc]
```

Another issue that I came across (V 1.09 - 2019) was that if any of the links on my page were broken, I'd get a 'content not found' error when I sent the convert to pdf request.

The next issue involved creating a PDF from a URL that was protected by a membership plugin password, all I could get was a PDF of the login screen, even if I was logged in.

Again, Dave came to the rescue, he reminded that in the advanced topic section of the readme file about creating a PDF of the page that’s currently being viewed in a browser. It suggests:

To create a link that shows the current page as a PDF do the following:

LIST VIEWER
First, create a link on your list viewer to the current page, but with ?pdf=1 as the url. Example:

```
<a class="your_css_class" href="?
pdf=1">CLICK TO CONVERT THIS PAGE TO A PDF</a>
```

Another anomaly (this one on Bluehost ising V 1.09 in 2019), I found that if I was trying to convert an index.php page to aPDF, I'd have to add index.php to the link above, or I'd get content not found errors:

```
<a class="your_css_class" href="index.php?
pdf=1">CLICK TO CONVERT THIS PAGE TO A PDF</a>
```

DETAIL PAGE
This gets a bit more complicated. You'll need to call the specific page and record number, like this:

```
<a class="your_css_class" href="http://www.your_site_URL/your_detail_page.php?pdf=1&
<?php echo $your_tableRecord['num']
?>">CLICK TO CONVERT THIS PAGE TO A PDF</a>
```

Next, for either, add this code to the top of the viewer after the get records code block:

```
 <?php if (@$_REQUEST['pdf']) { createPDF_fromOutput('inline', 'example.pdf'); } ?>
```

Another anomaly (this one on Bluehost in 2019). I found that if I was trying to convert an index.php file, I'd normally just use the link:

ONE (OR MORE) OF MY SERVERS WON'T SUPPORT THE PLUGIN
If you have site(s) (client(s)) on a server or servers that don't meet the requirements of this plugin, but you do have even one server that does support them, you could use that server as a “proxy server” (host) to do all of the heavy lifting required. (V1.01 and above) This approach will work with one "host" and any number of "clients".

Here’s how to set up the "host", "client" relationship:

1) Upload the createPDF plugin as above, and enable the plugin on both the site that you’ll be using as a server (host) and the site(s) that needs to be “served” (client)

2) In the createPDF.php plugin on the “host” set the values on these two lines:

```
$GLOBALS['CREATEPDF_PROXY']          = true;
$GLOBALS['CREATEPDF_PROXY_PASSWORD'] = 'enter_your_own_password_here';
```

NOTE: The password can be anything you like, just make sure that it is identical on the “host” and on any “client” sites.

3) In the createPDF.php plugin on the “client(s)” set the values on these two lines:

```
$GLOBALS['CREATEPDF_PROXY']          =
‘http://url_to_your_host_server/cmsAdmin/plugins/createPDF/createPDF_proxy.php’ ;
$GLOBALS['CREATEPDF_PROXY_PASSWORD'] = 'enter_your_own_password_here';
```

4) When you click on "Test Server Requirements", in the cmsAdmin/plugins menu on the "client" site, you should see an indication that you are "Using Proxy" and, if you haven't made any errors when typing in the required information in steps 2 and 3, you should see "Passed" next to the "Test downloading content from remote URL" and Test downloading content from proxy URL" tests.

After that, any time you implement a create pdf request on the "client" it should execute as planned.

CHANGE THE COLOR OF PAGE BACKGROUND TO WHITE FOR THE PDF

Dave Edis offered this idea..

He said, since the page is reloaded after the request to create a PDF is executed, you could replace the opening body tag with something like:

```
<?php if (@$_REQUEST['pdf']): ?>
   <body style="background-color:#FFFFFF">
<?php else: ?>
   <body style="background-color:#569F00">
<?php endif ?>
```

HIDING THE "CLICK TO CONVERT THIS PAGE TO A PDF" LINK ON THE PDF

You could also replace the createPDF execution link with something similar, like this for a list page:

```
<?php if (@$_REQUEST['pdf']): ?>&nbsp;<?php else: ?>
<a class="your_css_class" href="?pdf=1">CLICK TO CONVERT THIS PAGE
TO A PDF</a><?php endif ?>
TO A PDF</a><?php endif ?>
```

Or, for a detail page:

```
<?php if (@$_REQUEST['pdf']): ?>&nbsp;<?php else: ?><a class="your_css_class"
href="http://www.your_site_URL/your_detail_page.php?pdf=1&<?
php echo $your_tableRecord['num'] ?>">CLICK TO CONVERT THIS
PAGE TO A PDF</a>
<?php endif ?>
```

You can also add some parameters to the end of the convert code to make the PDF open in specific ways.

To make the reader open at 100% magnification, with scroll bars, a toolbar and a navigation pane, I used:

```
<a class="small" href="http://50.6.159.105/about.php?pdf=1&<?
php echo $resumeRecord['num']
?
>#toolbar=1&navpanes=1&scrollbar=1&zoom=100">CLICK TOCONVERT THISPAGE TO A PDF</a>
```

You can finds other useful parameters in the PDF at:

http://www.thecmsbcookbook.com/downloads/pdf_open_parameters_v9.pdf

CAVEATS:
Be careful when using in line styles to change font sizes. They can wreak havoc with rendered PDF.

I thought that I was being very cool and used a special class to change the font size in some links when displayed in the PDF and spent the better part of a day finding out that this was the cause of my PDF font sizes coming out much smaller than I wanted them to.

THINGS LEARNED FROM IMPLEMENTING THE createPDF PLUGIN ON THE COOKBOOK SITE

Aside from the information above, I found out a few things that were specific to my implementation. I was using highlight_string and was having issues with links in the PDF that were offset from the text that they were supposed to be connected to.

I was originally using

```
<?php
$recipe = wordwrap($record['recipe'],140) ;
$recipe = highlight_string($recipe) ;

?>
```

to render the recipes on my list pages.

and for my detail pages.

```
<?php
$recipe = wordwrap($table_of_contentsRecord['recipe'],140) ;
$recipe = highlight_string($recipe) ;

?>
```

Dave Edis took a stab at solving the issue. He said:

It looks like highlight_string() replaces all the spaces with &nbsp;. My guess is something about the long lines was causing problems with the PDF conversion.

This code will switch them back to spaces:

```
<?php
$recipe = $record['recipe'];
$recipe = highlight_string($recipe, true);
$recipe = str_replace('&nbsp;', ' ', $recipe);
$recipe = wordwrap($recipe, 140, '', true);
print $recipe;
?>
```

I needed to modify things like this to get the detail pages to render correctly:

```
<?php
$recipe = $record['recipe'];
```

```
$recipe = $record['recipe'];
$recipe = highlight_string($recipe, true);
$recipe = str_replace('&nbsp;', ' ', $recipe);
// $recipe = wordwrap($recipe, 140, '', true);
print $recipe;
?>
```

User MickC offered this code to rename the PDF that's created to something more meaningful then example.pdf.

In it's current form it creates the PDF from "your_detail_page" as an attachment and the creates file name for the PDF from the value of "field_1" and "field_2" of the table "your_table". To create the document as an inline PDF, just change 'attachment' to 'inline' in the last line of the code.

```
<?php require_once "lib/viewer_functions.php"; ?>
<?php
  $filename= $your_tableRecord['field_1']."-".$your_tableRecord['field_2'];
  $num= $your_tableRecord['num'];
  $url  = "your_detail_page.php?".$num;
  $data = createPDF_fromUrl($url);
  createPDF_display('attachment', $data, $filename.'.pdf');
?>
```

You can download a PDF test page that implements some if statements to change various parameters on the resulting PDF from:

http://www.thecmsbcookbook.com/downloads/pdftest.zip

My next issue (as yet unresolved) was using named anchor tags within the pages and having them operate correctly in the rendered PDF.

I'll post a solution as soon as I have one (or one of my subscribers sends one along).

CREATING PDF'S FROM VERY LARGE PAGES - Jan 6th, 2015

As the cookbook grew to well over 400 recipes (records) in late 2014, conversion to PDF became a problem. Server memory allocation issues were keeping the PDFs from being created.

Dave Edis from Interactive Tools came up with a simple, elegant solution for creating multiple PDFs, each containing a set number of recipes (records) , and displaying a paging menu at the top of the viewer.

At the top of the viewer, the load records call contains a conditional statement that determines when perPage limitations are implemented. In this case when there is a request for a PDF to be created. (The 'loadCreatedBy' => false, reduces the server load slightly). There’s also a createPDF_fromOutput instruction.

```
 if (@$_REQUEST['pdf']) {
   list($table_of_contentsRecords, $table_of_contentsMetaData) = getRecords(array(
     'tableName'     => 'table_of_contents',
     'perPage'       => 100,
     'loadCreatedBy' => false,
   ));
  }
  else {
   list($table_of_contentsRecords, $table_of_contentsMetaData) = getRecords(array(
     'tableName'     => 'table_of_contents',
     'loadCreatedBy' => false,
   ));
  }

   if (@$_REQUEST['pdf']) { createPDF_fromOutput('inline', 'example.pdf'); }
```

Then in the body, this code changes the background color of the page if a PDF is requested:

```
<?php if (@$_REQUEST['pdf']): ?>
    <body style="background-color:#FFF">
<?php else: ?>
    <body style="background-color:#9FF">
<?php endif ?>
```

And the code that creates a dynamic paging menu:

```
 <?php
  $recordCount    = mysql_count('table_of_contents');
  $recordsPerPage = 100;
  $pageCount      = 1;

  print "We now have $recordCount recipes!\n";

  while (($pageCount * $recordsPerPage) < ($recordCount + $recordsPerPage)) {
    $recipesLow  = sprintf("%03s", ($pageCount-1) * $recordsPerPage + 1 );
    $recipesHigh = min($recordCount, ($pageCount * $recordsPerPage) );

    print "Section $pageCount (Recipes $recipesLow - $recipesHigh)";
    print " - <a href='cookbook.php?page=$pageCount'>View HTML</a>";
    print " - <a href='cookbook.php?page=$pageCount&amp;pdf=1'>Download PDF</a>";

    $pageCount = $pageCount + 1; // show next page
  }
?>
```

CREATING PDF'S FROM SEARCH RESULTS - Nov 18th, 2019

I had a rather complex search form and needed to create a PDF of the search results so they could be easily downloaded and printed.

Daniel Louwe from Interactive Tools came to the rescue (again) with the following suggestion:

"Because you want to create a PDF of the search results, you need to simulate the form being submitted at the same time as the PDF generation. Clicking the link with "?
pdf=1" clears all of the search values, as well as the "save" value that
tells the page to show the search results.

One fairly minimal solution I can think of is instead of using a "convert this page to PDF" link, you could add a checkbox to the form with the name "pdf" and value "1" labelled something like "View search results as a PDF" so that the user can choose to receive a PDF of the results instead of a web page when they submit the form, rather than clicking a link afterwards."

For example, the final code in my form was

```
<tr>
              <td align="right" class="text_font">&nbsp;</td>
              <td align="left" valign="bottom" colspan="2"> <input type = "checkbox" id="pdf" name="pdf" value = "1"
<?php checkedIf(1,
@$_REQUEST['pdf']);?> /> <span class="text_font">
<b>Check to create a downloadable PDF of the search results instead of
a web page</b></span></td>

            </tr>
```

Daniel went on to suggest:
"Another simple - but more tedious - solution is to add "save=1" to the URL in addition to the relevant form values. A simple working example is **http://your_site.com/your_search_page.php?pdf=1&save=1&your_search_field_keyword[]=1**. The two

main tasks you'd have are 1) building the URL string based on the submitted form values, and 2) updating the code to account for when both $_REQUEST['pdf'] and $_REQUEST['save'] are on; e.g. removing the original "convert this page to PDF" link."

Note: The brackets [] after _keyword are because in my form this field in the table that I'm searching got its values from a pulldown list that accessed second table, and so this was that field's name in the form.

THE SAVE AND COPY BUTTON PLUGIN - Dec 29th, 2018

This free plugin form Dave Edis at Interactive Tools adds a new "Save & Copy" button to edit pages for all sections. The "Save & Copy" button copies both a record and its uploads.

This new button can be clicked to instantly create a copy of the current record.

This new record will not be saved until you click the save button (or click "Save & Copy" again).

Note: There is a known issue where if you click "Save & Copy" on a record, and then try to upload an image right away, the image doesn't appear and it actually adds it to the original record. To get around this just click "Save" and then "Modify" the new record again. IT says it hopes to address this limitation in a future release.

You can download the free plugin from:

http://www.interactivetools.com/add-ons/?user_submitted=1

THE MODIFY HOME PAGE PLUGIN - Dec 29th, 2018

This is a great little plugin created by Chris Waddell at Interactive Tools. It allows you to easily add a heading and a "quick pick" list to the page that a user sees (in addition to the usual navigation menu) when they first log on to CMS Builder.

To customize the plugin and make it ready fro implementation, you'll want to change some things and remove a few lines in the code.

First, find the list:

```
<ul>
     <li><a href="?menu=news&action=add">Add a News Item</a></li>
     <li><a href="?menu=jobs&action=add">Add a Job Listing</a></li>
     <li><a href="?menu=events&action=add">Add an Event</a></li>
   </ul>
```

This needs to be modified to reflect the tasks and editors that are appropriate for your site.

So you might change:

```
<li><a href="?menu=news&action=add">Add a News Item</a></li>
```

to:

```
<li><a href="?menu=projects&action=add">Add a project to your "project" list</a></li>
```

You can add as many items as you like.

The items in the list could also be links to external sites. (don't forget the http://)

```
<li><a href="http://www.yoursite.com">Visit my web site</a></li>
```

You'll also want to change the "To modify this title, find and change this text in the Modify Homepage plugin!", that's used as a title for the page to something more appropriate, or remove the text altogether.

You'll also want to remove the:

```
<p>
<strong>To modify this content, find and change this text in the Modify Homepage plugin.
</strong></p>
```

If you get adventurous, you can change the items (or the entire list) to show only for specific users classes.

Add some check boxes to your user accounts section and call them similarly to the way the plugin checks for admin status:

```
<?php if ($CURRENT_USER['isAdmin']): ?>
<p><b>Administrators:</b> Use the <a href="?
menu=database">Section Editors</a> to add sections and generate PHP
viewers.</p>
    <?php endif ?>
```

For example, you might create a check box field called "project" and use this code to display some code only if the check box is checked :

```
<?php if ($CURRENT_USER['project'] == 1): ?>
Some code...
<?php endif ?>
```

Tim Forrest from Toledoh Enterprises suggested adding a frame to display the free Pixlr image editor as part of the page.

```
<iframe id="pixlr" type="text/html" width="100%" height="600" src="http://pixlr.com/express/" frameborder="0">
</iframe>
```

I love the idea, but preferred to make it a link at the bottom of my list, by adding:

```
<br />
      <li>
<a href="http://pixlr.com/express/" target="_blank">Quickly crop or edit an image using the free Pixlr Express
image editor</a></li>
```

To further customize this plugin, you might want to pull data directly from a CMSB Section editor.

I set up a single record editor called "Quick Start Guide" with fields for "guide" information, and "resources". I wanted to show that information on the CMSB Interface's "Home Page" as well as in a separate viewer that could be accessed at all times.

Here's How I did that:

After the code

```
function modifyHomepage_content() {
  global $CURRENT_USER;
  ?>
__
```

I added a standard // load viewer library call

```
  $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."); }
__
```

Then I added the get records call

```
<?php
  // load records
  list($quick_start_guideRecords, $quick_start_guideMetaData) = getRecords(array(
    'tableName'   => 'quick_start_guide',

    'limit'       => '1',
  ));
  $quick_start_guideRecord = @$quick_start_guideRecords[0]; // get first record

?>
__
```

Then I added the code required to display my data.

```
<span class=" heading-text-bold">Welcome <?
php echo mysql_escape($CURRENT_USER['first_name']); ?> <?php echo
php echo mysql_escape($CURRENT_USER['first_name']); ?> <?php echo
mysql_escape($CURRENT_USER['last_name']); ?>,</span>
 <?php

  // load records
  list($quick_start_guideRecords, $quick_start_guideMetaData) = getRecords(array(
    'tableName'   => 'quick_start_guide',

    'limit'       => '1',
  ));
  $quick_start_guideRecord = @$quick_start_guideRecords[0]; // get first record

?>
 <span class="medium"><?php echo $quick_start_guideRecord['guide'] ?><?
php echo $quick_start_guideRecord['resources']
?></span>

<?php endif ?>
<span class=" medium">



      <a href="http://your_web_URL/listing.php" target="_blank">
<b>Click To View All Of Your Records</b></a>



  <b>This &quot;Quick Start Guide&quot; will always be available from the link on the menu on the left side of your
screen.</b></span>
```

I enabled a plugin called exampleHeaderLinks to add the link to my Quick Start Guide to the menu on the left side of the CMSB Interface.

You can download a copy of that plugin from:

http://www.thecmsbcookbook.com/downloads/exampleHeaderLinks.zip

THE GREY HIDDEN RECORDS PLUGIN - Dec 29th, 2018

This free plugin from Chris Waddell at Interactive Tools changes the color of records in the CMS Admin menus to Grey if their "hidden" field is checked. It has no effect on sections without a hidden field.

You can download the free plugin from:

http://www.interactivetools.com/add-ons/?user_submitted=1

THE SHOW HIDE FIELDS FOR USERS PLUGIN - Oct 15th, 2011

This very handy plugin was written by Chris from Interactive Tools allows customization of which fields appear in an editor depending on the current user's 'skill_level' field.

The first thing that you'll need to implement this plugin is a new single value pull down list field in the User Accounts (accounts) editor called skill_level. You can name it anything that suites your application because you'll reference that field in the plugin.

Decide on the skill levels you'll need for your application and enter them as the values to be chosen from the pull down list for each user account.

For this example there are 2 skill levels, Beginner and Expert.

If you want to manage the fields in a section called "my_section" you'll want to have a list of the field names in that section handy.

Let's assume you've got 4 fields named beginner_field_1, beginner_field_2. expert_field_1, and expert_field_2

Next open the plugin and insert the name of your skill level field in the code:

```
$GLOBALS['BETA_USER_SKILL_FIELDS_SKILL_FIELD_NAME'] = 'skill_level';
```

In in the code below, (remember, my_section refers to the section where you want to show or hide fields), both expert fields will be hidden from any user whose skill_level is "Beginner". You can add as many skill levels as you like, but the last one has no trailing comma.

```
$GLOBALS['BETA_USER_SKILL_FIELDS_CONFIG']['my_section'] = array(

  // user's skill values and lists of fields to hide
  'Beginner' => array('expert_field_1', 'expert_field_2'),
  'Expert'   => array()

);
```

Repeat this code block as many times as necessary and replace "my_section" with the name of the editor that you want to

Repeat this code block as many times as necessary and replace "my_section" with the name of the editor that you want to affect.

PULLING ARRAY VALUES FROM ANOTHER EDITOR
Here's where the fun begins. Set up this plugin so that your clients can change the hidden fields for the various skill levels without needing to dive into the back end of CMSB and mucking up the works.

Here's how:

THE NEW EDITOR
Create a single record editor with a multi-
value check box list field for each skill level you'll need to control.
beginner_hide and expert_hide are used for this example.

Insert the field names in the table you want to show or hide as the option values for the lists.
Note: Don't forget that you can use field names followed by the "pipe" character "|" followed by a user friendly name to make your list options more readable. Like this:

```
my_first_field_name_with_lots_of_complex_characters|The Simple Name
```

Check some boxes and save the record

MODIFY THE PLUGIN
CAVEAT: Remember nothing can precede the"Plugin Name" through "Requires at least" code block.

So, insert a:

```
?>
<?php
```

after that code block:

```
<?php

?>
<?php
```

Insert your load records calls in between the ?> and the <?php you just added, like this:

```
<?php


?>
<?php

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

  // load records
  list($your_list_tableRecords, $your_list_tableMetaData) = getRecords(array(
    'tableName'   => 'your_list_table',
    'allowSearch' => false,
    'limit'       => '1',
  ));
  $your_list_tableRecord = @$your_list_tableRecords[0]; // get first record

?>
```

Instead of the code:

```
 'Beginner' => array('expert_field_1', 'expert_field_2'),
  'Expert'   => array()
```

Insert the code:

```
 'Beginner'   => explode("\t", trim($your_list_tableRecord['beginner_hide'], "\t")),
 'Expert'   => explode("\t", trim($your_list_tableRecord['expert_hide'], "\t"))
```

This will populate your arrays with the checked values in each of the check box fields.

This will populate your arrays with the checked values in each of the check box fields.

You can add as many arrays as you need to, but remember the last one has no trailing comma

Your modified code should look something like this:

```
<?php



?>
<?php

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

  // load records
  list($your_list_tableRecords, $your_list_tableMetaData) = getRecords(array(
    'tableName'   => 'your_list_table',
     'allowSearch' => false,
    'limit'       => '1',
  ));
  $your_list_tableRecord = @$your_list_tableRecords[0]; // get first record

?>
<?php

// CAVEATS:
//   - it is not currently possible to hide Separator fields.

// UPDATE THESE VALUES

$GLOBALS['BETA_USER_SKILL_FIELDS_SKILL_FIELD_NAME'] = 'skill_level';

// For each section, copy this block
$GLOBALS['BETA_USER_SKILL_FIELDS_CONFIG']['my_section'] = array(

  // user's skill values and lists of fields to hide

 'Beginner'   => explode("\t", trim($your_list_tableRecord['beginner_hide'], "\t")),
 'Expert'   => explode("\t", trim($your_list_tableRecord['expert_hide'], "\t"))

);

// DON'T UPDATE ANYTHING BELOW THIS LINE
```

IMPORTANT: Changing the values in the arrays MAY NOT affect any existing records, so you'll need to test by creating NEW records to see which fields are actually hidden or shown.

To see what values are actually being inserted into your arrays, you can add:

```
showme($GLOBALS['BETA_USER_SKILL_FIELDS_CONFIG']
['master_subscription_pages']);
```

Just above the // DON'T UPDATE ANYTHING BELOW THIS LINE

And when you log into your CMSB admin Interface you'll see the array values superimposed over your interface. View the source of that page and you'll see the values much more clearly.

When you're done, remove or comment out the showme line with double slashes (//), or your users will see the superimposed values as well.

HIDING SEPARATORS
Instead of using standard separators, use HTML separators with code like these examples (the ID name of the separator is copied from your field list)

Red Headline Text

```
<tr>
 <td colspan='2'>
 <div class="content-box content-box-divider" name="__separator017__">
     <div class="content-box-header"><h3>
<font color="red">YOUR MENU INFORMATION</font></h3></div>
   </div>
 </td>
</tr>
```

or explanatory text inside a box

```
<tr>
 <td colspan='2'> <div class="content-box content-box-
divider" name="__separator016__">Your explanatory text goes
here.<br /> <br /></div>
 </td>
</tr>
```

Or explanatory text without a box

```
<tr>

 <td colspan='2'><div  name="__separator015__"><font color="red">IMPORTANT:
</font> Your explanatory text goes here. <br
/> <br /></div>
 </td>
</tr>
```

Then include each separator names in your field name list

```
__separator015__|Explanatory text
__separator017__|The Menu
```

THE CHANGE RECORDS PER PAGE DEFAULT PLUGIN - Mar 7th, 2013

One of the new features of CMS Builder (since version 2.17) is the ability select how many records are displayed on the list page of a section. If you go to the Section Editor section in the left hand menu, then click modify on a section you want to edit. At the top of the modify section page should be 5 tabs, if you select the advanced tab you should see a per page drop down that allows you to select how many records should be displayed per page.

If you want to institute a "pseudo" global approach to this issue, see the recipe called CHANGE THE DEFAULT 25 RECORDS PER PAGE ON A RECORD LIST.

And, if you're using Version 2.51 or later, and are looking for a more global change, you can use the plugin below offered by Dave Edis from Interactive Tools to change all of the default per page values to any value you need.

He suggests backing up all of your /data/schema/ files first and then try this:

Copy the code below to as blank document, name it perpagechanger.php, upload it to your plugins folder and activate the plugin under Admin>plugins.

```
<?php


### UPDATE THESE VALUES
$GLOBALS['PERPAGECHANGER_NEWVALUE'] = 1000; //

// DON'T UPDATE ANYTHING BELOW THIS LINE

// plugin menu - add link
pluginAction_addHandlerAndLink(t('Update Sections'), 'perPageChanger', 'admins');

//
function perPageChanger() {
  foreach (getSchemaTables() as $tableName) {
    $schema = loadSchema($tableName);
    $schema['_perPageDefault'] = $GLOBALS['PERPAGECHANGER_NEWVALUE'];
    saveSchema($tableName, $schema);
    alert("Updated '$tableName' to have a per page default value of {$GLOBALS['PERPAGECHANGER_NEWVALUE']}\n");
  }
}

?>
```

USING THE WEBSITE MEMBERSHIP PLUGIN

An effective membership system has a number of components. Some are listed here, if you think of others, please let me know. At this writing, accounting is a manual process, but I'm sure that some of you will have ideas along those lines. I'll share what I've done in each of these areas and expand this section as new information is available.

- Automated Signup
- Automatic/manual approval, pre-approval emails, email confirmation requirements
- Account duration
- Unlimited user "groups". Provide access to multiple directories to different members.
- "Individual" group option. Give your members access to their own private directory, unique to their username. Only they can access it.
- Customizable templates for all HTML and Email responses, including personalization fields. Example: "Hello [FIRST_NAME]"
- Mass mail members and member groups with personalization fields. Example: "Hello [FIRST_NAME]"
- New News Updates section. Important security or release information provided. Pop-up notification of important updates.
- Member failed login attempts receive easy to understand responses and tool to retrieve member login information via email.
- Automatically temporarily block large number of failed login attempts for a configurable period of time. Password-hacking protection!
- Permanently ban email addresses or entire domains which you do not wish to give membership accounts to.
- Form fields validated for accuracy required status, and formatting.
- User search by Username, First Name, Last Name, or Email Address
- Different admin assignment for different groups

ACCOUNTING
- Automated Recurring Billing with hooks to major pay services
- Day-at-a-glance sales/income totals
- Accounting Functions like account charges, billing cycle, customer edit credit card info, response to successful and unsuccessful registration attempts, automated email to remind of expiring accounts (non-recurring accounts and invalid CC info), etc.

THE WEBSITE MEMBERSHIP PLUGIN - Apr 16th, 2013

There are some interesting features built in to this plugin. I'm sure that there will be many more to come.

Here's the brief summary:

Here's the brief summary:

Website Membership works by allowing site visitors to sign-up for a user account. Once signed up they can log in to view password protected content on your website. At any time, if your visitor has forgotten their password they can request to have it emailed to them.

The changes in Version 1.08+ require some code changes in viewers to work correctly, I'll be documenting them over the next few weeks.

Version 1.10 Released 3/18/13

SERVER REQUIREMENTS: CMS v2.52

NEW FEATURES
- Security: Added blacklist of top 100k common passwords. Found in /lib/login_password_blacklist.txt
- Usability: Whitespace is removed from beginning and end of username and password on login (to prevent copy and paste errors)
- Usability: Passwords that start or end with spaces are no longer allowed.
NOTE: If upgrading, use Code Generator to recreate the Password Reset and Edit Profile pages to take advantage of new features.

MINOR CHANGES & BUG FIXES
- Misc Code and other minor improvements

Version 1.09 Released 1/29/13

NEW FEATURES
- Login Form: Users can now login with either username or email (on sites supporting usernames)
- Signup Form: Username support can be disabled by setting $useUsernames = false;
- Profile Form: Username support can be disabled by setting $useUsernames = false;
- Profile Form: Added sample commented code for adding "Agree to Terms of Service" checkbox
- Email Templates: can now be edited in CMS under: Admin > Email Templates
- Example Pages: can now be generated under: Admin > Code Generator
- Account Table: Changing account table name will cause new table to be created if it doesn't exist

MINOR CHANGES & BUG FIXES
- Enabled "Required System Plugin: True" in plugin so plugin can't be accidentally disabled
- Fixed bugs related to invalid links being generated in some instances
- Logoff now redirects back to referring page if no other url specified
- Login page now redirects back to referring page if no other url specified
- Removed files: Old email templates folder /emails/ has been removed
- Removed files: Old example files folder /examples/ has been removed
- Misc Code and other minor improvements

Version 1.08 Released 10/29/12

MINOR CHANGES & BUG FIXES
- Added a plugin hook 'wsm_loginSuccess' to allow custom code to always run after a user successfully logs in
- Added a plugin hook '_websiteLogin_init.pre_actionHandler' to support future social media login plugins
- Added new option 'WEBSITE_LOGIN_REQUIRED_FIELDS' to force users to fill out required profile fields
- Fixed bug where entire $_SESSION array would be erased when a user was not logged in

Version 1.07 Released 1/27/12

MINOR CHANGES & BUG FIXES
- On save of WSM_ACCOUNTS_TABLE, if it's not accounts, encrypt passwords if needed
- Example files now use $GLOBALS['WSM_ACCOUNTS_TABLE'] instead of 'accounts' so they recognize different accounts tablenames
- Sign-up example file no longer updates the _accesslist table if using a seperate table for website users

New in Version 1.06

NEW FEATURES
- You can now specify a separate accounts tables for website users (see WSM_ACCOUNTS_TABLE in plugin code)
- You can now login to the CMS and website as different users simultaneously (see WSM_SEPARATE_LOGIN in plugin code)

MINOR CHANGES & BUG FIXES
- Fixed bug where plugin code was being automatically run by command-line scripts
- Fixed typo on examples/user-signup.php. Changed $errors to $mailErrors on line 96

New in Version 1.05

MINOR CHANGES & BUG FIXES
- Fixed display of mail errors in email template
- Fixed issue with reset password feature causing an error

New in Version 1.04
NEW FEATURES
- Updated Plugin and sample_profile.php to support new Encrypted Passwords feature in v2.08
- Email templates are now available in /emails/

MINOR CHANGES & BUG FIXES
- You can now add false to websiteLogin_redirectToLogin(false) to prevent the user from being redirected back to the previous page after login
- Profile: You can now avoid having users logged out after changing their username and/or password by calling the new

websiteLogin_setLoginTo() function

New in Version 1.03
MINOR CHANGES & BUG FIXES

- Login: After login users are now directed to the last "login-required" page instead of WEBSITE_LOGIN_POST_LOGIN_URL
- Docs: Added section to readme.txt: How to display custom login errors (Not approved yet, etc)
- Errors: Message is now displayed for disabled accounts on login: "Your account has been disabled!"

New in Version 1.02
- Docs: Added section to readme.txt: How to add a custom text field
- Docs: Added section to readme.txt: How to add radio fields with values loaded from CMS
- Erase Account: Added new button on profile page to allow users to remove their accounts
- Misc Code and other minor improvements

New in Version 1.01
Signup: user is now emailed a random password to confirm their email address (they can't login without receiving password)
- Edit your Profile: added new form to allow user to update their profile details and change their password, and any other fields you've setup.
- Docs: Added section to readme.txt on how to use email as the username (if you don't want users making up their own usernames)

Along with other functionality that’s built in to this module, you can display certain information in your viewers on a restricted basis.

To determine if the current user is logged in, you can test with:

```
<?php if (current_user): ?>
only logged in user will see this...
<? php endif ?>
```

You could set up a separate (text or check box) field in the user accounts editor and assign it a value for that particular class of users. Then test with:

```
<?php if ($CURRENT_USER['your_field'] == "your_value"): ?
>Current user will be able to see this<?php endif ?>
```

Only user(s) who have that value entered/selected in their user record will be able to view the protected information on the web page.

Chris Waddell from Interactive Tools added these hints:

To determine if the current user is the "author" of a record, you can test with:

```
<?php if ($record['createdByUserNum'] == $CURRENT_USER['num']): ?
>Only the author will be able to see this<?php endif ?>
```

You may want to also allow admins to see your protected information:

```
<?
php if ($record['createdByUserNum'] == $CURRENT_USER['num'] || $CURRENT_USER['isAdmin']): ?
>Author and administrators
will be able to see this<?php endif ?>
```

ARE YOU TRAPPED IN A “YOU ARE ALREADY LOGGED IN” LOOP? - Aug 9th, 2020

The solution to this issue proved to be very illusive but it's now solved, and I hope that this will save someone a whole heap of time and frustration.

The one small coding change that actually fixed the issue was in the website membership plugin (V1.13) itself.

The $GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL'] had a full URL entered.

When I changed the full URL in $GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL'] = '**http://www.my-site.com/member_login.php**';
to a relative URL: $GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL'] = '/member_login.php';

The loop issue and the misdirects after login issues that my client had mentioned all disappeared.

And just to be sure, every time I changed the relative URL back to a full URL, the issue returned.

Another possibility is that you don't have a destination set in $GLOBALS['WEBSITE_LOGIN_POST_LOGIN_URL'] =

Try setting a relative URL her and see if that helps.

I'm still curious as to why that made a difference, but there you have it...

ADDING PROFILE IMAGES USING THE USER UPDATE FORM (REVISED FOR MYSQLI) - Aug 3rd, 2019

I needed to add a profile image to existing account records using the user update form that's part of the Website Membership Plugin. I also wanted to enable replacing that profile image when desired.

I was stumped until I came across this hidden gem provided by Chris way back in 2010.

Note that with a great deal of help from Daniel Louwe, 'ignoreHidden' => true was added to the get records call so that hidden records would not throw an error and I was able to revise the code to accommodate multiple profile images in the same record (My account records were for group therapy practices with more than one therapist and I wanted to keep all therapists on the same record and have a profile image for each)

*** The key is to increment all variables for each new profile image. So $current_user_with_uploads becomes $current_user_with_uploads1, then $current_user_with_uploads2. $uploadFieldName becomes $uploadFieldName1, then $uploadFieldName2 etc.

1) Add an upload field to the accounts table called profile_image and limit the maximum uploads to 2 (this allows an existing profile image to be replaced with a new one). If more than one profile image is required, add additional upload fields named profile_image_1, profile_image_2, etc.

2) Add the following to your existing form table (style as appropriate):

```
 <tr>
<td class="your_class" colspan="2"><b>FIRST PROFILE IMAGE</b></td>
</tr>
<tr>
<td colspan="2" class="your_class" valign="top">1 JPG Only - 2 MB max. - Uploading a new image will replace any existing
image.  Images will be uploaded when the &quot;revalidate or update&quot; button at the bottom of this page is
clicked/tapped.</td>
<td><?
php  list($accountRecords,) = getRecords(array('tableName' => 'accounts', 'where' => mysql_escapef('num = ?',
$CURRENT_USER['num']), 'allowSearch' => false, 'ignoreHidden' => true));
 $current_user_with_uploads = @$accountRecords[0];
?>
<?php if (sizeof(@$current_user_with_uploads['profile_image'])):
$upload2 = $current_user_with_uploads['profile_image'][0] ?>
<a href="<?php  echo $upload2['urlPath'] ?>"> <img src="<?
php  echo $upload2['thumbUrlPath'] ?>" width="<?php echo
$upload2['thumbWidth'] ?>" height="<?php echo $upload2['thumbHeight'] ?>" alt="" />

</a>
<?php  else: ?>
<span class="your_class">No (new) image uploaded</span>
<?php  endif ?>
<span class="your_class">Image To Upload:</span>
<input type="file" name="profile_image"></td>
</tr>

<tr>
<td class="your_class" colspan="2"><b>SECOND PROFILE IMAGE</b></td>
</tr>
<tr>
<td colspan="2" class="your_class" valign="top">1 JPG Only - 2 MB max. - Uploading a new image will replace any existing
image. Images will be uploaded when the &quot;revalidate or update&quot; button at the bottom is clicked/tapped.
</td>
<td><?
php   list($accountRecords,) = getRecords(array('tableName' => 'accounts', 'where' => mysql_escapef('num = ?',
$CURRENT_USER['num']), 'allowSearch' => false, 'ignoreHidden' => true));
 $current_user_with_uploads1 = @$accountRecords[0];
// if (!$current_user_with_uploads ) { dieWith404("Record not found!") ; } // code suggested by Daniel 10/2/18
?>
<?php if (sizeof(@$current_user_with_uploads1['profile_image_1'])):
$upload1 = $current_user_with_uploads1['profile_image_1'][0] ?>
<a href="<?php echo $upload1['urlPath'] ?>"> <img src="<?
php echo $upload1['thumbUrlPath'] ?>" width="<?php echo
$upload1['thumbWidth'] ?>" height="<?php echo $upload1['thumbHeight'] ?>" alt="" />
 </a>
 <?php else: ?>
 <span class="your_class">No (new) image uploaded</span>
<?php endif ?>
 <span class="your_class">Image To Upload:</span>
 <input type="file" name="profile_image_1"></td>
</tr>
```

3) Add

```
ENCTYPE="multipart/form-data"
```

to any other code in your opening <FORM> tag.

4) Add the following just before // update user

```
// +++++++++++++++++++++++++++++ Begin upload Image 1 +++++++++++++++++++++++++++++++++++
$uploadFieldName1 = 'profile_image_practitioner_1';
$uploadInfo1 = @$_FILES[$uploadFieldName1];
if (!empty($uploadInfo1['name']) && !$errorsAndAlerts) {
// attempt to save the upload
$errors1 = saveUpload('accounts', $uploadFieldName1, $CURRENT_USER['num'], null, $uploadInfo1, $newUploadNums1);
// check for errors
if ($errors1) {
$errorsAndAlerts .= "There was a problem with your upload: $errors1\n";
}
else {
// if the upload was successful, delete any other uploads associated with that record and field (so users only ever have
one Head Shot)
global $TABLE_PREFIX;
// create query
$where = mysql_escapef(" WHERE tableName = 'accounts' AND recordNum = ? AND num != ? AND fieldName = ?",
$CURRENT_USER['num'], $newUploadNums1[0], $uploadFieldName1);
// remove upload files
$query1 = "SELECT (asterisk) FROM `{$TABLE_PREFIX}uploads` $where";
$result1 = mysqli()->query($query1) or die("MySQL Error: ". htmlspecialchars(mysqli()-
>error) . "\n");
while ($row1 = $result1->fetch_assoc()) {
$files1 = array($row1['filePath'], $row1['thumbFilePath'], @$row1['thumbFilePath2'], @$row1['thumbFilePath3'],
@$row1['thumbFilePath4']);
foreach ($files1 as $filepath1) {
if (!$filepath1 || !file_exists($filepath1) || @unlink($filepath1)) { continue; }
}
}
if (is_resource($result1)) { mysql_free_result($result1); }
// remove upload records
mysqli()-
>query("DELETE FROM `{$TABLE_PREFIX}uploads` $where") or die("MySQL Error: ". htmlspecialchars(mysql_error()) .
"\n");
}
}
//+++++++++++++++++++++++++++++  End upload image 1 ++++++++++++++++++++++++++++

// ++++++++++++++++++++++++++++ Begin upload image 2 +++++++++++++++++++++++++++
$uploadFieldName2 = 'profile_image_practitioner_2';
$uploadInfo2 = @$_FILES[$uploadFieldName2];
if (!empty($uploadInfo2['name']) && !$errorsAndAlerts) {
// attempt to save the upload
$errors2 = saveUpload('accounts', $uploadFieldName2, $CURRENT_USER['num'], null, $uploadInfo2, $newUploadNums2);
// check for errors
if ($errors2) {
$errorsAndAlerts .= "There was a problem with your upload: $errors2\n";
}
else {
// if the upload was successful, delete any other uploads associated with that record and field (so users only ever have
one Head Shot)
global $TABLE_PREFIX;
// create query
$where = mysql_escapef(" WHERE tableName = 'accounts' AND recordNum = ? AND num != ? AND fieldName = ?",
$CURRENT_USER['num'], $newUploadNums2[0], $uploadFieldName2);
// remove upload files
$query2 = "SELECT  (asterisk) FROM `{$TABLE_PREFIX}uploads` $where";
$result2 = mysqli()->query($query2) or die("MySQL Error: ". htmlspecialchars(mysqli()-
>error) . "\n");
while ($row2 = $result2->fetch_assoc()) {
$files2 = array($row2['filePath'], $row2['thumbFilePath'], @$row2['thumbFilePath2'], @$row2['thumbFilePath3'],
@$row2['thumbFilePath4']);
foreach ($files2 as $filepath2) {
if (!$filepath2 || !file_exists($filepath2) || @unlink($filepath2)) { continue; }
}
}
if (is_resource($result2)) { mysql_free_result($result2); }
// remove upload records
mysqli()-
>query("DELETE FROM `{$TABLE_PREFIX}uploads` $where") or die("MySQL Error: ". htmlspecialchars(mysql_error()) .
"\n");
}
}
// ++++++++++++++++++++++++++ End upload image 2 ++++++++++++++++++++++++++++++++++
```

Chris added that more than half of that code is dedicated to removing any other uploads after an upload is successful so that users never have more than one profile image. This is important since I haven't provided any way for users to remove a profile image. They can, of course, replace their current image with another. (This is where the maximum upload limit of 2 comes in. If it was set to allow only 1, then the upload of the replacement image, seen as a second image before the original was removed, would fail.)

UPGRADING THE WEBSITE MEMBERSHIP PLUGIN FROM V1.03 - Dec 29th, 2018

IMPORTANT: There have been some major changes since earlier versions of the plugin, so you won’t be able to just overwrite the old plugin this time.

According to Dave Edis, from Interactive Tools, “There's no easy upgrade so you need to use this (latest version) on new sites or recreate the pages.”

YOU MUST MODIFY SOME OF YOUR WEB PAGES TO USE THIS NEW VERSION

According to Robin Brayer from Interactive tools, the pages that need to be updated to upgrade the Website Membership plugin from 1.03 to the latest version are:

login
password-reset
user-profile
user-signup

He said, “The simplest thing to do is to replace the whole block of php at the top of each membership page (login, password-reset, user-profile and user-
signup pages) with the new block from the example files. If there are any custom
changes, you'll need to re-integrate them.”

NOTE: This recipe I've tried to present a pretty detailed, step by step approach, so even inexperienced programmers shouldn’t have too much trouble upgrading.

So...

Here we go...

CAVEAT: If you’re not running at least version 2.08 of CMS Builder, you’ll have to upgrade to that version. V 1.04 of the membership plugin won’t work without it. Later versions may require the then current version of CMSB to operate. (membership V1.06 and V1.07 require CMSB 2.13)

UPGRADING CMS BUILDER
You can download the current CMSB version from:

http://www.interactivetools.com/upgrade/

To begin the upgrade to the latest version of CMSB, backup your entire MySQL database through the general settings page of your current CMSB Admin interface.

It's a good idea to download your entire cmsAdmin directory to your local computer just in case the upgrade doesn’t work as expected and you have to restore the old CMSB version. I'd suggest that you download a copy of your existing membership related pages as well, since you're going to be making some pretty drastic changes to them.

Once you’ve backed up all the files, upload all of the new folders and files from the latest version of CMSBuilder to your on-
line cmsAdmin folder and overwrite all of the old CMSB operating files with these new files.

Beginning with Version 2.08 the concept of user password encryption was incorporated to CMSB.

IMPORTANT: Make sure that you don’t choose to encrypt your passwords at this point because your current version of the website membership plugin won’t work with encrypted passwords.

ALSO: If you’ve made any custom changes to your CMSB operating files for added functionality, you’ll have to make those changes again to restore that functionality.

UPGRADING THE PLUGIN

Once the new version is installed and working correctly, it’s time to begin the upgrade to V1.07 of the website membership plugin

First download the latest version of the Website Membership plugin from:

http://www.interactivetools.com/add-ons/

and extract the files to a new folder so you don’t overwrite your older plugin files.

*** You'll have to update a number of file paths at the top of the new plugin so that your log in and update files can be found. Sp do that before you upload the new version to your plugins folder.

Next, make a copy of all of the pages that you’re planning to update. (the log in page, the user profile page, the password reminder page, etc.)

If you’re like me, you’ve customized the code on those pages to match your site’s requirements, so you won’t be able to just cut and paste a lot of the code.

Instead, you’ll probably find yourself comparing the active code line by line and updating as required.

THE LOG IN PAGE
Start with the easiest one, which is probably your login page.

Open the copy you made of your current user login page and also the current plugin’s sample login page (user-login.php) from the “examples” directory.

Examine the active code of both, line by line and if you find something that you need to update, change that code and then move to the next.

When you’re done, move on to another pair of pages, like the user signup form.

The first change in the signup form that I found that needed changing was to change the //load

The first place in the signup form that I found that needed changing was to change the // load viewer library code from absolute to relative paths.

The next was to change the password related values in the // Add User section to accommodate the use of encrypted passwords.

After that there are changes the send email section.

In the old form, that code started with:

```
_websiteLogin_sendSignupMessage();
```

Since the plugin now incorporates the ability to automatically send emails using emails/user-new-signup.php The code in
the new form starts with:

```
    // send message
        $emailTemplate = "emails/user-new-signup.php";
```

First, copy any custom error message code you created, then replace the code from:

```
    _websiteLogin_sendSignupMessage();
```

up to:

```
    $_REQUEST       = array(); // clear form values
```

with the new block of code.

THE PASSWORD REMINDER PAGE
If you've got a password reminder page, the active code it will have to be replaced.

Begin with the code in the error checking section at the top of the page, and just below your "Already logged in / logoff" code after the brace, add:

```
### send reset email
  if (@$_REQUEST['action'] == 'sendPasswordReminder') {
    global $SETTINGS, $TABLE_PREFIX;

    // display errors
    if (!@$_REQUEST['usernameOrEmail']) { $errorsAndAlerts .= "No username or email specified!
<br />\n"; }

    // send emails
    if (@$_REQUEST['usernameOrEmail']) {
      $where = mysql_escapef("? IN (`username`,`email`)", $_REQUEST['usernameOrEmail']);
      $user  = mysql_get('accounts', null, $where);

      // send message
      if ($user) {

        // get reset password url
        $resetCode  = _generatePasswordResetCode( $user['num'] );
        $resetQuery = "?userNum={$user['num']}&resetCode=$resetCode";

        //
        $emailTemplate = "emails/user-password-reset.php";
        $emailHeaders  = emailTemplate_load(array(
                          'template'     => websiteLogin_pluginDir() . "/$emailTemplate",
                          'subject'      => '', // set in template
                          'from'         => '', // set in template
                          'to'           => $user['email'],
                          'placeholders' => array(
                            'username' => $user['username'],
                            'loginUrl' => "http://" . $_SERVER['HTTP_HOST'] . $GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL'],
                            'resetUrl' => "http://" . $_SERVER['HTTP_HOST'] . $GLOBALS['WEBSITE_LOGIN_RESET_URL'] .
$resetQuery,
                          ),
                        ));
        $mailErrors   = sendMessage($emailHeaders);
        if ($mailErrors) { die("Mail Error: $mailErrors"); }

        //
        $errorsAndAlerts .= "Thanks, we've emailed you instructions on resetting your password.
<br /><br />
```

```
        If you don't receive an email within a few minutes check your
        spam filter for messages from {$emailHeaders['from']}<br />\n";

        // clear form
        $_REQUEST['usernameOrEmail'] = '';
      }

      //
      if (!$user) { $errorsAndAlerts .= "No matching username or email was found!<br />\n"; }
    }
  }
```

THE PASSWORD REMINDER EMAIL
To implement the password reminder (now a password reset request) you'll need to have 2 new files on your site in addition to the revised "password reminder".

1) A sample password reminder email template that comes with v1.07 of the plugin can be found in the "emails" folder. Change the message text if required, and upload the file to your plugins folder. This send an email to the member requesting their password.

2) A user-password-reset.php that can be found in the examples folder of the plugin. This file updates the user's password Re-skin this file and upload it to the same directory that stores the other user forms, usually the root directory of your site.

Since the idea of resetting passwords may be new to your members, you may find it handy to change the original sample email message from:

```
 Hi <?php echo htmlspecialchars($PLACEHOLDERS['username']) ?>,<br /><br />

You requested a password reset for <?
php echo htmlspecialchars($_SERVER['HTTP_HOST']) ?>.<br /><br />

To reset your password click this link:<br />
<a href="<?php echo $PLACEHOLDERS['resetUrl'] ?>"><?
php echo $PLACEHOLDERS['resetUrl'] ?></a><br /><br />
```

to:

```
Hi <?php echo htmlspecialchars($PLACEHOLDERS['username']) ?>,<br /><br />

For security reasons, passwords can no longer emailed. If you have forgotten your password, you will have to reset it to
a new value for access to <?php echo htmlspecialchars($_SERVER['HTTP_HOST']) ?>.
<br /><br />

To reset your password click this link:<br />
<a href="<?php echo $PLACEHOLDERS['resetUrl'] ?>"><?
php echo $PLACEHOLDERS['resetUrl'] ?></a><br /><br />
```

THE PROFILE CHANGE REQUEST PAGE
If you're not too confused yet, the next change is to your user profile change request page:

The first change is in the "// new password checking" section at the top of your viewer. Copy the entire block and paste it just before the corresponding block in your existing viewer. Change any messages that you've modified, and then delete your old code block.

Then after //update password insert the entire new "// update password if needed", code block:

```
 // update password if needed
    $password        = @$_REQUEST['newPassword2'] ? @$_REQUEST['newPassword2'] : $CURRENT_USER['password'];
    if (@$SETTINGS['advanced']
['encryptPasswords']) { $passwordHash = getPasswordDigest($password); }
    else                                        { $passwordHash = $password; }
```

Next, in your $query = "UPDATE `{$TABLE_PREFIX}accounts` SET section, replace:

```
password        = '".mysql_escape( $CURRENT_USER['password'] )."',
```

with:

```
password        = '".mysql_escape( $passwordHash )."',
```

That should do it.

If you are using CAPTCHA creator, see the recipe entitled, "IMPLEMENTING CAPTCHA ON THE WEBSITE MEMBERSHIP PLUGIN 1.04+ SIGNUP FORM".

If you're using the updated SpamBot plugin 2.0 and you are getting strange results returned in your email form fields, consider adding <?php $GLOBALS['SEP_DISABLED'] = 1; ?
> to the top of those pages, before any includes or requires. This
will disable the plugin on those pages annd your results should look normal again.

If you find anything that I've forgotten, or that needs further explanation, please send me an email and I'll add your discovery to this recipe.

UPDATING OR INSTALLING V1.07 - Aug 15th, 2012

If you updated to or installed ONLY Version 1.07 of the website membership plugin with code you downloaded before Jan 30 2012, and find that when a current user uses the profile.php file to change their password, it doesn't actually change the password, here's the probable culprit:

There was a bug discovered in the profile page example code that was squashed (and the example code revised) on Jan 30, 2012.

To fix it in older builds, search for:

```
$query = "UPDATE `{$TABLE_PREFIX}accounts` SET
```

and change it to:

```
$query = "UPDATE `{$TABLE_PREFIX}" . @$GLOBALS['WSM_ACCOUNTS_TABLE'] . "` SET
```

PERSONALIZING PAGES USING THE WEB SITE MEMBERSHIP PLUGIN - Aug 18th, 2011

Adding a personal touch like "Welcome Aubry Jones" to a page that uses the membership plugin is pretty easy, and it can be done with as little as one line of code.

Here's how...

```
<?php if (@$CURRENT_USER): ?> Welcome <?
php echo mysql_escape($CURRENT_USER['first_name']); ?> <?php echo
mysql_escape($CURRENT_USER['last_name']); ?> <?php endif ?>
```

By substituting any field that exists in your membership account table, and by setting up "elseif" and "else" criteria, you can display many other types of information to users.

IMPLEMENTING THE EMAILONAPPROVED PLUGIN AND A MANDATORY PASSWORD CHANGE ON FIRST LOGIN - Jul 7th, 2020

Website Membership Plugin Required **https://www.interactivetools.com/plugins/website-membership/**

When a prospective member filled out the membership application, I wanted their application to be approved manually. I wanted an e-
mail to be sent stating that their application had been received and that as soon as it was processed they would get a second “welcome” e-mail with their username and a temporary password.

As password encryption became the norm, it became more difficult and then impossible to retrieve a password from the database and send it to a member.

To get around this obstacle, here’s an approach to send members a generic password ands force them to change their password the first time they log in.

NOTE: Because of security measures implemented on your server, you might have to change your admin email to a valid email address on your hosting account, or use an SMTP server email account (set in the Admin > General tab) for emails to automatically be sent by CMS Builder.

1) If you don’t have it already, you’ll need to download the emailOnApproved plugin from:

http://www.thecmsbcookbook.com/downloads/emailOnApproved.zip

2) in emailOnApproved.php search for $message=<<<   TEXT

2) In emailOnApproved.php search for $message=<<<__TEXT__

remove the {$_REQUEST['password']} and replace it with a generic password that you'll also enter into your user-signup
form in step 6

3) You'll also need to download and install the latest version of the Website Membership plugin.

4) Create 2 new check boxes in the 'account' section of your CMS, an 'Approved' check box and a 'First Time Login' check box. Change the checked value to NO and the unchecked value to YES in the First Time Login field.

5) In websiteMembership.php search for return $CURRENT_USER;

add this code just before that line:

```
if (@$_REQUEST['action'] == 'login')   {if (@$CURRENT_USER && (@$CURRENT_USER['first_time_login'] ==
'0'||@$CURRENT_USER['first_time_login'] == ""))
{ redirectBrowserToURL("cp.php");exit;} // if first time login redirect
to change password page
else; }
```

7) In the USER_SIGNUP Email template (you'll find the templates at the bottom of the ADMIN menu group), delete the username, password and login reference and insert the text that's appropriate for your site.

For a membership site, it could be:

"Thanks for signing up.

We'll review your application and email your login credentials to you as soon as your application is approved."

6) In your user-signup form, search for: $colsToValues['password']   = $passwordHash;

Delete that code and replace it with: $colsToValues['password']   = 'YourGenericPassword'; (replacing YourGenericPassword with the one you used in step 2. Keep the single quotes before and after YourGenericPassword.)

7) In your user-signup form, change this:

```
$errorsAndAlerts  = "Thanks, We've created an account for you and emailed you your password.\n";
$errorsAndAlerts .= "If you don't receive an email from us within a few minutes check your spam filter for messages from
{$fromEmail}\n";
```

to something like this:

```
 $errorsAndAlerts  = "Thanks, we've created an account for you. As soon as you're approved we'll email you your
password.\n";
$errorsAndAlerts .= "If you don't receive an email from us within a reasonable time your spam filter for messages from
{$fromEmail}\n";
```

Create a web page named cp.php with the following code and upload to your server in the website root directory:

At the top of your page:

```
<?php $GLOBALS['SEP_DISABLED'] = 1; ?>
<?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."); }
?>
<?
php $GLOBALS['WEBSITE_MEMBERSHIP_PROFILE_PAGE'] = true; // prevent redirect loops for users missing fields listed in
$GLOBALS['WEBSITE_LOGIN_REQUIRED_FIELDS'] ?>
<?
php # Developer Notes: To add "Agree to Terms of Service" checkbox (or similar checkbox field), just add it to the
accounts menu in the CMS and un-comment agree_tos lines

  //
  $useUsernames   = true; // Set this to false to disallow usernames, email will be used as username instead

  // error checking
  $errorsAndAlerts = "";
  if (@$_REQUEST['missing_fields']) { $errorsAndAlerts = "Please fill out all of the following fields to continue.\n"; }
 // if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); }


  ### Update User Profile
  if (@$_POST['save']) {
    // update user
    if (!$errorsAndAlerts) {
     $colsToValues = array();
      //    add more form fields here by copying the above line!
      // ... add more form fields here by copying the above line!
      $colsToValues['first_time_login'] =     '1';
      $colsToValues['updatedByUserNum'] = $CURRENT_USER['num'];
      $colsToValues['updatedDate=']     = 'NOW()';
      mysql_update(accountsTable(), $CURRENT_USER['num'], null, $colsToValues);

      // on success
      websiteLogin_setLoginTo( $colsToValues['username'], $CURRENT_USER['password'] );  // update login session username
in case use has changed it.
      $errorsAndAlerts = "Thanks, we've updated your password.\n";
    }
  }


  ### Change Password
  if (@$_POST['changePassword']) {
  //update fields
    $colsToValues = array();
      $colsToValues['first_time_login'] =     '1';
      $colsToValues['updatedByUserNum'] = $CURRENT_USER['num'];
      $colsToValues['updatedDate=']     = 'NOW()';
      mysql_update(accountsTable(), $CURRENT_USER['num'], null, $colsToValues);
  // change passwords
    $encryptPasswords = @$SETTINGS['advanced']['encryptPasswords'];

    // error checking
    $_REQUEST['oldPassword'] = preg_replace("/^\s+|\s+$/s", '', @$_REQUEST['oldPassword']); // v1.10 remove leading and
trailing whitespace
    $oldPasswordHash  = $encryptPasswords ? getPasswordDigest(@$_REQUEST['oldPassword']) : @$_REQUEST['oldPassword'];
    if     (!@$_REQUEST['oldPassword'])                            { $errorsAndAlerts .= "Please enter your current
password\n"; }
    elseif ($oldPasswordHash != $CURRENT_USER['password'])          { $errorsAndAlerts .= "Current password isn't
correct!\n"; }
    $newPasswordErrors = getNewPasswordErrors(@$_REQUEST['newPassword1'], @$_REQUEST['newPassword2'],
$CURRENT_USER['username']); // v2.52
    $errorsAndAlerts  .= nl2br(htmlencode($newPasswordErrors));

    // change password
    if (!$errorsAndAlerts) {
      $passwordHash = $encryptPasswords ? getPasswordDigest($_REQUEST['newPassword2']) : $_REQUEST['newPassword2'];
      mysql_update( accountsTable(), $CURRENT_USER['num'], null, array('password' => $passwordHash)); // update password
      websiteLogin_setLoginTo( $CURRENT_USER['username'], $_REQUEST['newPassword2'] );                // update current
login session
      unset($_REQUEST['oldPassword'], $_REQUEST['newPassword1'], $_REQUEST['newPassword2']);          // clear form
password fields

      $errorsAndAlerts = "Thanks, we've updated your password!\n";
      redirectBrowserToURL("members-only.php");
    }
  } ### END: Change Password


  // prepopulate form with current user values
  //foreach ($CURRENT_USER as $name => $value) {
  //  if (array_key_exists($name, $_REQUEST)) { continue; }
  //  $_REQUEST[$name] = $value;
 // }

?>
```

And in the body (NOTE: You can style your page to match your site design):

```
<?php if (@$errorsAndAlerts): ?>
      <div align="left" class="your_class" style="color:#F00">
        <?php echo $errorsAndAlerts; ?>
      </div>
      <?php endif ?>
      <div style="width:90%" align="left">
        <div class="your_class" align="center">PLEASE CHANGE YOUR PASSWORD</div>


        <span class="your_class"><b>Welcome <?
php echo $CURRENT_USER['first_name'] ?>,

        Since this is the first time you've logged in, we ask that you
        change your password to protect your privacy.

        NOTE: Once you've changed your password, you'll no longer be logged in,
        and you'll have to </b> </span>
<a class="your_class" href="member_login.php">LOGIN AGAIN</a> <span
class="your_class"><b>with your new credentials.</b>



        <!-- CHANGE PASSWORD FORM -->
        <div > <b>Change your Login Password - (Don't forget to write down the new one!)
</b>
          <form method="post" action="?">
          <input type="hidden" name="changePassword" value="1" /
          <input type="hidden" name="changePassword" value="1" />
          <p>
          <table border="0" cellspacing="0" cellpadding="1">
            <tr>
              <td>Enter Your Current Password</td>
              <td><input type="password" name="oldPassword" value="<?
php echo htmlencode(@$_REQUEST['oldPassword']);
?>" size="40" /></td>
            </tr>
            <tr>
              <td> Enter Your New Password</td>
              <td><input type="password" name="newPassword1" value="<?
php echo htmlencode(@$_REQUEST['newPassword1']);
?>" size="40" /></td>
            </tr>
            <tr>
              <td> Enter Your New Password (again)</td>
              <td><input type="password" name="newPassword2" value="<?
php echo htmlencode(@$_REQUEST['newPassword2']);
?>" size="40" /></td>
            </tr>
            <tr>
              <td >&nbsp;</td>
              <td align="center">
                <input class="button" type="submit" name="submit" value="Change Password &gt;&gt;" />
</td>
            </tr>
          </table>
        </form>
      </div>

      <!-- /CHANGE PASSWORD -->
      <?php if (@$errorsAndAlerts): ?>
      <div class="your_class" style="color:#F00">
        <?php echo $errorsAndAlerts; ?>

      </div>
      <?php endif ?>
```

7) If you're implementing the mandatory password change on an existing site, you might want to change all the First Time Login check boxes to checked so that existing users won't get a reset your password message the next time they log in. To change all the check boxes with one click, you can use the Field Resetter Plugin that you can download from
http://www.thecmsbcookbook.com/downloads/fieldResetter.zip

8) TEST TO MAKE SURE EVERYTHING WORKS AS PLANNED!!!

NOTE: THE STEPS OUTLINED IN THE (LEGACY) RECIPE BELOW WILL ALLOW THIS PLUGIN TO WORK CORRECTLY WITH ENCRYPTED PASSWORDS (CMSB VERSION 2.08+ AND THE WEBSITE MEMBERSHIP PLUGIN V1.05+ (BUT NOT WITH V1.09 or later (SEE ABOVE).

V 1.10 OF THE MEMBERSHIP PLUGIN IS HIGHLY RECOMMENDED

NOTE: Thanks to Steve from MustBeOnLine.com for discovering a coding error in the zipped plugin (now fixed). He discovered that there was a double $errors = in the line of code:

```
$errors = $errors=mail($_REQUEST['email'],"Your membership has been approve!",$message,$headers);,
```

It should be:

```
$errors = mail($_REQUEST['email'],"Your membership has been approve!",$message,$headers);,
```

Download and install the emailOnApproved plugin. Do a search on the forum for the latest version, or you can download my modified (and corrected) version from here:

http://www.thecmsbcookbook.com/downloads/emailOnApproved.zip

This modified plugin will allow you to set up a manual approval process and send an e-mail to your new members when they are approved.

Before encrypted passwords it was easier to email login credentials to a member after their account had been manually approved. (Their application was approved, their payment had been verified, etc.)

Since the implementation of encrypted passwords, the process became a bit more involved.

Here’s are the steps necessary:

First, create a visible password text field in the “Accounts” section (I call it visible_password in this recipe) and an "approved" check box

Then have your signup form automatically fill the visible_password field with the automatically generated password before it’s encrypted.

You’ll need to comment out (or remove) the send email instructions in the signup form

You’ll also want to change the “show thanks” message that is presented on a successful signup.

Then in the emailOnApproved plugin, you’ll want to change the ‘password’ field to ‘visible_password’

Here are the specifics:

First add a text field called "visible_password" to your “accounts” section. (You can call it anything you want to, but be consistent)

Next add a check box field called "approved"

Then open your signup form and somewhere in the mysql_query("INSERT INTO `{$TABLE_PREFIX}accounts` SET code insert the following code as a separate line:

```
visible_password = '$password',
```

it doesn’t matter where in the series you insert the line, as long as it’s a separate line.

Next search for // send message and comment out the entire section with a   after the section:

Just under that section of code you should find the section called // show thanks

You’ll want to modify that message to something like:

```
 // show thanks
      $errorsAndAlerts  = "Thanks, We've created an account for you. As soon as you're approved we'll email you your password.\n";
      $errorsAndAlerts .= "If you don't receive an email from us within a few minutes check your spam filter for messages from {$SETTINGS['adminEmail']}.\n";
   //   $errorsAndAlerts .= "
<a href='{$GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL']}'>Click here to login</a>.";
```

Notice that I’ve removed the reference to {$emailHeaders['from']} in the “spam filter” text and replaced it with {$SETTINGS['adminEmail']} (you’ll get an error if you don’t because you removed the original variable in the previous step) , and commented out the login URL line with a double forward slash (you can remove this line if you’d prefer).

Now save your signup form and open the emailOn Approved plugin.

The instructions on how to implement this plugin are below.

Search for ‘password’ and replace this with ‘visible_password’ (unless you called your field something else)

That’s it.

LEGACY INFORMATION

BACKGROUND
When I first activated the original emailOnApproved plugin, I was getting Undefined index errors reported on saving a record until I added this line suggested by Dave Edis to the error checking section of the code:

```
 if ($Table name != 'accounts') { return; }
```

So now the error checking section looks like

```
 // error checking
    if ($tableName != 'accounts') { return; }
  if (!array_key_exists($fieldname, $CURRENT_USER)) {
    die(__FUNCTION__ .": You must create an accounts fields called '$fieldname'!");
  }
```

And there are no more Undefined index errors.

I decided that it would make more sense to include the new member’s username, temporary password and a login URL in this email, so I modified the original:

```
 // send email
  $wasChecked   = intval(!$oldRecord[$fieldname] && $_REQUEST[$fieldname]);
  $wasUnchecked = intval($oldRecord[$fieldname]  && !$_REQUEST[$fieldname]);

  if ($wasChecked) {
    $errors = sendMessage(array(
      'from'    => $SETTINGS['adminEmail'],
      'to'      => $_REQUEST['email'],
      'subject' => "You have been approved!",
      'text'    => "Congradulations!

You have been approved for our website.
Your password is: {$_REQUEST['password']}

See you soon!
",
    ));
    if ($errors) { die($errors); }
  }

}

?>
```

To this:

NOTE: Make sure that there are no spaces after the $message=
<<< __TEXT__ and that the __TEXT__; is flush against the
left margin or you'll generate errors.

```
  // send email
  $wasChecked   = intval(!$oldRecord[$fieldname] && $_REQUEST[$fieldname]);
  $wasUnchecked = intval($oldRecord[$fieldname]  && !$_REQUEST[$fieldname]);

$message=<<< __TEXT__
    Congratulations!

Your subscription has been  processed successfully and you now have access to the "Members Only" area of our web site.
Your user name is: {$_REQUEST['username']}
and your temporary password is: {$_REQUEST['password']}.
Once you have successfully logged in, you can change your password and update your profile information.


<a href="http://www.your_web_site_URL.com{$GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL']}">Click here to login</a>

Best,

The Subscription Committee

__TEXT__;

$headers  = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
$headers .="FROM:". $SETTINGS['adminEmail'];

   if ($wasChecked) {
    $errors = $errors=mail($_REQUEST['email'],"Your membership has been processed!",$message,$headers);
    if ($errors!=1) { die("Mail Error: $php_errormsg"); }
  }
}
 ?>
```

ADDING CUSTOMIZABLE TEXT TO THE EMAILONAPPROVED PLUGIN WELCOME MESSAGE - Sep 9th, 2015

You can add custom text from a field ('member_welcome'_in a single record editor ('common_information') or add multiple variables from multiple fields.

The trick is to keep the load viewer library (don't forget to change the server path) and load records calls, and the variables you create within the function itself.

Then place each variable in a set of braces {} inside the $message=<<< __TEXT__ block

```
<?php

// UPDATE THESE VALUES
// DON'T UPDATE ANYTHING BELOW THIS LINE

addAction('record_postsave',  'emailOnApproved_sendPassword', null, 4);
//
function emailOnApproved_sendPassword($tableName, $isNewRecord, $oldRecord, $recordNum) {
  global $CURRENT_USER, $SETTINGS;
  $fieldname = 'approved';
  // error checking
    if ($tableName != 'accounts') { return; }
    if ($tableName != 'accounts') { return; }
  if (!array_key_exists($fieldname, $CURRENT_USER)) {
    die(__FUNCTION__ .": You must create an accounts fields called '$fieldname'!");
  }
  // send email
  $wasChecked   = intval(!$oldRecord[$fieldname] && $_REQUEST[$fieldname]);
  $wasUnchecked = intval($oldRecord[$fieldname]  && !$_REQUEST[$fieldname]);

// load viewer library
  $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
  $dirsToCheck = array('/path_to_server/','','../','../../','../../../');
  foreach ($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 'common_information'
  list($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName'   => 'common_information',
    'where'       => '', // load first record
    'loadUploads' => true,
    'allowSearch' => false,
    'limit'       => '1',
  ));
  $common_informationRecord = @$common_informationRecords[0]; // get first record
  $member_welcome  = $common_informationRecord['member_welcome'];

$message=<<< __TEXT__
    Welcome {$_REQUEST['first_name']} {$_REQUEST['last_name']},

{$member_welcome}



Your membership has been  processed successfully and you now have access to the Members Only area of our web site.

Your user name is: {$_REQUEST['username']}
and your temporary login password is: XXXXX

Once you have successfully logged in, you must change your password and can then update your own profile information.

<a href="{$GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL']}">Click here to login</a>

Best,

The Membership Committee
__TEXT__;


$headers  = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
$headers .="FROM:". $SETTINGS['adminEmail'];

   if ($wasChecked) {
    $errors = mail($_REQUEST['email'],"Your membership has been successfully processed!",$message,$headers);
    if ($errors!=1) { die("Mail Error: $php_errormsg"); }
  }

}

?>
```

CHANGING THE TEXT OF THE SIGNUP EMAIL - Jun 1st, 2011

Another thing you'll probably want to do is to change changing the text of the sign up e-mail that's sent to
prospective members.

At the bottom of the websiteMembership.php plugin file, you'll find the following block of code.

```
    // send email
    $to         = $_REQUEST['email'];
    $subject   = "{$_SERVER['HTTP_HOST']} Account Details";
    $message   = "Hi {$_REQUEST['email']},\n"
             . "\n"
             . "Thanks for signing up to {$_SERVER['HTTP_HOST']}.\n"
             . "\n"
             . "Your username is: {$username}\n"
             . "Your password is: {$_REQUEST['password']}\n"
             . "\n"
             . "Please click here to login:\n"
             . "http://{$_SERVER['HTTP_HOST']}
  {$GLOBALS['WEBSITE_LOGIN_LOGIN_FORM_URL']}\n"
             . "\n"
             . "Thanks!\n";
    $mailResult = @mail($to, $subject, $message, "From: {$SETTINGS['adminEmail']}");
    if (!$mailResult) { die("Mail Error: $php_errormsg"); }

  }


  ?>
```

While this is OK for a one step sign up approach, you'll probably not want to include the user name, password, or login address in this e-mail.

In the version below, I've changed that to a more logical set of instructions. Including a payment link if required.

Note that /n is the code for a new line, there is only one ; and it marks the end of the message $section, the . is the operator that says “add” this line to the string,

```
// send email
  $to        = $_REQUEST['email'];
  $subject   = "{$_SERVER['HTTP_HOST']} Membership Application";
  $message   = "Hi {$_REQUEST['email']},\n"
. "\n"
. "Thanks for applying to be a member of {$_SERVER['HTTP_HOST']}.\n"
. "\n"
            . "As soon as your membership has been approved you'll receive a confirmation email with a link to our
login page.\n"
            . "Once you log in you'll be able to enter your member profile information so that others can contact you.\n"
           . "\n"
            . "Remember, you can not be approved if you haven't paid your first year's dues.\n"
            . "\n"
            . "You can pay your dues on-line using PayPal here:\n"
            . "http://www.your_website.org/paypal.php\n"
            . "\n"
            . "Thanks and Welcome!\n"
             . "\n"
             . "The Membership Committee\n";

  $mailResult = @mail($to, $subject, $message, "From: {$SETTINGS['adminEmail']}");
  if (!$mailResult) { die("Mail Error: $php_errormsg"); }

}
```

FORCING UPPER CASE, LOWER CASE, AND INITIAL CAPS TO BE SENT TO THE DATABASE - Jun 1st, 2011

Sometimes it's important to send only certain case formats to your database. State abbreviations, first and last names, and middle initials are a few examples.

It's pretty easy, but you'll have to implement any changes on both the signup and the profile revision form pages.

In the “// add user” section of the signup form, fairly near the top of the page, you’ll find:

```
mysql_query("INSERT INTO `{$TABLE_PREFIX}accounts` SET
```

And in the “// update user” section of the profile page, you’ll find:

```
 $query = "UPDATE `{$TABLE_PREFIX}accounts` SET
```

Just before each of those lines, insert any of the following, creating a new line of code for each field that you want to control:

To force each word to begin with a capital letter:

```
@$_REQUEST['your_field']=ucwords(@$_REQUEST['your_field']);
```

To force only the first character in the field to begin with a capital letter:

```
@$_REQUEST['your_field']=ucfirst(@$_REQUEST['your_field']);
```

To force the entire field to be converted to capital letters:

```
@$_REQUEST['your_field']=strtoupper(@$_REQUEST['your_field']);
```

ADDING/CHANGING FIELDS AFFECTED BY THE SAMPLE SIGN UP FORM - Jun 1st, 2011

You can add, change, and remove any fields that you want to be automatically affected by the membership plugin sign up form. Here’s how.

First, in the user accounts table create any fields that don’t exist.

Then open sample_signup.php

Near the top, you’ll find the error checking code section that starts with:

```
  // error checking
    $errorsAndAlerts = "";
```

If any of your new fields are going to be “required’ fields, here’s where you’d list them. If you’re removing any, make sure you remove them here too. Copy existing lines of code and change the field name and message to match your requirements.

Next scroll down to the “add user” section and copy an existing line of code and change the field name to match your requirements.

Next, scroll down to the form in the body of your page and add the required fields by again copying and pasting rows into your form and changing the information as required.

If you want some of your fields to be text areas instead of text fields, you can change this field format:

```
<td class="body-text-bold">City</td>
<td>
<input type="text" name="city" value="<?
php echo htmlspecialchars(@$_REQUEST['city']); ?>" size="50" />
</td>
</tr>
```

To this for a 50 column, 6 row text area:

```
<tr>
<td class="body-text-bold">Describe Your Qualifications</td>
<td>
<TEXTAREA NAME="describe_discipline" COLS=50 ROWS=6 value="<?php echo
htmlspecialchars(@$_REQUEST['desctribe_discipline']); ?>"></TEXTAREA>
</td>
</tr>
```

You can further modify the look of a form using MAXLENGTH="n" to limit your fields to n characters (good for zip codes and States for example.)

DISPLAYING THE CURRENT VALUE OF A CHECKBOX OR RADIO BUTTON - Jun 1st, 2011

If you want to display the current values in a checkbox or radio button field when a member logs in to revise their profile, you can take this advice offered by Jason Sauchuk from Interactive Tools:

A checkbox in CMS Builder can only store a value of either 1 or 0.

What you need to do is to display your form element as either selected or unselected based on the value currently stored in the database. There are two solutions that you could use.

1) You can use a single check box. If a member had previously checked yes, the box on the form will be checked, otherwise, it will be blank:

```
<tr>
  <td >Your Question Goes Here:</td>
  <td ">Check for "Yes" <br /> <input type="checkbox" name="your_field" value="1"  <?php
if(@$_REQUEST['your_field']){echo "checked='checked'";}?> /> </td>
</tr>
```

2) The other option would be to use radio buttons for yes and no. This way they can only select 1:

```
<tr>
  <td >Your Question Goes Here:</td>
  <td ><input type="radio" name="your_field" value="1"  <?
php if(@$_REQUEST['your_field']){echo "checked='checked'";}?>
/> Yes please <br />
<input type="radio" name="your_field" value="0" <?php if(!@$_REQUEST['your_field'])
{echo "checked='checked'";}?>> No
Thanks</td>
</tr>
```

RETAINING THE VALUE IN A TEXTAREA FORM FIELD AFTER FORM IS RE-SENT - Jul 2nd, 2012

In a member signup form I had a textarea form field that would lose the information that was entered each time the form was re-
sent (as in error checking, etc.) and it took Jason Sauchuk from Interactive Tools to help me see the tree in the forest.

He said (very patiently, I might add),

"To put information into a textarea, you output it between the <textarea> tags."

For example:

<textarea name="your_name" COLS=50 ROWS=6 ><?
php echo htmlspecialchars(@$_REQUEST['your_field']); ?></textarea>

RETAINING THE VALUE OF A PULL DOWN LIST AFTER A FORM IS RE- SENT - Dec 27th, 2022

Here's what I've used for a single value list field:

```
<select name="venue_parking" >
<option value"" <?php if($_REQUEST['venue_parking'] == "")
{echo "selected=\"selected\"";} ?> >Select </option>
<option value"Street" <?php if($_REQUEST['venue_parking'] == "Street")
{echo "selected=\"selected\"";} ?>
>Street</option>
<option value"On Site Free" <?php if($_REQUEST['venue_parking'] == "On Site Free")
{echo "selected=\"selected\"";} ?> >On
Site Free</option>
<option value"On Site Paid" <?php if($_REQUEST['venue_parking'] == "On Site Paid")
{echo "selected=\"selected\"";} ?> >On
Site Paid</option>
</select>
```

And here's what was suggested for a multi value list field (never tested):

```
<select name="foo[]" multiple="multiple" size="3">
    <option value="" <?
php if(isset($_POST['foo'])) { foreach($_REQUEST['foo'] as $tmp) {  if($tmp == "") { echo
"selected=\"selected\""; break; }}}?>>Please Select One Or More Values</option>
    <option value="bar" <?
php if(isset($_POST['foo'])) { foreach($_REQUEST['foo'] as $tmp) {  if($tmp == "bar") { echo
"selected=\"selected\""; break; }}}?>>bar</option>
    <option value="hello" <?
php if(isset($_POST['foo'])) { foreach($_REQUEST['foo'] as $tmp) {  if($tmp == "hello") {
echo "selected=\"selected\""; break; }}}?>>hello</option>
    <option value="world" <?
php if(isset($_POST['foo'])) { foreach$_REQUEST['foo'] as $tmp) {  if($tmp == "world") {
echo "selected=\"selected\""; break; }}}?>>world</option>
```

</select>

ADDING/CHANGING FIELDS AFFECTED BY THE SAMPLE PROFILE UPDATE FORM - Jun 1st, 2011

The process here is similar to modifications to the sign up form except here you'd be updating the "error checking", "update user" and "form" sections of your page. Again it's easier to copy lines of code and change information as required.

SETTING ACCESS RIGHTS FOR NEW USERS BY SECTION - Aug 30th, 2014

If you've had trouble setting a new user's access rights "by Section", there are a few counter-intuitive steps
that need to be added to the process.

If you look at the sample signup form in the Website Membership Plugin version1.10, you'll find this block of code.

```
 // NOTE: You can repeat this block to grant access to multiple sections
       mysql_insert('_accesslist', array(
         'userNum'      => $userNum,
         'tableName'    => '_sample',   // insert tablename you want to grant access to, or 'all' for all sections
         'accessLevel'  => '0',         // access level allowed: 0=none, 6=author, 9=editor
         'maxRecords'   => '',          // max listings allowed (leave blank for unlimited)
         'randomSaveId' => '123456789', // ignore - for internal use
       ));
     }
```

SETTING 'BY SECTION' ACCESS RIGHTS
You'll first have to set the access permission for that user to By Section, to allow you to set the user's permissions for individual sections.

for individual sections.

Copy the code block above, and in the first copy of the code on the page, set the tableName to 'all', and the accessLevel to '1' (which means ‘By Section’), like this:

```
mysql_insert('_accesslist', array(
 'userNum'      => $userNum,
    'tableName'    => 'all',   // insert tablename you want to grant access to, or 'all' for all sections
   'accessLevel'  => '1',         // access level allowed: 0=none, 6=author, 9=editor
 //  'maxRecords'   => '',          // max listings allowed (leave blank for unlimited)
    'randomSaveId' => '123456789', // ignore - for internal use
  ));
```

Notice that the maxRecords has been commented out. According to Daryl Maximo, a programmer at Interactive Tools, "This is because unless there’s a number in the maxRecords value, the value will actually be set to '0'" If that happens, you won't get the result that you're expecting, or an error will be thrown and the access levels won’t be set at all.

SETTING ACCESS RIGHTS FOR INDIVIDUAL SECTIONS
Now that access by section has been set, the next code blocks can set the individual section access rights. My example below sets the access rights to the section tableName ‘listing’ for this new user to only those records where they are the Author (6). Use as many copies of this code block required to set each one of your section access scenarios.

Again, be careful about that maxRecords value. If you want to allow unlimited records, just comment out the line by putting a double forward slash // in front of it.

```
        mysql_insert('_accesslist', array(
         'userNum'      => $userNum,
         'tableName'    => 'listing',   // insert tablename you want to grant access to, or 'all' for all sections
         'accessLevel'  => '6',         // access level allowed: 0=none, 6=author, 9=editor
       'maxRecords'   => '1',          // max listings allowed (leave blank for unlimited)
         'randomSaveId' => '123456789', // ignore - for internal use
       ));
```

So the complete code for the page would be:

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


  // load viewer library
  $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
    $dirsToCheck = array('/path_to_your_serverl/','','../','../../','../../../');
  foreach ($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 'accounts'
  list($accountsRecords, $accountsMetaData) = getRecords(array(
    'tableName'   => 'accounts',
    'loadUploads' => true,

  ));

?><!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" />
<title>access-update2</title>

</head>

<body>
<?php foreach ($accountsRecords as $record): ?>
<?php

$accountsRecords = mysql_select(accountsTable(), 'isAdmin != 1');
foreach($accountsRecords as $record){

  $userNum = $record['num'];

  mysql_insert('_accesslist', array(
    'userNum'      => $userNum,
    'tableName'    => 'all',   // insert tablename you want to grant access to, or 'all' for all sections
    'accessLevel'  => '1',         // access level allowed: 0=none, 6=author, 9=editor
    //'maxRecords'   => '',          // max listings allowed (leave blank for unlimited)
    'randomSaveId' => '123456789', // ignore - for internal use
  ), true);

  // NOTE: You can repeat this block to grant access to multiple sections
  mysql_insert('_accesslist', array(
    'userNum'      => $userNum,
    'tableName'    => 'lm_listing',   // insert tablename you want to grant access to, or 'all' for all sections
    'accessLevel'  => '6',         // access level allowed: 0=none, 6=author, 9=editor
    //'maxRecords'   => '',          // max listings allowed (leave blank for unlimited)
    //'maxRecords'   => '',          // max listings allowed (leave blank for unlimited)
    'randomSaveId' => '123456789', // ignore - for internal use
  ), true);

}
?>
<?php endforeach ?>
  </body></html>
```

LIMITING DIRECTORY ACCESS AND PROTECTING DIRECTORIES - Jun 1st, 2011

Thanks to Dave Edis from Interactive Tools for this part.

Since all the files are PHP they are executed server side and protected from prying eyes by default.

1) In your user accounts editor, create a multi value list field called allowedFolders
The values I used are:
board_access
committee_access
general_access

Go into the User Accounts list and check some values in your account to authorize access to some of those folders.

2) In the root directory of your site create 3 folders:
board_access
committee_access
general_access

3) In each folder upload a test file called access.php

the code in each of the access.php files is:

```
<?php
  require_once "../cmsAdmin/lib/viewer_functions.php";
  if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); }
  if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); }

  //
  $folderName  = basename(dirname(__FILE__));
  $allowAccess = preg_match("/\t$folderName\t/", $CURRENT_USER['allowedFolders']);
  if (!$allowAccess) { die("You don't have access to this folder!"); }
?>
<!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" />
<title>Access Test</title>
</head>
```

```
</head>

<body>
<?php echo $CURRENT_USER['allowedFolders'] ?>
</body>
</html>
```

The code in the body just prints out the directories that the particular member is authorized to view.

After you’ve got the basic code working, you can paste the code required into an existing page, upload it to one of those directories, and make sure it all works as planned.

LIMITING FILE ACCESS AND PROTECTING INDIVIDUAL FILES - Jun 1st, 2011

If you’ve only got a few files to protect, you can use something like this:

Set up a single value list field in the user accounts table with the membership levels required, for example:

General
Committee
Board

Assign access to one of these levels to your user account.

Then use code similar to this to protect whole pages or portions of a page. If the user is not logged in, line 3 in the code will force them to log in before they can access this page:

```
<?php   require_once "cmsAdmin/lib/viewer_functions.php";
if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); }
  if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); }
?>
<!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" />
<title>Information Access Test</title>
</head>

<body>
<?php if ($CURRENT_USER['membership_level'] == "Committee"): ?
>You are logged in and are authorized to see this
Committee level information.
<?php elseif ($CURRENT_USER['membership_level'] == "Board"): ?
>You Are Logged in and are authorized to see this Board of
Directors level information.
<?php elseif ($CURRENT_USER['membership_level'] == "General"): ?
>You Are Logged in and are authorized to see this
General membership level information.
<?PHP else: ?
> You must log in before you can see the information that is authorized for your level of membership.
<?PHP endif ?>

</body>
</html>
```

If you’re using a multi value list to define your authorization levels, you can’t use the == test, since your field can actually contain more than just an exact match. You can however use the strpos function to determine if a particular membership level is contained in the list.

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.

Here’s the code using strpos

```
<?php if (strpos($CURRENT_USER['membership_level'], 'committee_access')): ?>
You Are Logged in and are authorized to see committee level level information.
<?PHP endif ?>
```

```
<?PHP endif ?>
```

SETTING CRITERIA FOR WHICH RECORDS WILL SHOW ON AN "ACCESS PERMISSIONS" REPORT BASED ON THE AMOUNT OF CHECKED LIST FIELD VALUES. - Apr 13th, 2016

There were a number of reports and other pages associated with my membership site that needed access restricted to particular members. I.E. The membership committee did not have any reason to have access to the Exhibition Submission reports, etc.

To accomplish this I created a multi value check box list field of access permissions (allowedFolders) in the user database, and checked the appropriate permissions on a member by member basis.

All members were granted member_access and some had other permissions as well.

To aid in the assignment and reassignment process, I created a report that shows the all the checked value label in the allowedFolders list field for all user accounts.

The problem was that the list page showed all users, and I wasn't sure how to exclude those records with only the one value checked.

Damon Edis form Interactive Tools came to the rescue by adding the substr_count function in the foreach loop to count the tab spaces in the allowedFolders string. He explained: “One value equals two tab spaces so I added the ' more than 2' condition.”

```
 <table class="text_font" width="100%" border="1"  cellspacing="0">
  <tr>
    <td style="padding-left:10px;">Last Name</td>
    <td style="padding-left:10px;">First Name</td>
    <td style="padding-left:10px;" >Assignments</td>
    <td style="padding-left:10px;" >Administrator</td>

  </tr>


<?php foreach ($accountsRecords as $record ): ?>
        <?php if(substr_count($record['allowedFolders'], "\t") > 2): ?>

      <tr>
    <td style="padding-left:10px;"><?php echo $record['last_name'] ?></td>
    <td style="padding-left:10px;"><?php echo $record['first_name'] ?></td>
    <td style="padding-left:10px;"><?php echo join(', ', $record['allowedFolders:labels']);?>
</td>
    <td style="padding-left:10px;"><?php if ($record['isAdmin'] == 1): ?>Yes<?php else :?
>No<?php endif?></td>
  </tr>
       <?php endif?>
          <?php endforeach; ?>


        </table>
```

LIMITING ACCESS TO RECORDS BY AUTHOR - Jun 13th, 2011

To limit access to only those records in a multi-record section that were created by the current user, since each record is flagged with the user number of the user that created it (['createdByUserNum']), you can use this approach:

```
<?php foreach ($your_sectionRecords as $record): ?>
 <?php if ($record['createdByUserNum']== @$CURRENT_USER['num']): ?>

<!-- the fields that you want to show go here -->

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

RESTRICTING LIST VALUES TO AUTHORS ONLY - Apr 13th, 2016

On a site where I had employed the membership plugin, I was trying to set up a list that gets it's options from the "title" field of another table (your_source_table), using the "get options from mySQL query" option.

The trick was that I wanted to restrict the options that a member user can see to only those where they are the option's author, but as an administrator I would need to see them all.

Thanks to user Zip222, I came up with this solution.

First I set up a multi-record editor we'll call "your_source_table" with only one field called "title".

Next I created a list field in my 'receiving" editor and for the list values I chose the third option in the pull down menu, "Get Options From MySQL Query (advanced)

I put the following code in the box to replace the dummy code that is there.

NOTE: The first entry in the SELECT determines the field from which the"Value" is returned (so that could be 'num', which would return the record number, etc) and the second determines the Label that appears in the pull down list.

```
SELECT title, title

  FROM <?php echo $TABLE_PREFIX ?>your_source_table_name
<?php if (!$CURRENT_USER['isAdmin']): ?>
<?php   $where="WHERE createdByUserNum='".$CURRENT_USER['num']."'"; ?>
<?php echo $where ?>
<?php endif   ?>
```

PROTECTED CLIENT ACCESS TO SPECIFIC RECORDS IN A MULTI-RECORD SECTION - Mar 11th, 2015

Using CMSB V2.64 and Website Membership V1.10, I needed to set up a protected client area on a site where client access is granted to certain records in a multi-record section based on a master project assignment list. I needed to be able to grant any client access to a list page with all the projects they were authorized to see but I also needed to protect project detail pages from other clients who were logged in, but did not have access to view particular records.

Here's what was needed:

1) A multi-record "Project Name" section with the name of the project in a text field called 'title' (this field was set as "don't allow removing of records" on the "advanced" tab) and a warning about creating a new record for each new project so that project information did not fall into the wrong hands)

2) A multi-record "Project Information" section with a pull down list field called 'project_name' that get's its option values from the num field in the "Project Name" table and it's option labels from the title field in the same table. (This field is used to assign progress report names)

3) A multi-value pull down list field in the "accounts" section that gets it's data from the 'Project Name" table in the same way. (This field is used to assign users permission to view project progress reports)

On the list page, called client.php, at the top of the document, after the load records calls, I included the code:

```
<?php if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); } ?>
```

And in the body:

```
<?php $rec_count = 0?>
<?php foreach ($project_informationRecords as $record): ?>
<?php if (strpos($CURRENT_USER['project_access'], $record['project_name'])): ?>
<div  style="text-align:left;" >
<?php $usernum1 = $CURRENT_USER['num'] ?>
<a class="special" href="clientdetail.php?usernum=<?php echo $usernum1 ?>&num=<?
php echo $record['num']?>"><span
style="text-decoration:underline;" class="navigation_font"><i><?
php echo strtoupper($record['project_name:label'])
?></i></span></a>
</div>
<?php endif ?>
<?php endforeach ?>

<?php if (!$project_informationRecords): ?>
<span class="text_font">Sorry, You Don't Have Permission To Access Any Project Progress ReportsIf you feel that you
should have access, please </span><a href="http://www.your_site/contact.php">
<span class="text_font">"CONTACT
US</span></a>
```

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

And on the detail page at the top of the document, below the load records calls:

```
<?php if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); } ?>
<?php $usernum = $_REQUEST['usernum']; ?>
<?php $record['num'] = $_REQUEST['num']; ?>
<?php @$errorsAndAlerts = "?>
<?php if ($CURRENT_USER['num'] == $usernum ):?>
<?php $show_data = 1 ?>
<?php else: ?>
<?php $show_data = 0 ?>
<?
php $errorsAndAlerts = "Sorry, You're Currently Not Authorized To View Project Progress Reports." ?
>
<?php endif ?>
```

And in the body:

```
<table align="center" width="50%" border="0" cellpadding="3">
<tr>
<td align="center" class="heading_font">PROJECT PROGRESS REPORTS
<?php if($show_data == 1):?>
<?php echo strtoupper($project_informationRecord['project_name:label']) ?>
FOR <?php echo strtoupper($CURRENT_USER['first_name']) ?> <?
php echo strtoupper($CURRENT_USER['last_name']) ?>

<a href='http://www.your_site.com/client.php?action=logoff'>
<span class="text_font">WHEN YOU'RE FINISHED, PLEASE
CLICK/TAP HERE TO LOG OUT</span></a>

</td>
</tr>
<tr>
<td >
YOUR DETAIL PAGE DATA GOES HERE...
</td>
</tr>
</table>
<?php else :?>
<span class="navigation_font">
<?php echo $errorsAndAlerts ?>
For Authorization, Please <a href="contact.php"><span class ="text_font">
<font color="#D01E1E">Contact
Us</font></span></a>.
<?php endif ?>
```

CHECKING TO SEE IF AN EMAIL ADDRESS (ACCOUNT) EXISTS IN THE ACCOUNTS DATABASE - Aug 6th, 2012

In my organization’s scenario, if a member has not paid their dues for a period of 2 years their record is deleted from the database.

To deal with those members who still have a record in the database, I created (with a lot of help from Jason Sauchuk from Interactive Tools) a function that checked a prospective member’s e-mail and reported whether their account still exists.

Here’s what I did.

On the member signup page I created this form with the following explanatory text. (I'll show the form in the context of the page code below)

To find out if you're still in our member database, enter the e-mail address you used to sign up with in the box below and click on &quot;submit&quot;. If your records are found, all you’ll have to do to reinstate your account is to pay your current dues. Once your payment is processed, you’ll receive an e-mail with your login username and password.
Then you’ll again be able to enjoy the benefits of membership and update your member profile information.

```
  <?php if (!@$CURRENT_USER): ?>
        <form action="?" method="post">
    <input type="hidden" name="action" value="pastInformation" />
   <span class="body-text-bold">Enter your e-mail address:</span>
    <input type="text" name="pastInformation" value="<?
php echo htmlspecialchars(@$_REQUEST['pastInformation']) ?>"
size="20" />
    <input type="submit" name="submit" value="Submit" />
    </form>
```

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

To display the appropriate error messages, like "please enter an email address" if the submitted form was empty, or "your information was found" or "your information was not found" on the member page, I added the following to the top of the page after the record calls:

```
<?php
// clear errors and set form to show on page load
 $errorsAndAlerts = "";
  $showForm = true;

 if (@$_REQUEST['submit']) {
    $showForm = false;
    if (!@$_REQUEST['pastInformation']) {
       $emptyField = "<span class='heading-text-
yellow'>Please enter an email address</span>\n";
    $showForm = true;
    }
     }
  // error checking
  $errorsAndAlerts = alert();

  if (@$CURRENT_USER) {
    $errorsAndAlerts = "<span class='heading-text-
yellow'>YOU ARE ALREADY LOGGED IN!</span><a class='special'
href='{$GLOBALS['WEBSITE_LOGIN_POST_LOGIN_URL']}'>CLICK HERE TO CONTINUE TO THE WEB SITE</a> or <a class='special'
href='?action=logoff'>LOG OUT</a>.";
  }

?>
```

I wanted the form and instructions to show along with the, "please enter an email address" error message and wanted it to be hidden if the email address was either found or not found in the accounts database.

I added this code to the body of the page:

```
<?php if (@$emptyField): ?>
<?php $showform = true; ?>
<div >
<br />
<?php echo $emptyField; ?><br /></div>
<?php elseif (@$errorsAndAlerts): ?>
<?php $showform = false; ?>
 <div ><br />
<?php echo $errorsAndAlerts; ?><br />
</div>

<?php else: ?>

...some code and text...

 <?php if ($showForm == 'true'): ?>

... some text...

<?php if (!@$CURRENT_USER): ?>
<form action="?" method="post">
<input type="hidden" name="action" value="pastInformation" />
<span class="body-text-bold">Enter your e-mail address:</span>
<input type="text" name="pastInformation" value="<?
php echo htmlspecialchars(@$_REQUEST['pastInformation']) ?>"
size="20" />
<input type="submit" name="submit" value="Submit" />
</form>
<?php endif ?>

...more instructions...

<?php endif ?>
```

In the websiteMembership plugin in the "// perform website login actions" section, I added the following to the list of actions:

```
 if (@$_REQUEST['action'] == 'pastInformation') { _websiteLogin_pastInformation(); }
```

Then, immediately before the "// perform website login actions" section of code, I added this block of code to check for the existence of the email address in the accounts database.

```
// Existing Account Request
function _websiteLogin_pastInformation() {
  global $SETTINGS, $TABLE_PREFIX;

  $exists=0;
 if(@$_REQUEST['pastInformation']){
  $where = "email ='".mysql_escape(@$_REQUEST['pastInformation'])."'";
  $exists=mysql_select_count_from('accounts',$where);
 }

  if ($exists) {
    alert("<span class='heading-text-yellow'>Congratulations!!!
<br />Your Account Exists</span><br /><span
class='your_class'>You do not need to create a new account.</span>\n");
    }
    if (!$exists) {
alert("<span class='your_class'>Sorry!!!<br />Your Account No Longer Exists</span>
<br /><span class='your_class'>You'll
have to fill out a new membership application after you you've paid your current year's dues.
<br />To search for your
information using another email address <a href='becomeamember.php'>
<span class='your_class'>CLICK HERE</span></a><br
/></span>\n");

  }
}
```

To make contents that show on your viewer dependent on the values returned, you can use a combination of:

```
 <?php if ($showForm == 'true'): ?> your contents <?php endif ?>
```

and

```
<?php if ($showForm = 'false' && (strpos($errorsAndAlerts,'Your-case-sensitive--text-
string'))): ?> your contents <?php
endif ?>
```

or add a ! before the variables (!$showform = etc. to cover situations that do not meeting those criteria

REDIRECTING VISITOR TO A SPECIFIC PAGE AFTER UPDATING THEIR PROFILE, ETC. - Jun 1st, 2011

Here's a tidbit that was offered by Jason Sauchuk from Interactive tools.

He said:

You can redirect to any page after you finish updating the information in the database with this function

```
redirectBrowserToURL("YOUR_PAGE.php");
```

ADDING A FIELD TO SHOW WHEN A MEMBER LAST LOGGED IN - Jun 1st, 2011

If you add a date field called lastLoginDate, to the User Accounts database, the website membership plugin will automatically store the last time the user logged in for you

(This has been added to version 2.08 as a default field under User Accounts that is automatically updated when you login to the CMS)

WORKING WITH DATE FIELDS IN THE MEMBER SIGNUP AND PROFILE UPDATE FORMS - Jan 20th, 2011

BACKGROUND
Along with other information, my client needed to list the dates of birth for her students in their records in the accounts database.

She wanted to include date fields for 3 students. Student 1, 2 and 3.

This meant that the parents would have to be able to enter the dates into date fields in the database record they were creating, and also be able to update those date fields when required.

Here's the code we used to accomplish these tasks, (with a lot of help from Jason Sauchuk from Interactive Tools).

THE MEMBER APPLICATION FORM
In the original application form, in the PHP code at the top of the page, in the //add user section:

```
$student_1_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_1'],@$_REQUEST['day_1'],@$_REQUEST['year_1']));
$student_2_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_2'],@$_REQUEST['day_2'],@$_REQUEST['year_2']));
$student_3_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_3'],@$_REQUEST['day_3'],@$_REQUEST['year_3']));
```

and in the mysql_query("INSERT INTO `{$TABLE_PREFIX}accounts` SET section:

```
 student_1_dob          = '".mysql_escape($student_1_dob)."',
 student_2_dob          = '".mysql_escape($student_2_dob)."',
 student_3_dob          = '".mysql_escape($student_3_dob)."',
```

Then in the form, the following code (in the appropriate place for your particular form):
NOTE: You only have to define the variables $lowestYear and $highestYear once per viewer.

```
<tr>
     <td>Date of Birth - Student 1</td>
     <td><?php

$lowestYear  = 1920;
$highestYear = 2006;
?>
Month:
<select name="month_1">
  <?php foreach(range(1,12) as $month_1): ?>
    <option value="<?php echo $month_1;?>"><?php echo date("F",strtotime("0000-
$month_1"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_1">
  <?php foreach(range(1,31)as $day_1): ?>
      <option value="<?php echo $day_1;?>"><?php echo $day_1;?></option>
  <?php endforeach ?>
</select> Year:
<select name="year_1">
   <?php foreach (range($lowestYear,$highestYear) as $year_1):?>
       <option value="<?php echo $year_1;?>"><?php echo $year_1;?></option>
   <?php endforeach?>
</select>

   </tr>
```

```
<tr>
     <td>Date of Birth - Student 2</td>
     <td>Month:
<select name="month_2">
  <?php foreach(range(1,12) as $month_2): ?>
    <option value="<?php echo $month_2;?>"><?php echo date("F",strtotime("0000-
$month_2"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_2">
  <?php foreach(range(1,31)as $day_2): ?>
      <option value="<?php echo $day_2;?>"><?php echo $day_2;?></option>
  <?php endforeach ?>
</select> Year:
<select name="year_2">
   <?php foreach (range($lowestYear,$highestYear) as $year_2):?>
       <option value="<?php echo $year_2;?>"><?php echo $year_2;?></option>
   <?php endforeach?>
</select>

   </tr>
```

And,

```
<tr>
     <td>Date of Birth - Student 3</td>
     <td>Month:
<select name="month_3">
  <?php foreach(range(1,12) as $month_3): ?>
    <option value="<?php echo $month_3;?>"><?php echo date("F",strtotime("0000-
$month_3"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_3">
  <?php foreach(range(1,31)as $day_3): ?>
      <option value="<?php echo $day_3;?>"><?php echo $day_3;?></option>
  <?php endforeach ?>
</select> Year:
<select name="year_3">
   <?php foreach (range($lowestYear,$highestYear) as $year_3):?>
       <option value="<?php echo $year_3;?>"><?php echo $year_3;?></option>
   <?php endforeach?>
</select>

   </tr>
```

THE PROFILE UPDATE FORM
For the profile update page, the code at the top of the page, in the //update user section is:

```
 $student_1_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_1'],@$_REQUEST['day_1'],@$_REQUEST['year_1']));
$student_2_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_2'],@$_REQUEST['day_2'],@$_REQUEST['year_2']));
$student_3_dob = date("Y-m-
d",mktime(0,0,0,@$_REQUEST['month_3'],@$_REQUEST['day_3'],@$_REQUEST['year_3']));
```

And the code in the $query = "UPDATE `{$TABLE_PREFIX}accounts` SET section is:

```
student_1_dob          = '".mysql_escape($student_1_dob)."',
student_2_dob          = '".mysql_escape($student_2_dob)."',
student_3_dob          = '".mysql_escape($student_3_dob)."',
```

In the form, in order to pre-populate the individual parts of the date field, Jason came up with the following code:

```
<tr>
     <td>Date of Birth - Student 1</td>
     <td><?php
$lowestYear  = 1935;
$highestYear = 2008;

//pre-populate date fields from the database
  //get most current $CURRENT_USER
  $CURRENT_USER = _websiteLogin_getCurrentUser();

 $_REQUEST['month_1']=date("n",strtotime($CURRENT_USER['student_1_dob']));
  $_REQUEST['day_1']=date("j",strtotime($CURRENT_USER['student_1_dob']));
  $_REQUEST['year_1']=date("Y",strtotime($CURRENT_USER['student_1_dob']));
?>
Month:
<select name="month_1">
  <?php foreach(range(1,12) as $month_1): ?>
   <option value="<?php echo $month_1;?>" <?
php selectedIf($month_1,@$_REQUEST['month_1']);?>><?php echo
date("F",strtotime("0000-$month_1"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_1">
  <?php foreach(range(1,31)as $day_1): ?>
      <option value="<?php echo $day_1;?>" <?
php selectedIf($day_1,@$_REQUEST['day_1']);?>><?php echo $day_1;?></option>

  <?php endforeach ?>
</select> Year:
<select name="year_1">
   <?php foreach (range($lowestYear,$highestYear) as $year_1):?>
       <option value="<?php echo $year_1;?>" <?
php selectedIf($year_1,@$_REQUEST['year_1']);?>><?php echo
$year_1;?></option>
   <?php endforeach?>
</select>

   </tr>
```

```
<tr>
     <td>Date of Birth - Student 2</td>
     <td><?php

//pre-populate date fields from the database
  //get most current $CURRENT_USER
  $CURRENT_USER = _websiteLogin_getCurrentUser();

 $_REQUEST['month_2']=date("n",strtotime($CURRENT_USER['student_2_dob']));
  $_REQUEST['day_2']=date("j",strtotime($CURRENT_USER['student_2_dob']));
  $_REQUEST['year_2']=date("Y",strtotime($CURRENT_USER['student_2_dob']));
?>
Month:
<select name="month_2">
  <?php foreach(range(1,12) as $month_2): ?>
   <option value="<?php echo $month_2;?>" <?
php selectedIf($month_2,@$_REQUEST['month_2']);?>><?php echo
date("F",strtotime("0000-$month_2"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_2">
  <?php foreach(range(1,31)as $day_2): ?>
      <option value="<?php echo $day_2;?>" <?
php selectedIf($day_2,@$_REQUEST['day_2']);?>><?php echo $day_2;?></option>

  <?php endforeach ?>
</select> Year:
<select name="year_2">
   <?php foreach (range($lowestYear,$highestYear) as $year_2):?>
      <option value="<?php echo $year_2;?>" <?
php selectedIf($year_2,@$_REQUEST['year_2']);?>><?php echo
$year_2;?></option>
   <?php endforeach?>
</select>

   </tr>
```

And,

```
<tr>
     <td>Date of Birth - Student 3</td>
     <td><?php

//pre-populate date fields from the database
  //get most current $CURRENT_USER
  $CURRENT_USER = _websiteLogin_getCurrentUser();

 $_REQUEST['month_3']=date("n",strtotime($CURRENT_USER['student_3_dob']));
  $_REQUEST['day_3']=date("j",strtotime($CURRENT_USER['student_3_dob']));
  $_REQUEST['year_3']=date("Y",strtotime($CURRENT_USER['student_3_dob']));
?>
Month:
<select name="month_3">
  <?php foreach(range(1,12) as $month_3): ?>
   <option value="<?php echo $month_3;?>" <?
php selectedIf($month_3,@$_REQUEST['month_3']);?>><?php echo
date("F",strtotime("0000-$month_3"));?></option>
  <?php endforeach ?>
</select> Day:
<select name="day_3">
  <?php foreach(range(1,31)as $day_3): ?>
      <option value="<?php echo $day_3;?>" <?
php selectedIf($day_3,@$_REQUEST['day_3']);?>><?php echo $day_3;?></option>

  <?php endforeach ?>
</select> Year:
<select name="year_3">
   <?php foreach (range($lowestYear,$highestYear) as $year_3):?>
      <option value="<?php echo $year_3;?>" <?
php selectedIf($year_3,@$_REQUEST['year_3']);?>><?php echo
$year_3;?></option>
   <?php endforeach?>
</select>

   </tr>
```

POPULATE A SIGNUP AND PROFILE REVISION FORM WITH EXISTING RADIO BUTTONS VALUE - Feb 18th, 2011

Let’s say you’ve got a radio button field called “interest” on your profile form with a number of possible values and you want to echo the current value from your user's record in their profile revision form.

It’s easy when you follow these 3 simple steps:

- Add a field to your User Accounts editor. (For this example we'll call your field 'interest')

- Add this error checking code to your signup.php or profile.php viewer:

```
 if (!@$_REQUEST['interest']) { $errorsAndAlerts .= "You must select your interest!\n"; }
```

- Add this code to the list of mysql queries:

```
 interest = '".mysql_escape( $_REQUEST['interest'] )."',
```

- Add this user input field to your form:

```
 <tr>
  <td valign="top">Interest</td>
  <td>
    <?php $fieldname = 'interest'; ?>
    <?php $idCounter = 0; ?>
    <?php foreach (getListOptions('accounts', $fieldname) as $value => $label): ?>
     <?php $id = "$fieldname." . ++$idCounter; ?>
     <input type="radio" name="<?php echo $fieldname ?>" id="<?php echo $id ?>"
          value="<?php echo htmlspecialchars($value) ?>" <?
php checkedIf(@$_REQUEST[$fieldname], $value) ?> />
     <label for="<?php echo $id ?>"><?php echo htmlspecialchars($value) ?></label>

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

- Before you go “live”, test that the form lets you save each possible value, displays the last saved value, and that saved values are also displayed correctly in the User Accounts record.

PRE-POPULATING RADIO BUTTONS IN FORMS FROM A MASTER LIST - Jul 22nd, 2019

The trick here was to populate the list of values from the values available in separate master list, and use that list to populate a radio button list field in the accounts section (some of the functions in this example require the Website Membership plugin).

Here’s are the steps...

1) Create a multi-
record section (called “my_values”) with a 'hidden' check box and a text field called “contents”

2) In the accounts section, create a radio button list field (called my_list_field) that gets its options from the MySQL query:

```
SELECT num, contents
 FROM `<?php echo $TABLE_PREFIX ?>my_values`
WHERE hidden = 0
ORDER BY dragSortOrder  ASC ;
```

3) Near the top of the viewer that contains your form, after the load records calls, add the following variable to the list of variables:

```
@$my_list_field = @$_REQUEST['my_list_field'] ;
```

4) After any error checking include the following in your $colsToValues array

```
@$colsToValues['my_list_field']         = $_REQUEST['my_list_field'] ;
```

5) to pre-populate your form from current user values in the accounts table

```
 // pre-populate form with current user values
 foreach ($CURRENT_USER as $name => $value) {
   if (array_key_exists($name, $_REQUEST)) { continue; }
   $_REQUEST[$name] = $value;
 }
```

5) In your form, to show a radio button list of all values that are not hidden in the "my_values" section, use the code:

```
<tr>
```

```
<td valign="top">My Available Values</td>
<td>
<?php $fieldname = 'your_list_field'; ?>
<?php $idCounter = 0; ?>
<?php foreach (getListOptions(accountsTable(), $fieldname) as $value => $label): ?>
<?php $id = "$fieldname." . ++$idCounter; ?>
<input type="radio" name="<?php echo $fieldname ?>" id="<?php echo $id ?>"
 value="<?php echo htmlencode($value) ?>" <?
php checkedIf(@$_REQUEST[$fieldname], $value) ?> />
<label for="<?php echo $id ?>"><?php echo htmlencode($label) ?></label>
<?php endforeach ?>
</td>
</tr>
```

Another Approach (shows all values):

```
<?php $fieldname = 'practice_country'; ?>
<?php $idCounter = 0; ?>
<?php foreach (getListOptions(accountsTable(), $fieldname) as $value => $label): ?>
<?php $id = "$fieldname." . ++$idCounter; ?>
<input id="<?php echo $id ?>" name="<?php echo $fieldname ?>" type="radio" value="<?
php echo htmlencode($value) ?>" />
<label for="<?php echo $id ?>"><?php echo htmlencode($label) ?></label>

<?php endforeach ?>
```

<b>To show only values that are not hidden</b>:
Since getListOptions() will get all available options for a list field, if you want to only show countries that have been selected by non-hidden records, here's a different approach.

```
<?php $fieldname = 'practice_country'; ?>
<?php $tablename = accountsTable(); ?>
<?php
  // get all table records
  list($tableRecords, $tableRecordsMeta) = getRecords(array(
    'tableName'     => $tablename,
    'loadUploads'   => false,
    'loadCreatedBy' => false,
    'allowSearch'   => false,
  ));

  // group records by target field to find option keys
  $tableRecordsByField = array_groupBy($tableRecords, $fieldname);
  $keys                = array_keys($tableRecordsByField);

  // create option list
  $listOptions = [];
  foreach($keys as $key) {
    if (!empty( $key )) {
      $listOptions[ $key ] = $tableRecordsByField[ $key ][ $fieldname.':label' ];
    }
  }
?>
```

This will create the variable $listOptions that should contain all of the currently in-use list values and labels for
the target table/field. This should let you then update your foreach to look like this:

```
<?php foreach ($listOptions as $value => $label): ?>
              <?php $id = "$fieldname." . ++$idCounter; ?>
              <input id="<?php echo $id ?>" name="<?php echo $fieldname ?>" type="radio" value="
<?php echo
htmlencode($value) ?>" <?php checkedIf(@$_REQUEST[$fieldname], $value) ?>/>
              <label for="<?php echo $id ?>"><?php echo htmlencode($label) ?></label>

              <?php endforeach ?>
```

Note that this will only work for single-value list fields, such as radio buttons or single dropdowns.

WORKING WITH SINGLE VALUE CHECK BOXES IN FORMS - Dec 23rd, 2015

Here's how to add a checkbox (called non_member) to a form and have it populate a single value check box field in your multi record editor

(Of course, you'll need to have a single value check box field called non_member in your multi record editor)

Check boxes are handled a bit differently from text field, so here's the code you'd need.

First create a variable (makes it easier to keep track of your code)

```
$non_memberCheckBox = intval(@$_REQUEST['non_member']);
```

Then, in your INSERT INTO or UPDATE mysql_query:

```
non_member   = '".intval( $non_memberCheckBox )."',
```

Then in your form you would add:

```
<label for='non_member><span class="your_class">Non-Member:</span>&nbsp; </label>
      <input type = "checkbox" id="non_member" name="non_member" value = "1" <?
php checkedIf(1,
@$_REQUEST['non_member']);?> />
```

So that if the box is checked your non_member field will be populated with a value of "1" If not it will be populated with "0"

Another iteration of this is when $colsToValues is used (like in the profile update and user signup form)

When a browser submits a form if a checkbox is not ticked it doesn't send a value at all as opposed to sending an empty or negative value.

To get around this issue, in the $colsToValues array, insert:

```
$colsToValues['non_member']    = (@$_REQUEST['non_member'])? '1' : '0' ;
```

Which sends a '0' when the box is unchecked.

In your form, your input field would look the same as above:

```
<input type = "checkbox" id="non_member" name="non_member" value = "1" <?
php checkedIf(1, @$_REQUEST['non_member']);?>
```

WORKING WITH MULTI VALUE CHECK BOXES IN FORMS

WORKING WITH MULTI VALUE CHECK BOXES IN FORMS - Dec 29th, 2015

According to Jason Sauchuck, from Interactive Tools, Multi-value checklists and multi-
value pull downs are both stored
the same way in CMS Builder,

In this example I'm populating the list of possible option values (num) and option labels (text) for the levels_of_care field from another table

Using the basic approach in the website membership plugin's profile update form, and with some help fro Ross Fairbairn, also from Interactive Tools, I was able to populate a multi value check box list field (levels_of_care in this example), and update the list in my table (accounts in this case),use the following code in my form:

```
<tr>
            <td valign="top">Levels of Care</td>
            <td><?php $fieldname = 'levels_of_care'; ?>
```

```
              <?php
if(is_array(@$_REQUEST[$fieldname])){
$fieldValues = $_REQUEST[$fieldname];
}
else{
$fieldValues = explode("\t",trim(@$_REQUEST[$fieldname],"\t"));
}
?>
               <?php $idCounter = 0; ?>
               <?php foreach (getListOptions('accounts', $fieldname) as $value => $label): ?>
               <?php $id = "$fieldname." . ++$idCounter; ?>
               <input type="checkbox" name="<?php echo $fieldname ?>[]" id="<?
php echo $id ?>"
value="<?php echo htmlspecialchars($value) ?>" <?php if(in_array($value,$fieldValues)):?
> checked="checked" <?php endif
?>/>
               <label for="<?php echo $id ?>"><?php echo htmlspecialchars($label) ?></label>


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

And this code in the colsToValues array:

```
   $colsToValues['levels_of_care'] = "\t" . implode("\t", $_REQUEST['levels_of_care']) . "\t";
```

CREATING DYNAMIC FORM FIELDS - Nov 10th, 2018

I was creating a front end form that allowed companies that had more than one location and wanted to list those locations without cluttering up the form with lots of blank fields.

Here’s how...

1) In the head of your form page add the following javascript:

```
<script type="text/javascript">

function yesnoCheck1() {
   if (document.getElementById('yesCheck1').checked) {
      document.getElementById('ifYes1').style.display = 'block';
   }
   else document.getElementById('ifYes1').style.display = 'none';

}
function yesnoCheck2() {

   if (document.getElementById('yesCheck2').checked) {
      document.getElementById('ifYes2').style.display = 'block';
   }
   else document.getElementById('ifYes2').style.display = 'none';

}

function yesnoCheck3() {
   if (document.getElementById('yesCheck3').checked) {
      document.getElementById('ifYes3').style.display = 'block';
   }
   else document.getElementById('ifYes3').style.display = 'none';

}
</script>
```

Then in the form itself, add the following:

```
 <form>
             <table align="center" >
                 <tr>
                   <td class="text_font" >Location 1 Street Address</td>
                   <td><input  type="text" name="location_street_address_1" value="<?
php echo
ucfirst(htmlencode(@$_REQUEST['location_street_address_1'])); ?>"  size="50" /></td>
                 </tr>
                 <tr>
                   <td class="text_font" >Apt/Suite/Floor</td>
                   <td><input  type="text" name="location_suite_apartment_or_floor_1" value="
<?php echo
ucfirst(htmlencode(@$_REQUEST['location_suite_apartment_or_floor_1'])); ?
>" size="50" /></td>
                 </tr>
                 <tr>
                   <td class="text_font">City</td>
                   <td><input type="text" name="location_city_1" value="<?php echo
htmlspecialchars(@$_REQUEST['location_city_1']); ?>" size="50" /></td>
```

```
                </tr>
                <tr>
                  <td class="text_font">State</td>
                  <td><input type="text" name="location_state_1" value="<?php echo
htmlspecialchars(@$_REQUEST['location_state_1']); ?>" MAXLENGTH="1" size="50" />
</td>
                </tr>
                <tr>
                  <td class="text_font">Zip Code</td>
                  <td><input type="text" name="location_zip_1" value="<?php echo
htmlspecialchars(@$_REQUEST['location_zip_1']); ?>" MAXLENGTH="10" size="50" />
</td>
                </tr>
              </table>
              <span class="text_font"><b>
              <hr />
              Add A Second Location?
              <input type="radio" onclick="javascript:yesnoCheck1();" name="yesno1" id="yesCheck1">

              <hr />
              </b></span>
              <div id="ifYes1" <?php if(@!$_REQUEST['location_street_address_2']):?
>style="display:none" <?php endif
?>>
                <table align="center" >
                  <tr>
                    <td class="text_font" >Location 2 Street Address</td>
                    <td><input  type="text" name="location_street_address_2" value="<?
php echo
ucfirst(htmlencode(@$_REQUEST['location_street_address_2'])); ?>"  size="50" /></td>
                  </tr>
                  <tr>
                    <td class="text_font" >Apt/Suite/Floor</td>
                    <td><input  type="text" name="location_suite_apartment_or_floor_2" value="
<?php echo
ucfirst(htmlencode(@$_REQUEST['location_suite_apartment_or_floor_2'])); ?
>" size="50" /></td>
                  </tr>
                  <tr>
                    <td class="text_font">City</td>
                    <td><input type="text" name="location_city_2" value="<?php echo
htmlspecialchars(@$_REQUEST['location_city_2']); ?>" size="50" /></td>
                  </tr>
                  <tr>
                    <td class="text_font">State</td>
                    <td><input type="text" name="location_state_2" value="<?php echo
htmlspecialchars(@$_REQUEST['location_state_2']); ?>" MAXLENGTH="2" size="50" />
</td>
                  </tr>
                  <tr>
                    <td class="text_font">Zip Code</td>
                    <td><input type="text" name="location_zip_2" value="<?php echo
htmlspecialchars(@$_REQUEST['location_zip_2']); ?>" MAXLENGTH="10" size="50" />
</td>
                  </tr>
                </table>
                <span class="text_font"><b>
                <hr />
                Add A Third Location?
                <input type="radio" onclick="javascript:yesnoCheck2();" name="yesno2" id="yesCheck2">

                <hr />
                </b></span>
                <div id="ifYes2" <?php if(@!$_REQUEST['location_street_address_3']):?
>style="display:none" <?php endif
?>>
                  <table align="center" >
                    <tr>
                      <td class="text_font" >Location 3 Street Address</td>
                      <td><input  type="text" name="location_street_address_3" value="<?
php echo
ucfirst(htmlencode(@$_REQUEST['location_street_address_3'])); ?>"  size="50" /></td>
                    </tr>
                    <tr>
                      <td class="text_font" >Apt/Suite/Floor</td>
                      <td>
<input  type="text" name="location_suite_apartment_or_floor_3" value="<?php echo
ucfirst(htmlencode(@$_REQUEST['location_suite_apartment_or_floor_3'])); ?
>" size="50" /></td>
                    </tr>
                    <tr>
                      <td class="text_font">City</td>
                      <td><input type="text" name="location_city_3" value="<?php echo
htmlspecialchars(@$_REQUEST['location_city_3']); ?>" size="50" /></td>
                    </tr>
                    <tr>
                      <td class="text_font">State</td>
                      <td><input type="text" name="location_state_3" value="<?php echo
htmlspecialchars(@$_REQUEST['location_state_3']); ?>" MAXLENGTH="2" size="50" />
</td>
```

```
                    </tr>
                    <tr>
                      <td class="text_font">Zip Code</td>
                      <td><input type="text" name="location_zip_3" value="<?php echo
htmlspecialchars(@$_REQUEST['location_zip_3']); ?>" MAXLENGTH="10" size="50" />
</td>
                    </tr>
                  </table>

                  </div>
              </div>
              </form>
```

To add new fields, at the end of each block of location entry fields there's an “Add another location?” radio button that added another block of location entry fields.

**To add more blocks of fields, add new functions to the javascript and add new blocks to the form. Just make sure to add closing </div> tags and to increment the function numbers.

The <?php if(@!$_REQUEST['location_street_address_n']):?>style="display:none" <?php endif ?> code insures that if there is data entered in that block, the block will remain visible for updating.

PASTE AWARE WORD COUNTER WITH WORD LIMITING FOR FORM TEXTAREA FIELDS - Dec 13th, 2018

Here’s a simple to implement word counter to use when you want to limit the length of the input to a front end html form textarea field.

In the head of the web page insert the following:
(Change the number of words to suit your needs)

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js">
</script>
 <script type="text/javascript"><!--

$(document).ready(function() {
  $("#word_count").on('keyup', function() {
    var words = this.value.match(/\S+/g).length;

    if (words > 200) {
      // Split the string on first 200 words and rejoin on spaces
      var trimmed = $(this).val().split(/\s+/, 200).join(" ");
      // Add a space at the end to make sure more typing creates new words
      $(this).val(trimmed + " ");
    }
    else {
      $('#display_count').text(words);
      $('#word_left').text(200-words);
    }
  });
});
// --></script>
```

Then, for the textarea field in the form, use something like:

```
<textarea name="your_name" id="word_count" COLS=50 ROWS=8></textarea>

                    Total word Count : <span id="display_count">0</span> words. Words left : <span
id="word_left">200</span>
                    If you paste more than 200 words your entry will be truncated.
```

PRE-POPULATE A PULL DOWN FORM FIELD FROM THE VALUES IN A MASTER LIST - Jun 3rd, 2014

I had created a multi-record master list section called master_list with a field called title. I wanted to pre-populate the options values in a form from the values in the title field of the records in that section.

At the top of my page I insert a load records call for the master_list table

```
 list($master_listRecords, $master_listMetaData) = getRecords(array(
    'tableName'   => 'master_list',
  ));
```

In the body, before the form, I created an array out of the title field values in each record.

```
<?php
 $names = array();
 foreach ($master_listRecords as $record){
    $names[$record['title']]=$record['title'];
 }
?>
```

In the form I inserted the following to loop through the array and populate the pull down field:

```
 <label for='exhibition_name'><span class="my_class">Exhibition Name:</span>
      </label>
<select  type="text" id="exhibition_name" name="exhibition_name">
<option value=""><span class="body-text-bold">Select an Exhibition Name</span>
</option>

<?php foreach($names as $name): ?>
   <option value="<?php echo $name;?>"><?php echo $name;?></option>
<?php endforeach?>
</select>
```

Here’s another approach using the record number as the option value for a record in a table called ‘email_signup_location’ that contains a field called ‘location’.

The Javascript retains the selected value in the form after submission so that if there are any errors thrown, the pull down does not need to be manually repopulated.

```
<select type="text"  id="source" name="source" width="200" style="width: 200px" >
<option value="" >Your Location</option>
                    <?
php foreach(mysql_select("email_signup_location") as $email_signup_locationRecord): ?>
                    <option value="<?php echo $email_signup_locationRecord['num'];?>"><?
php echo
$email_signup_locationRecord['location'];?></option>
                    <?php endforeach?>
</select>
<script type="text/javascript">
  document.getElementById('source').value = "<?php echo $_POST['source'];?>";
</script>
```

CREATING A LOGIN LOG TO SEE WHICH MEMBERS SIGN IN AND WHEN (UPDATED FOR MYSQLI) - Jul 22nd, 2019

Chris Waddell from Interactive Tools offered this approach:

NOTE: Records will only be added to the "Login Log" section for actual membership logins, not by logins to the CMSB interface.

1) Create a Multi Record section called "Login Log".

2) Modify it and remove all the fields except "num" and "createdDate" (note that you'll need to "Enable System Field Editing" under the "Advanced Commands..." dropdown to remove some of the fields.)

3) Add a List Field called "Who". Leave "Display As" set to "pull down".

Modify your "Who" field as follows:

List Options: Get options from database (advanced)

  Section Tablename: accounts
  Use this field for option values: num
  Use this field for option labels: username

4) Create a list field for any other fields in the user account that you'd like to display. I have a First Name and a Last Name field, so I've created a "First Name" list field and a "Last Name" list field as above.

Each has the same list options except for the "Use This Field For Option Labels" which I changed to "first_name" and "last_name".

If you want to add other fields to the log, just create more list fields as above, but change the "Use This Field For Option Labels" as appropriate.

4) Now, make some more changes to your Login Log section, this time at the top of the page:

In the General tab, change "ListPage Fields" to "createdDate, who, first_name, last_name".
Under the Viewer Urls tab, delete all the existing "Filename Fields" entries.
Under the Searching tab, set "Search Fields" to "createdDate, who, first_name, last_name".
Finally, under the Sorting tab, set "Order By" to "createdDate DESC". Now click Save Details.

5) Now open up cmsAdmin/plugins/websiteMembership/websiteMembership.php in a text editor and find "redirect on success". Add the following code immediately before "redirect on success":

```
  // CUSTOM CODE! add record to login_log updated for mysqli
  global $TABLE_PREFIX;
 mysqli()->query(mysql_escapef("INSERT INTO {$TABLE_PREFIX}login_log SET
createdDate = NOW(),
who = ?, last_name = ?,
 first_name = ?",
$CURRENT_USER['num'],
$CURRENT_USER['num'],
$CURRENT_USER['num']))
or die("Mysql error adding login_log record: ". htmlspecialchars(mysqli()->error) . "\n");
```

If you don't want to log admins, us this code instead:

```
// CUSTOM CODE! add record to login_log updated for mysqli
 if (!@$CURRENT_USER['isAdmin'])
 {global $TABLE_PREFIX;
mysqli()->query(mysql_escapef("INSERT INTO {$TABLE_PREFIX}login_log SET
createdDate = NOW(),
 who = ?, last_name = ?,
 first_name = ?",
$CURRENT_USER['num'],
$CURRENT_USER['num'],
$CURRENT_USER['num']))
or die("Mysql error adding login_log record: ". htmlspecialchars(mysqli()->error) . "\n");
 }
```

In the Custom Code add (or remove) a your_field = ?, and a $CURRENT_USER['num'], for each new field you want to populate.

Now you can create a list viewer that your client can access and restrict viewer access to admins only.

Here's a simple example that you can style to match your site design.

At the top of your listing page:
NOTE: Don't forget to change the $dirsToCheck = array('/PATH_TO_YOUR_SERVER/','','../','../../','../../../'); to match your server path. (you can find this in the admin>code generator code for any section.

```
<?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($login_logRecords, $login_logMetaData) = getRecords(array(
    'tableName'   => 'login_log',
  ));
  if (!$CURRENT_USER['isAdmin']) { websiteLogin_redirectToLogin(); }
?>
<!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>Login Log</title>
</head>
<body>

<?php foreach ($login_logRecords as $record): ?>
<?php echo date("F jS  Y, g:i a ", strtotime($record['createdDate'])) ?>

<?php echo $record['who:label'];?> - <?php echo $record['first_name:label'];?> <?
php echo $record['last_:label'];?>
<hr/>
<?php endforeach ?>
</body>
</html>
```

AUTOMATIC RENEWALS USING THE WEBSITE MEMBERSHIP PLUGIN - Jan 31st, 2023

AUTOMATIC RENEWALS USING THE WEBSITE MEMBERSHIP PLUGIN - Jan 31st, 2023

This is a rather long recipe, covering the process of automatically renewing memberships based on a specified membership period.

There are a number of pieces to this puzzle:

Setting the expiration date when a member first signs up and creates their member record

Determining when a renewal has expired

Directing the member to a renewal page for payment

After payment, updating the member record to reflect the “new” expiry date so the process can repeat automatically.

Automatically sending a renewal letter on expiration

SETTING THE INITIAL EXPIRATION DATE AND DETERMINING WHEN A RENEWAL HAS EXPIRED

Since my membership period is 1 year, when a member signs up, the “expiresDate” field in the accounts database is automatically populated with a date that is 1 year from the date that the member record was created. I also set the “neverExpires” checkbox to zero using this code in my sign-up form. (you can read more about sign-up form modification elsewhere in this chapter).

```
expiresDate = (NOW() + INTERVAL 1 YEAR),
neverExpires     = '0',
```

NOTE: Although I could have used the neverExpires field, to deal with special memberships that should never expire, I chose to create a “special_membership” check box field in the database.

REDIRECTING EXPIRED MEMBERS TO A RENEWAL PAGE

Thanks to some advice from Dave Edis from Interactive Tools, I inserted this code near the top of all of the “members only” pages (after the login_redirect) to redirect members to a “renew_now” page if their membership had expired.

```
<?php if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); } ?>

<?php
$isMembershipExpired = strtotime($CURRENT_USER['expiresDate']) <= time();
$notSpecial = ($CURRENT_USER['special_membership'] == 0);
if ($isMembershipExpired && $notSpecial) { redirectBrowserToURL("renew_now.php"); }
?>
```

The renewal page includes a PayPal link and I use the program “Linklok” to handle transaction verification and to redirect the user to a hidden “thank you” page on the site after a successful payment.

THE RENEW_NOW.PHP CODE

At the top of the page:

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php
 $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
  $dirsToCheck = array('server_path_to_your_root_directory/','','../','../../','../../../');
  foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }}
  if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }
 ?>

<?
php if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); } ?>
```

And in the body:

```
<div align="center">
<br /><h1><span class="Page-Titles">Thank's For</span><br />
<span class="Page-Titles">
Being a Subsciber!</span></h1>
 <h3><span class="X-Large-Text">Unfortunately, your subscription has expired,<br />
 and you'll need to renew your subscription to enjoy <br />
another full year of uninterupted access to the Cookbook</span>.</h3>
<h2><span class="X-Large-Text">Great News! </span></h2>

 <h3><span class="X-Large-Text">If you renew now with PayPal, <br />
</span><span class="X-Large-Text"> <br />
your special annual renewal rate is only <br />
</span></h3>
<h2><span class="X-Large-Text">$15.00 US!</span></h2>
<h3><span class="X-Large-Text">which is $40% off the current subscription rate. <br />
</span></h3>
<table width="75%" border="0" cellpadding="15">
```

```
<tr>
<td valign="top"><div align="center"> <img src="images/paypal-
logo.jpg" alt="" height="99" width="100" border="0"
/></div></td>
<td valign="top"><p align="center"> </p>
<div align="center"> <a class="special" href="https://www.paypal.com/cgi-bin/webscr?
amount=15.00
&amp;item_number=Your_Item_Number
&amp;item_name=Your Renewal
&amp;business=payments@your_site.com
&amp;cpp_header_image=http://www.your_site.com/images/CMSBPP.png
&amp;currency_code=USD&amp;lc=US&amp;add=1
&amp;cmd=_cart&amp;no_shipping=0
&amp;return=http://www.your_site.com/linklokipnret.php
&amp;notify_url=http://www.your_site.com/linklokipn.php
&amp;no_note=1&amp;bn value=PP-ShopCartBF&amp;">
<img src="/images/btn_buynowCC_LG.gif" /></a><br />
</div></td>
</tr>  <tr>
 <td align="center" valign="middle"><span class="Large-
Text">Renew you subscription now, for only <s>$24.95</s>
</span><span class="X-Large-Text">$15.00</span></td>
</tr>
</table>
```

Linklok implementation proved to be easy.

The URL for Linklok is:

http://www.vibralogix.com/linklokipn/

If you should decide to purchase linklok, or any other Vibralogic programs, I'd appreciate your using my affiliate link:

http://www.shareasale.com/r.cfm?B=12671&U=520135&M=3826

Although the documentation outlines many ways to customize your installation, for a basic implementation there are only two files required for linklok to operate: linklokipn.php and linklokipnret.php

linklokipn.php is the only file that you'll need to customize, and all the user defined parameters appear at the top of the file:

The information included in the $Products variable is: "unique_product_id,product name,product currency and minimum price accepted,path to your hidden directory and the URL of your "your_update_page.php" page, leave this one set to "0",the number of minutes until the link sent to the purchaser expires";

```
$Products[] = "Your_Product_ID,Your Product name,USD=99.95,soyerveortmvobd/your_update_page.php,0,1440";

// Setup admin and security variables
$LinklokURL = "http://www.your_site.com/linklokipn.php";      // URL of the linlok.php page on your site
$PaypalEmail = "payments@your_site.com";                      // PRIMARY Paypal email address
$SellerCompany = "Your Company";                              // Your company name
$SellerEmail = "info@your_site.com";                          // Your email address for order inquiries
$SellerURL = "http://www.your_site.com";                      // Your website URL
$LinkKey = "orange23";                                        // Encryption key for download links
$PDTtoken="12876Vdwerwrol_dmYMHfewrwurjdE3QsupkkvTo4Fv39-
7y";       // Auto Return PDT token from paypal-profile-website
payment preferences

$DelayEchecks = "Y";                                          // Set to Y to delay eCheck orders until cleared.
$EmailTemplate = "";                                          // Optional Email Template in either .txt or .html format
$DownloadTemplate = "";                                       // Optional download page template
$ErrorTemplate = "";                                          // Optional Error page template
$CopyEmail = "any_email@your_provider.net";                                       // Receive copy of order emails.
email address or ""
$ManualPassword = "kookamonga";                               // Password for manual order entry. "" to disable
$HTMLEmail = "Y";                                             // Set to Y to use HTML formatted emails or N to send in
plain-text
$Txnid = "";                                                  // Optional to stop possible multiple calls from IPN
$WarningTemplate = "";                                        // Check payment warning email template
$WarningTemplatePage= "";                                     // Check payment warning thankyou page template
```

UPDATING THE MEMBER'S RECORD (The soyerveortmvobd/your_update_page.php file)

Current versions of CMSB include password encryption, so matching an encrypted password with a submitted plain text password is not possible. Fortunately, it is possible to use the built in login functionality to make the hidden update function pretty straight forward:

```
<?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."); }
```

```
 if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); }
 if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); }  ?>
<?php
  mysqlStrictMode(false);
 mysql_query("UPDATE `{$TABLE_PREFIX}accounts`
    SET  expiresDate = NOW() + INTERVAL 1 YEAR
   WHERE num = '".mysql_escape( $CURRENT_USER['num'] )."'")
      or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
       $userNum = mysql_insert_id();
    // redirect after a successful update
    header("Location: http://www.your_site.com/your_update_success_page.php");
 ?>
<!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>
</body>
</html>
```

AUTOMATING THE RENEWAL PROCESS WITH AUTOMATIC E-MAILS
Although my specific application did not require notification prior to membership expiration, you might want to send out mailings a few weeks before a member's account expires.

I did not implement this, but here are some preliminary manual steps you can use:

GENERATING EMAIL LISTS OF EXPIRING MEMBERS
Here's the code for the top of your page:

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?php
    // load viewer library
  $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
  $dirsToCheck = array('/server_path_to_your_root_directory/','','../','../../','../../../');
  foreach ($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($accountsRecords, $accountsMetaData) = getRecords(array(
    'tableName'   => 'accounts',
'where' => " '{$CURRENT_USER['isAdmin']}' ",
  ));

?>
```

The code: 'where' => " '{$CURRENT_USER['isAdmin']}' ", prevents anyone who is not an admin from generating an email list.

And in the body of the page:
The trick is to only list emails for those non-special members whose accounts have expired.
Using an if statement with a further modification of Dave's Expired Account checking code solved that challenge:

```
 <hr />
<div align="left">
?php foreach ($accountsRecords as $record): ?>
<?php  $isMembershipExpired = strtotime ($record['expiresDate']) <= time(); ?>
<?php if ($record['special_membership'] == '0' && $isMembershipExpired == '1'): ?>
<span class="your_class"><?php echo $record['email'] ?>;&nbsp;</span>
<?php endif ?>
<?php endforeach ?>
</div>
<hr />
```

If you want to send emails to different lists of expired members here are some approaches:

GENERATE A LIST OF MEMBERS WHOSE ACCOUNTS HAVE RECENTLY EXPIRED
You can compare the updatedDate value with the current date and only generate a list of those member records that have expired and who fall between the two dates.

GENERATE A LIST OF EXPIRED MEMBERS THAT WERE SENT A LETTER AND DIDN'T RENEW
Here you'll need a field that contains the last date that you sent out a letter so you can compare that date with those members whose accounts expired before that date. (Yu can also generate a list of those who expired after that date.) The Last Mailing Date can be in a single record editor if you have one.

If your service provider limits the amount of email recipients that can be included in one email, you can break your lists down into groups using the information in the recipe “BREAKING E-MAIL LISTS INTO GROUPS OF "N" ADDRESSES EACH”

DOUBLE OPT IN MAIL LIST SYSTEM IN CMSB (PHP7 AND GOOGLE RECAPTCHA) - Sep

7th, 2017

Here’s the updated double opt in email signup system for PHP7 and Google Recaptcha. I hope it proves helpful...

There are 4 pages that act on a section called email_signup:

email_signup.php - the initial signup form that creates a record in the database with the hidden field checked and the confirmed field unchecked. The page also sends a confirmation email to the email entered with a link that the recipient can click on to confirm that they wanted to sign up.

confirm.php - the confirmation page that the person accesses through the link in their email to confirm that they wanted to sign up.

unsubscribe.php - the form used to unsubscribe. As above a confirmation email is sent to the email entered, and no action is taken until the recipient clicks on the link in the email.

unsubscribe_confirm.php - as above, the page that the person addresses through the link in their email to confirm that they wanted to unsubscribe.

Here’s the .ini.php code for creating the email_signup section, email_signup,ini.php

```
<?php  if (!@$LOADSTRUCT) { die("This is not a program file."); }
return array (
  '_detailPage' => '',
  '_disableAdd' => '0',
  '_disableErase' => '0',
  '_disableModify' => '0',
  '_disablePreview' => '1',
  '_disableView' => '1',
  '_filenameFields' => '',
  '_hideRecordsFromDisabledAccounts' => '0',
  '_indent' => '0',
  '_listPage' => '',
  '_maxRecords' => '',
  '_maxRecordsPerUser' => '',
  '_perPageDefault' => '1000',
  '_previewPage' => '',
  '_requiredPlugins' => '',
  '_tableName' => 'email_signup',
  'listPageFields' => 'last_name,first_name,email,hidden,confirmed,remove,source,createdDate,updatedDate',
  'listPageOrder' => 'source DESC, confirmed DESC, last_name, first_name',
  'listPageSearchFields' => '__ALL__',
  'menuHidden' => '0',
  'menuName' => 'Email Signup',
  'menuOrder' => '0000000004',
  'menuPrefixIcon' => '',
  'menuType' => 'multi',
  'num' => array(
    'order' => 1,
    'type' => 'none',
    'label' => 'Record Number',
    'isSystemField' => '1',
  ),
  'createdDate' => array(
    'order' => 2,
    'type' => 'none',
    'label' => 'Created',
    'isSystemField' => '1',
  ),
  'createdByUserNum' => array(
    'order' => 3,
    'type' => 'none',
    'label' => 'Created By',
    'isSystemField' => '1',
  ),
  'updatedDate' => array(
    'order' => 4,
    'type' => 'none',
    'label' => 'Last Updated',
    'isSystemField' => '1',
  ),
  'updatedByUserNum' => array(
    'order' => 5,
    'type' => 'none',
    'label' => 'Last Updated By',
    'isSystemField' => '1',
  ),
  'hidden' => array(
    'order' => 6,
    'label' => 'Hidden',
    'type' => 'checkbox',
    'fieldPrefix' => '',
    'checkedByDefault' => '0',
    'description' => '',
    'checkedValue' => 'Yes',
    'uncheckedValue' => 'No',
  ),
  'confirmed' => array(

    'order' => 7,
    'label' => 'Confirmed',
    'type' => 'checkbox',
    'fieldPrefix' => '',
    'checkedByDefault' => '0',
    'description' => '',
    'checkedValue' => 'Yes',
    'uncheckedValue' => 'No',
  ),
  'remove' => array(
    'order' => 8,
    'label' => 'Remove',
    'type' => 'checkbox',
    'fieldPrefix' => '',
    'checkedByDefault' => '0',
    'description' => '',
    'checkedValue' => 'Yes',
    'uncheckedValue' => 'No',
  ),
  'first_name' => array(
    'order' => 9,
    'label' => 'First Name',
    'type' => 'textfield',
    'defaultValue' => '',
    'fieldPrefix' => '',
    'description' => '',
    'fieldWidth' => '',
    'isPasswordField' => '0',
    'isRequired' => '0',
    'isUnique' => '0',
    'minLength' => '',
    'maxLength' => '',
    'charsetRule' => '',
    'charset' => '',
  ),
  'last_name' => array(
    'order' => 10,
    'label' => 'Last Name',
    'type' => 'textfield',
    'defaultValue' => '',
    'fieldPrefix' => '',
    'description' => '',
    'fieldWidth' => '',
    'isPasswordField' => '0',
    'isRequired' => '0',
    'isUnique' => '0',
    'minLength' => '',
    'maxLength' => '',
    'charsetRule' => '',
    'charset' => '',
  ),
  'email' => array(
    'order' => 11,
    'label' => 'Email',
    'type' => 'textfield',
    'defaultValue' => '',
    'fieldPrefix' => '',
    'description' => '',
    'fieldWidth' => '',
    'isPasswordField' => '0',
    'isRequired' => '0',
    'isUnique' => '0',
    'minLength' => '',
    'maxLength' => '',
    'charsetRule' => '',
    'charset' => '',
  ),
  'source' => array(

    'order' => 12,
    'label' => 'Source',
    'type' => 'list',
    'fieldPrefix' => '',
    'description' => '',
    'isRequired' => '0',
    'isUnique' => '0',
    'listType' => 'pulldown',
    'optionsType' => 'text',
    'optionsText' => 'Exhibition
Meeting
Street Fair
Web Site',
  ),
);
?>
```

emailsignup.php

The code at the top of the page, above the head, after the records calls required for your site:

```
<?php
  // load records from 'email_signup'
  list($email_signupRecords, $email_signupMetaData) = getRecords(array(
    'tableName'   => 'email_signup',
    'loadUploads' => true,
    'allowSearch' => false,
  ));
?>
<?php $signup_email = ' the_email_address_you_want_to_use_for_return_and_reply' ?>
<?php
if (@$_REQUEST['submit']) {

function validateGoogleCaptcha(){

    $errorsAndAlerts = "";

    if (!@$_REQUEST['g-recaptcha-
response'])     { $errorsAndAlerts .= "Please check the anti-spam 'I am not a robot'
checkbox!\n";
    $showSignupForm = true; // don't change this value
    }
    else {
      // check recaptcha
      $postdata = array();
      $postdata['secret']   = 'your Google Recaptcha secret code';
      $postdata['response'] = @$_REQUEST['g-recaptcha-response'];
      $postdata['remoteip'] = $_SERVER['REMOTE_ADDR'];
      $url = "https://www.google.com/recaptcha/api/siteverify?". http_build_query($postdata, '', '&');
      list($json, $httpStatusCode, $headers, $request) = getPage($url, 5, '', true);
      $recaptchaResponse = json_decode($json, true);

      if (!$recaptchaResponse['success']) {
        if (is_array($recaptchaResponse['error-codes'])) {
          if (in_array('missing-input-secret', $recaptchaResponse['error-
codes']))     { $errorsAndAlerts .= "There's a
problem with recaptcha, please let us know! (no secret)\n"; }
          if (in_array('invalid-input-secret', $recaptchaResponse['error-
codes']))     { $errorsAndAlerts .= "There's a
problem with recaptcha, please let us know! (invald secret)\n"; }
          if (in_array('missing-input-response', $recaptchaResponse['error-
codes']))   { $errorsAndAlerts .= "Please
fill out the recaptcha box!\n";
          $showSignupForm = true; // do we need this line?
           }
          if (in_array('invalid-input-response', $recaptchaResponse['error-
codes']))   { $errorsAndAlerts .= "Please
fill out the recaptcha box again, your answer was incorrect!\n";
          $showSignupForm = true; // do we need this line?
          }
        }
        if (!@$errorsAndAlerts) { @$errorsAndAlerts .= "Invalid captcha response, please try again or contact us
directly and let us know."; }
        @trigger_error("Failed recaptcha on signup form", E_USER_NOTICE);
      }
    }
    return $errorsAndAlerts;
  }
}
?>
<?php
// submit form
if (@$_REQUEST['submit']) {
 @$errorsAndAlerts .= validateGoogleCaptcha();
  // error checking

  if (!@$_REQUEST['first_name'])    { $errorsAndAlerts .= "Please enter your first name\n"; }
  if (!@$_REQUEST['last_name'])  { $errorsAndAlerts .= "Please enter your last name\n"; }
    if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter your email address\n"; }

// email checking
    if ($_REQUEST['email'] || $_REQUEST['email2']) {
      if (!@$_REQUEST['email'])                              { $errorsAndAlerts .= "Please enter your email address\n"; }
      elseif (!@$_REQUEST['email2'])                              { $errorsAndAlerts .= "Please re-
enter your email
address\n"; }
      elseif ($_REQUEST['email'] != $_REQUEST['email2']) { $errorsAndAlerts .= "Sorry, the e mail addresses you entered
don't match!\n"; }
    }

 // check for duplicate emails
    if (!$errorsAndAlerts) {

      $count = mysql_select_count_from('email_signup', "'".mysql_escape($_REQUEST['email'])."'" IN (email)");
      if ($count > 0) { $errorsAndAlerts .= "That email address is already signed up, please choose another!\n"; }
    }
  // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

  // add record
  if (!@$errorsAndAlerts) {
    $tablename   = 'email_signup';
      $colsToValues = array();
      $colsToValues['createdDate=']     = 'NOW()';
      $colsToValues['updatedDate=']     = 'NOW()';
      $colsToValues['createdByUserNum'] = 0;
      $colsToValues['updatedByUserNum'] = 0;
      $colsToValues['first_name']         = $_REQUEST['first_name'];
      $colsToValues['last_name']         = $_REQUEST['last_name'];
      $colsToValues['email']           = $_REQUEST['email'];
      $colsToValues['hidden'] = 1;
      $hideMissingFieldErrors = true;
      $newRecordNum = mysql_insert($tablename, $colsToValues, $hideMissingFieldErrors);

    // display thanks message and clear form
    $errorsAndAlerts = "Thanks for submitting your information.  Before we can add your email address to our list,
you'll need to confirm your intent by clicking on the link in the email that you will receive shortly.  If you do not
see the email, check your spam folder.";


    // send email to applicant
    $to=$_REQUEST['email'];

$subject = 'Email List Signup Request';
$headers = "From: $signup_email" . "\r\n";
$headers .= "Reply-To: $signup_email" . "\r\n";
$headers .= "MIME-Version: 1.0\r\n";
$headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";
$message = '<html><body>';
$message .= '<table rules="all" style="border-color: #666;" cellpadding="10">';
$eml = $_REQUEST['email'];
$message .= "<tr ><td>
<h2 align='center'>EMAIL LIST SIGNUP REQUEST</h2>There is just one more step to be included on our
email distribution list.To make sure that no one else signed you up for this list, please click on this link or paste it
into your browser.
 <a
href='http://your_site.com/confirmed.php?
submit=1&confirmed=1&hidden=0&email=$eml'>http://your_site.com/confirmed.php?
submit=1&confirmed=1&hidden=0&email=$eml</a></td></tr>";
$message .= "</table>";
$message .= "</body></html>";


// Send
if (mail($to,$subject,$message, $headers))
{
 echo 'Mail sent!';
} else
{
 echo 'Error! Mail was not sent.';
};
  }

}

?>
```

And the active code in the body of the page:

```
 <table width="92%" border="0" align="center">
           <tr>
             <td valign="top"><form method="post" action="">
                 <input type="hidden" name="submit" value="1" />
                 <?php if (@$errorsAndAlerts): ?>
                 <div class="heading_font" align="left" style="color:#C00">
                   <?php echo $errorsAndAlerts; ?>

                 </div>
                 <?php endif ?>
                 <table align="left"  border="0" cellspacing="0" cellpadding="2">
                   <tr>
                     <td class=" text_font" valign="top"><b>First Name</b></td>
                     <td><input type="text" name="first_name" value="<?php echo
htmlspecialchars(@$_REQUEST['first_name']) ?>" size="30" /></td>
                   </tr>
                   <tr>
                     <td class="text_font" valign="top"><b>Last Name</b></td>
                     <td><input type="text" name="last_name" value="<?php echo
htmlspecialchars(@$_REQUEST['last_name']) ?>" size="30" /></td>
                   </tr>
                   <tr>
```

```
                    <td class="text_font" valign="top"><b>Email Address</b></td>
                    <td><input type="text" name="email" value="<?
php echo htmlspecialchars(@$_REQUEST['email']) ?>"
size="30" /></td>
                  </tr>
                  <tr>
                    <td class="text_font" valign="top"><b>Re-enter Your Email Address</b>
</td>
                    <td><input type="text" name="email2" value="<?
php echo htmlspecialchars(@$_REQUEST['email2']) ?>"
size="30" /></td>
                  </tr>
                  <tr>
                    <td valign="top">&nbsp;</td>
                    <td>&nbsp;</td>
                  </tr>
                  <tr>
                    <td colspan="2" class="text_font" style=" font-
weight: bold;" valign="top">Please check the "I'm
not a robot" box below before submitting.

                      <div class="g-recaptcha" data-theme="light" data-
sitekey="Your Google Recaptcha site
key"></div></td>
                  </tr>
                  <tr>
                    <td style="padding: 5px;">
<input type="submit" name="add" value="Click To Submit &gt;&gt;" />

                      </td>
                  </tr>
                </table>
              </form></td>
          </tr>
        </table>
```

confirmed.php

The code at the top of the page, above the head, after the records calls required for your site:

```
<?php
// submit form
if (@$_REQUEST['submit']) {

  // error checking
  $errorsAndAlerts = "";
  if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter the email address you used when you signed up.\n"; }

   // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

 // update user
   if (!$errorsAndAlerts) {

    $emailExists        = mysql_count('email_signup',         ['email' => $_REQUEST['email']]);
$emailConfirmed      = mysql_count('email_signup', ['email' => $_REQUEST['email'], 'confirmed' => 1]);
$userNum            = 0;
if ($emailExists && !$emailConfirmed) {
  $updateNum    = null;
  $updateWhere  = [ 'email' => $_REQUEST['email'] ];
  $colsToValues = [ 'hidden' => 0, 'confirmed' => '1', 'updatedDate=' => 'NOW()' ];
  mysql_update('email_signup', $updateNum, $updateWhere, $colsToValues);
  $userNum = mysqli()->insert_id;
$errorsAndAlerts = "Thanks, your email address has been succesfully added to our maillist";

}
    elseif($emailExists && $emailConfirmed){//email exists, but has already been confirmed
      $errorsAndAlerts.="That Email address has already been confirmed.If you'd like to sign up with another Email
address,<a  href='http://your_site.com/email_signup.php'>
<span class='heading_font' >CLICK ON THIS LINK</span></a>";
    }

     elseif (!$emailExist){ //email does not exist in the database
    $errorsAndAlerts.="Sorry, that email address doesn't exist in the database.Please enter the same Email address that you
used when you signed up, and click on submit.";
      }

}}

?>
```

And the active code in the body of the page:

```
 <form method="post" action="">
       <input type="hidden" name="submit" value="1" />

       <table width="90%" border="0" cellpadding="5" cellspacing="0">
         <tr>
           <td valign="top">&<?php if (@$errorsAndAlerts): ?>
       <div class="text_font" align="left" style="color: #C00; font-weight: bold;">
         <?php echo $errorsAndAlerts; ?>
       </div>
       <?php endif ?></td>
           <td>&nbsp;</td>
         </tr>
         <tr>
           <td align="left" class="text_font" valign="top">
<b>If you see an error above, it's probably because <input
type="text" name="email" value="<?php echo htmlspecialchars(@$_REQUEST['email']) ?
>" size="30" /> is not the Email
address that you used when you signed up.

             Please change it and click on submit.</b></td>
           <td>&nbsp;</td>
         </tr>
         <tr>
           <td valign="top">&nbsp;</td>
           <td>&nbsp;</td>
         </tr>
       </table>
       <input type="submit" name="add" value="Click to Submit &gt;&gt;" />
     </form>
     <p align="left" class="text_font">

     If you no longer want to receive information about us, <a style="text-
decoration:underline; color:#000"
href="http://www.yoursite.com/unsubscribe.php">
<span class="text_font">CLICK/TAP HERE</span></a> to unsubscribe.</p>
```

unsubscribe.php

The code at the top of the page, above the head, after the records calls required for your site:

```
<?php
  // load records from 'email_signup'
  list($email_signupRecords, $email_signupMetaData) = getRecords(array(
    'tableName'   => 'email_signup',
    'loadUploads' => true,
    'allowSearch' => false,
  ));
?>
<?
php $signup_email = 'the_email_address_you_want_to_use_for_return_and_reply' // the email to use for return and
reply?>
<?php
if (@$_REQUEST['submit']) {

function validateGoogleCaptcha(){

    $errorsAndAlerts = "";

    if (!@$_REQUEST['g-recaptcha-
response'])     { $errorsAndAlerts .= "Please check the anti-spam 'I am not a robot'
checkbox!\n";
    $showSignupForm = true; // don't change this value
    }
    else {
      // check recaptcha
      $postdata = array();
      $postdata['secret']   = 'your Google Recaptcha secret code';
      $postdata['response'] = @$_REQUEST['g-recaptcha-response'];
      $postdata['remoteip'] = $_SERVER['REMOTE_ADDR'];
      $url = "https://www.google.com/recaptcha/api/siteverify?". http_build_query($postdata, '', '&');
      list($json, $httpStatusCode, $headers, $request) = getPage($url, 5, '', true);
      $recaptchaResponse = json_decode($json, true);

      if (!$recaptchaResponse['success']) {
        if (is_array($recaptchaResponse['error-codes'])) {
          if (in_array('missing-input-secret', $recaptchaResponse['error-
codes']))     { $errorsAndAlerts .= "There's a
problem with recaptcha, please let us know! (no secret)\n"; }
          if (in_array('invalid-input-secret', $recaptchaResponse['error-
codes']))     { $errorsAndAlerts .= "There's a
problem with recaptcha, please let us know! (invald secret)\n"; }
          if (in_array('missing-input-response', $recaptchaResponse['error-
codes']))   { $errorsAndAlerts .= "Please
fill out the recaptcha box!\n";
          $showSignupForm = true; // do we need this line?
           }
          if (in_array('invalid-input-response', $recaptchaResponse['error-
          if (in_array('invalid-input-response', $recaptchaResponse['error-
codes']))   { $errorsAndAlerts .= "Please
fill out the recaptcha box again, your answer was incorrect!\n";
          $showSignupForm = true; // do we need this line?
          }
        }
        if (!@$errorsAndAlerts) { @$errorsAndAlerts .= "Invalid captcha response, please try again or contact us
directly and let us know."; }
        @trigger_error("Failed recaptcha on signup form", E_USER_NOTICE);
      }
    }
    return $errorsAndAlerts;
  }
}
?>
<?php $redirect = '0' // set a variable called $redirect to a value of 0 ?>
<?php if(@$_REQUEST['submit']):?>
<?php
    $errorsAndAlerts = "";
    $errorsAndAlerts .= validateGoogleCaptcha();



  if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter your email address\n"; }

// email checking
    if ($_REQUEST['email'] || $_REQUEST['email2']) {
      if (!@$_REQUEST['email'])                          { $errorsAndAlerts .= "Please enter your email address\n"; }
      elseif (!@$_REQUEST['email2'])                          { $errorsAndAlerts .= "Please re-
enter your email
address\n"; }
      elseif ($_REQUEST['email'] != $_REQUEST['email2']) { $errorsAndAlerts .= "Sorry, the e mail addresses you entered
don't match!\n"; }
    }


 // check for existing emails
   if (!$errorsAndAlerts) {

      $count = mysql_select_count_from('email_signup', "'".mysql_escape($_REQUEST['email'])."' IN (email)");
         if ($count < 1) { $redirect = '1'; // if no matching email address change the variable $redirect to a value of
1
         }
    }

  // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

  ?>
<?php
  // error checking
  if (!@$errorsAndAlerts && $redirect == '0') {


    // display sorry message and clear form
    $errorsAndAlerts = "We'll be sorry to see you go.To make sure that no one else is trying to remove your email
address from our list, you'll need to confirm your intent by clicking on the link in the email that you will receive
shortly.  If you don't see the email, check your spam folder.";

    // send email to applicant
    $to=$_REQUEST['email'];
$subject = 'Email List Removal Request';

$headers = "From: $signup_email" . "\r\n";
$headers .= "Reply-To: $signup_email" . "\r\n";
$headers .= "MIME-Version: 1.0\r\n";
$headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";
$message = '<html><body>';
$message .= '<table rules="all" style="border-color: #666;" cellpadding="10">';
$eml = $_REQUEST['email'];
$message .= "<tr ><td><div align='left'><img src='http://www.your_site.com/images/email-
masthead-400px.png'
style='border:hidden;'/></div>
<h2 align='center'>EMAIL LIST LIST REMOVAL REQUEST</h2>We're sorry to see you go.There's
just one more step to be removed from our email distribution list.To make sure that no one else is trying to remove you
from this list, please click on this link or paste it into your browser.
 <a
href=http:/your_site.com/unsubscribe_confirm.php?
submit=1&remove_me=1&email=$eml'>http://your_site.com/unsubscribe_confirm.php?
submit=1&remove=1&email=$eml</a></td></tr>";
$message .= "</table>";
$message .= "</body></html>";

// Send
if (mail($to,$subject,$message, $headers))
{
 echo 'Mail sent!';
} else
{
 echo 'Error! Mail was not sent.';
};
  }
?>
<?php if ($redirect == '1'):?>
<?php

header("Location:http://your_site.com/your_unsubscribe_contact_page.php");

exit;
?>
<?php endif ?>
<?php endif?>
```

And the active code in the body of the page:

```
<table width="100%"   border="0" align="center" cellpadding="10">
        <tr>
          <td valign="top">
<div align="center" class="heading_font">EMAIL LIST UNSUBSCRIBE
            </div>
            <div align="center" style="width:80%; text-align:left">
            <div align="center">
              <p class="heading_font">Please Use The Form Below
                To Unsubscribe From Our Email List </p>
            </div>
            <form method="post" action="">
              <input type="hidden" name="submit" value="1" />
              <?php if (@$errorsAndAlerts): ?>
              <div align="left" style="color: #C00; font-family: Verdana, Arial, Helvetica, sans-
serif; font-weight:
bold; font-size: 15px;">
                <?php echo $errorsAndAlerts; ?>

              </div>
              <?php endif ?>
              <table align="left"  border="0" cellspacing="0" cellpadding="2">
                <tr>
                  <td class="Medium-Text" valign="top"><b>Enter The Email Address
                    To Be Removed</b></td>
                  <td><input type="text" name="email" value="<?
php echo htmlspecialchars(@$_REQUEST['email']) ?>"
size="30" /></td>
                </tr>
                <tr>
                  <td class="Medium-Text" valign="top"><b>Re-enter The Email Address</b>
</td>
                  <td><input type="text" name="email2" value="<?
php echo htmlspecialchars(@$_REQUEST['email2']) ?>"
size="30" /></td>
                </tr>
                <tr>
                  <td colspan="2" class="text_font" style="color: #<?php echo
$site_colorsRecord['menu_background_color'] ?>; font-
weight: bold;" valign="top">Please check the "I'm not a robot" box
below before submitting.

                    <div class="g-recaptcha" data-theme="light" data-
sitekey="your_Google Recaptcha site
key"></div></td>
                </tr>
                <tr>
                  <td colspan="2" valign="top">
                    <input type="submit" name="submit" value="Click To Submit &gt;&gt;" /></td>
                </tr>
              </table>
            </form></td>
        </tr>
      </table>
```

unsubscribe_confirm.php

The code at the top of the page, above the head, after the records calls required for your site:

```
<?php
// submit form
if (@$_REQUEST['submit']) {


  // error checking
  $errorsAndAlerts = "";
  if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter the email address you used when you signed up.\n"; }

   // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

 // update user
   if (!$errorsAndAlerts) {

    $emailExists        = mysql_count('email_signup',         ['email' => $_REQUEST['email']]); //check to ensure that
email exists in the table
$emailRemoved     = mysql_count('email_signup', ['email' => $_REQUEST['email'], 'remove' => 1]); // check to ensure that
email exists and has not been removed yet
$userNum           = 0;
if ($emailExists && !$emailRemoved) {
  $updateNum    = null;
  $updateWhere  = [ 'email' => $_REQUEST['email'] ];
  $colsToValues = [ 'hidden' => 1, 'remove' => '1', 'updatedDate=' => 'NOW()' ];
  mysql_update('email_signup', $updateNum, $updateWhere, $colsToValues);
  $userNum = mysqli()->insert_id;
$errorsAndAlerts = "Thanks, your email address has been successfully removed from our maillist.To sign up again, <a
style='text-
decoration:underline; color:#C00;' href='http://www.your_site.com/email_signup.php'>
<span
class='text_font'><font color='#C00'>CLICK HERE</font></span>
</a> for our email sign up page.";

}
    elseif($emailExists && $emailRemoved){//email exists, but has already been removed
      $errorsAndAlerts.="That Email address has already been removed.To remove another address, <a
style='text-
decoration:underline; color:#C00;' href='http://www.your_site.com/unsubscribe.php'><span
class='text_font'><font color='#C00'>CLICK HERE</font></span>
</a> to return to our unsubscribe page.";
    }

     elseif (!$emailExist){ //email does not exist in the database
    $errorsAndAlerts.="Sorry, that email address doesn't exist in the database.>To be removed from our list, <a
style='text-
decoration:underline; color:#C00;' href='http://www.your_site.com/unsubscribe.php'><span
class='text_font'><font color='#C00'>CLICK HERE</font></span>
</a> to return to our unsubscribe page and enter the email
address that you used when you signed up.";
      }

}}

?>
```

And the active code in the body of the page:

```
<div align="center" class="heading_font">
        <h2>Email List Unsubscribe Confirmation</h2>
      </div>
      <table width="900px"   border="0" align="center" cellpadding="10">
        <tr>
          <td class="heading_font" ><?php if (@$errorsAndAlerts): ?>
            <div align="left" style="color: #C00; font-family: Verdana, Arial, Helvetica, sans-
serif; font-weight: bold;
font-size: 15px;">
              <?php echo $errorsAndAlerts; ?>

            </div>
            <?php endif ?>

            </td>
        </tr>
      </table>
```

UNSUBSCRIBE FROM YOUR SIGN-UP LIST AUTOMATICALLY - Mar 7th, 2013

This double opt out variation on the sign-up theme (See the recipe on DOUBLE OPT IN MAIL LIST IN CMSB) automatically removes a subscriber from your email list. It requires that you create a “remove” check box field in your sign-up editor (mine is called newsletter_signup) and that you use an if statement to check for the value of that field when outputting your list.

I.E.:

```
<?php if  ($record['hidden'] == 0 && $record['confirmed'] == 1 && $record['remove'] == 0): ?
>
                   <?php echo $record['email'] ?>;&nbsp;
                  <?php endif ?>
```

There are 2 viewers required as well. An “unsubscribe” viewer and a “confirmation” viewer

When a subscriber wants to unsubscribe, they enter their email address in an “unsubscribe” form and to help prevent spam, they’re required to fill in a “captcha” field. (See the recipe on IMPLEMENTING CAPTCHA ON THE WEBSITE MEMBERSHIP PLUGIN 1.04+ SIGNUP FORM)

The form sends an email to that email address, with a confirmation link that the subscriber has to click (or tap) on to complete the unsubscribe process.

The confirmation page uses error correction to see if the email address exists in the database and if it does, a “remove” check box field is “checked” and a confirmation message is presented to the visitor along with a link to the subscription form. If the box is already checked, an message to that effect is returned along with a link back to the unsubscribe form.

Here’s the active code for the unsubscribe viewer:
TOP OF THE PAGE
Note: change the path to your viewer and any required load records calls at the head of your page, and the domain name in the // send email to applicant section.

```
<?php
include_once('captcha/captchac_lib.php');
session_write_close ();

if (@$_REQUEST['submit']) {
  $errorsAndAlerts = "";
    if (!@$_REQUEST['Turing'])    {$errorsAndAlerts .= "To help prevent spam, you must enter the characters in the image
into the empty box below before submitting.\n";}
    else
    {
        $Turing_code = $_REQUEST["Turing"];
       if ( CheckCaptcha($Turing_code) !=1 )  {  $errorsAndAlerts .= "The characters you entered do not match those
displayed. Please try again.\n";}

    }
}
?>
<?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."); }

// any required load records calls go here

 ?>

<?php
// submit form
if (@$_REQUEST['submit']) {

  // error checking

    if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter your email address\n"; }

// email checking
    if ($_REQUEST['email'] || $_REQUEST['email2']) {
      if (!@$_REQUEST['email'])                            { $errorsAndAlerts .= "Please enter your email address\n"; }
      elseif (!@$_REQUEST['email2'])                            { $errorsAndAlerts .= "Please re-
enter your email
address\n"; }
      elseif ($_REQUEST['email'] != $_REQUEST['email2']) { $errorsAndAlerts .= "Sorry, the e mail addresses you entered
don't match!\n"; }
    }

 // check fo rexisting emails
    if (!$errorsAndAlerts) {

      $count = mysql_select_count_from('newsletter_signup', "'".mysql_escape($_REQUEST['email'])."' IN (email)");
      if ($count < 1) { $errorsAndAlerts .= "We can't find that email address in our list. Please enter the email
address that you signed up with and try again.\n"; }
    }

  // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

  if (!@$errorsAndAlerts) {

    // display thanks message and clear form
    $errorsAndAlerts = "We'll be sorry to see you go.To make sure that no one else is trying to remove your email
address from our list, you'll need to confirm your intent by clicking on the link in the email that you will receive
shortly.  If you don't see the email, check your spam folder.";

    // send email to applicant
    $sendto=$_REQUEST['email'];
$subject='We\'re sorry to see you go.';
$header = "From: newsletter@".$_SERVER["SERVER_NAME"]."\n";
$header .= "Content-Type: text/html; charset=iso-8859-1\n";
$header .= 'Content-Type: text/html; charset=iso-8859-1\r\n';
$message = ' Thanks.We\'re sorry to see you go.There\'s just one more step to be removed from our newsletter
distribution list.To make sure that no one else is trying to remove you from this list, please click on this link or
paste it into your browser.
 http://www.your_domain.com/unsubscribe_confirm.php?
submit=1&remove=1&email='.$_REQUEST['email'];";

$_REQUEST = array();
// Send
if (mail($sendto,$subject,$message,$header, "-
fwebmaster@".$_SERVER["SERVER_NAME"]))
{
 echo 'Mail sent!';
} else
{
 echo 'Error! Mail was not sent.';
};
  }

}


?>
<!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">
```

BODY OF THE PAGE

```


<table    align="center" cellpadding="0">

        <tr>
    <td >
      <div align="center">
You'll receive an email from us to confirm your intention to unsubscribe.
To make sure that no one else is trying to remove you from our list, you'll need to click on the link in that email.

</div>
<table width="72%"  align="center">

  <tr>
    <td valign="top">

     <form method="post" action="">
<input type="hidden" name="submit" value="1" />



<?php if (@$errorsAndAlerts): ?>
  <div align="left" style="color: #FFFF00; font-family: Verdana, Arial, Helvetica, sans-
serif; font-weight: bold;
font-size: 15px;">
    <?php echo $errorsAndAlerts; ?>
  </div>
<?php endif ?>


<table align="left"   cellspacing="0" cellpadding="2">
 <tr>
  <td valign="top"><b>Enter the email address that you'd like to removefrom our list</b>
</td>
  <td><input type="text" name="email" value="<?
php echo htmlspecialchars(@$_REQUEST['email']) ?>" size="30" /></td>
</tr>
<tr>
  <td valign="top"><b>Re-enter the email address</b></td>
  <td><input type="text" name="email2" value="<?
php echo htmlspecialchars(@$_REQUEST['email2']) ?>" size="30" /></td>
</tr>
<tr>
  <td valign="top">&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
                    <td  valign="top"><b>CAPTCHA Image Verification</b>

                    To help prevent spam, type the characters in the image into the empty box below  before submitting.

                   <a href="/captcha/whatisturing.html"
onclick="window.open('/captcha/whatisturing.html','_blank','width=300, height=300, left=' + (screen.width-
450) + ',
top=100');return false;">WHAT'S CAPTCHA?</a></td>
                    <td>

                    <table cellpadding=5 cellspacing=0 >


 <tr>
```

```
<tr>
<td style="padding: 5px;" width="300" height="100">
<img src="/captcha/captchac_code.php" id="captcha">
<a class="text_font" href="#" onclick=" document.getElementById('captcha').src = document.getElementById('captcha').src
+ '?' + (new Date()).getMilliseconds()">
</a></td>
</tr>
<tr>
  <td style="padding: 5px;">
<input type="text" name="Turing" value="" maxlength="100" size="20" height="50" />

    <b>Enter Characters Here</b >
       </td>
</tr>
<tr>
  <td style="padding: 5px;">
<input type="submit" name="add" value="Click To Submit &gt;&gt;" /><a class="text_font"
href="#" onclick=" document.getElementById('captcha').src = document.getElementById('captcha').src + '?' + (new
Date()).getMilliseconds()">HAVING TROUBLE READING
    THE CHARACTERS DISPLAYED?
    CLICK HERE TO CHANGE THEM.</a></td>
</tr>
          </table>
</td></tr>
</table></form></td></tr></table></td>
  </tr>
</table>
```

And here’s the code for the confirmation viewer (called unsubscribe_confirm.php)
NOTE: This approach assumes that all email addresses will always be drawn from your database. If your implementation will be drawing email addresses form other sources in addition to the database, see the recipe called HANDLING UNSUBSCRIBE REQUESTS FROM EMAILS NOT IN YOUR DATABASE

TOP OF THE PAGE

```
<?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."); }

// any required load records calls go here

 ?>
<?php
// submit form
if (@$_REQUEST['submit']) {

  // error checking
  $errorsAndAlerts = "";
  if (!@$_REQUEST['email'])  { $errorsAndAlerts .= "Please enter the email address you used when you signed up.\n"; }

   // turn off strict mysql error checking for: STRICT_ALL_TABLES
  mysqlStrictMode(false); // disable Mysql strict errors for when a field isn't defined below (can be caused when fields
are added later)

 // update user
   if (!$errorsAndAlerts) {

      $where = "email='".mysql_escape($_REQUEST['email'])."'";

      if(mysql_select_count_from('newsletter_signup',$where))
{ //check to ensure that email exists in the table
    $where.=" AND remove='1'";
    if(!mysql_select_count_from('newsletter_signup',$where))
{ //email exists and has not been removed yet
        $query = "UPDATE `{$TABLE_PREFIX}newsletter_signup` SET
              remove           = '1',
              updatedDate      = NOW()
              WHERE email = '".mysql_escape( $_REQUEST['email'] )."'";
        mysql_query($query) or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
        $userNum = mysql_insert_id();

        // on success
        $errorsAndAlerts = "Thanks, your email address has been successfully removed from our maillist.To sign up again, <a
href='http://www.your_domain.com/email_signup_web.php'>
<span style='color: #FFFF00; font-family: Verdana, Arial,
Helvetica, sans-serif; font-weight: bold; font-size: 15px;'>CLICK HERE</span>
</a> for our Newsletter subscription
page.";
    }
    else{//email exists, but has already been removed
      $errorsAndAlerts.="That Email address has already been removed.To remove another address, <a
href='http://www.your_domain.com/unsubscribe.php'><span style='color: #FFFF00; font-
family: Verdana, Arial, Helvetica,
sans-serif; font-weight: bold; font-size: 15px;'>CLICK HERE</span>
</a> to return to our unsubscribe page.";
    }
      }
      else{ //email does not exist in the database
    $errorsAndAlerts.="Sorry, that email address doesn't exist in the database.To be removed from out list, <a
href='http://www.your_domain.com/unsubscribe.php'><span style='color: #FFFF00; font-
family: Verdana, Arial, Helvetica,
sans-serif; font-weight: bold; font-size: 15px;'>CLICK HERE</span>
</a> to return to our unsubscribe page and enter the
email address that you used when you signed up.";
      }

    }
  }

?>
<!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:

```
<table width="40%"  align="center">

  <tr>
    <td valign="top">

<?php if (@$errorsAndAlerts): ?>
  <div align="left" style="color: #FFFF00; font-family: Verdana, Arial, Helvetica, sans-
serif; font-weight: bold;
font-size: 15px;">
    <?php echo $errorsAndAlerts; ?>
  </div>
<?php endif ?>

</td>
  </tr>
</table>
```

HANDLING UNSUBSCRIBE REQUESTS FROM EMAILS NOT IN YOUR DATABASE - Aug 25th, 2014

There are situations where the subscriber that wants to unsubscribe may not have come from the email list or media distribution list databases. If this fits your client’s situation, you might want to redirect subscribers with emails not found to a contact form instead of just a "re-enter the email address you used when you signed up" message.

To accomplish this, you’ll have to change code in your unsubscribe form viewer and then create a contact form viewer.

THE UNSUBSCRIBE FORM VIEWER

In the code of your unsubscribe form viewer:

```
1) insert this code immediately after your load records code block

<?php $redirect = '0' // set a variable called $redirect to a value of 0 ?>
```

2) In the // check for existing emails section change: (change ‘newsletter_signup’ to the table name where you store your list)

```
 if (!$errorsAndAlerts) {
    $count = mysql_select_count_from('newsletter_signup', "'".mysql_escape($_REQUEST['email'])."' IN (email)");
      if ($count < 1) { $errorsAndAlerts .= "We can't find that email address in our list. Please enter the email
address that you signed up with and try again. <br /> <br /> \n"; }
    }
```

To:

```
if (!$errorsAndAlerts) {
   $count = mysql_select_count_from('newsletter_signup', "'".mysql_escape($_REQUEST['email'])."' IN (email)");
      if ($count < 1) { $redirect = '1'; // if no matching email address change the variable $redirect to a value of 1
}
    }
```

3) To the // error checking if statement change:

```
if (!@$errorsAndAlerts) {
```

to this:

```
if (!@$errorsAndAlerts && $redirect == '0') // Sends email only if there are no errors and there's a matching email
address in the database
{
```

4) CAVEAT: Add this redirect code Before the DOCTYPE declaration to redirect the visitor to your contact form viewer if there's no matching email in the database. It must be before any HTML tag or you'll get errors.

```
<?php if ($redirect == '1'):?>
<?php

header("Location:your_contact_form_viewer.php");

exit;
?>
<?php endif ?>
```

THE CONTACT FORM VIEWER.

The form can be created in any email form generator. I'm using the free Forms Generator that was put together as a SourceForge project (**http://phpfmg.sourceforge.net/**). There's an implementation recipe called ANOTHER (FREE) PHP FORMS GENERATOR THAT ALLOWS ATTACHMENTS

The active code in my viewer puts the form in an iframe, but you can use any form or method that works for your application:

```
 <table border="0">
<tr>
<td> <div align="left" >
<span class="heading_font">You'll need to submit this contact form to let me knowthat you want to unsubscribe from my
email list.</span>
<br /> <br />
<span class=" text_font">I'm sorry that you've decided not to receive my updates in the future.I hope that you'll let me
know why you're unsubscribing. <br /></span>
<br />
<iframe frameborder="none" src="http://www.my_site.com/contact/form.php" style="width:420px;height:420px;border:none;"
Allowtransparency="true"></iframe>
 </td>
</tr>
</table>
```

OFFERING PRIVATE EMAIL LIST CAPTURE TO YOUR MEMBERS - Jun 22nd, 2011

Want to allow your members to create independent, private, double opt in email lists.

It's easier than you think.

Building on two recipes, LIMITING ACCESS TO RECORDS BY AUTHOR and DOUBLE OPT IN MAIL LIST IN CMSB, here's how:

CONCEPT:
In my situation, the visitor was looking at the detail page of a record that was created by specific member. That page also contained my email signup form.

The process I used was to query the createdByUserNum field of that detail page record, and use that value to populate the createdByUserNum field of my "email_signup" table.

To display only the email addresses that belong to that member on an email list page, I restricted the records displayed to only the email address records that were tagged with the member's user number.

CODE CHANGES
First, create an email_signup table following he instructions in the DOUBLE OPT IN MAIL LIST IN CMSB recipe.

Then add the active error checking and execution code, and the email signup form to the detail page referenced above.

On that detail page, above the error correction and execution code add the following to define the variable: $owner :

```
<?php $owner = $your_detail_page_tableRecord['createdByUserNum'] ?>
```

And in the //add record section, change

And in the //add record section, change

```
createdByUserNum = '0',
```

to:

```
createdByUserNum = '$owner',
```

Now, each email_signup table record will be tagged with the user number of the member who created the record displayed on the detail page.

The last part is to limit the email addresses displayed to only those email_signup records attributed to that member.

Here are the simple changes to the email list display page described in the DOUBLE OPT IN MAIL LIST IN CMSB recipe.

In both foreach loops, add:

```
<?php if ($record['createdByUserNum']== @$CURRENT_USER['num']): ?>
```

Below the:

```
<?php if  ($record['hidden'] == 0 && $record['confirmed'] == 1): ?>
```

And add an <?php endif ?> after each existing <?php endif ?>

That should do it.

MERGE/PURGE NEWSLETTER MAIL LISTS WITH CMSB AND THE WEBSITE MEMBERSHIP PLUGIN - Aug 18th, 2011

In the recipe above, an organization has a newsletter signup form on their site that automatically populates a CMSB multi record database called “Newsletter Signup”. (The Newsletter Signup database has only 4 fields: Hidden, First Name, Last Name, and Email)

When a subscriber joins the organization they use the membership module to automatically create a record in the “accounts” database. After the new member has been approved (by checking an “approved” checkbox in the accounts database) , they want to be able to remove their e-mail address from the Newsletter Signup e-mail list. They also want to be able to create reports that show who joined after getting the newsletter, the date they subscribed, and the date they joined.

Jason Sauchuk from Interactive tools was up to the challenge.

He came up with the original code to compare the e-mail addresses in both the” Accounts” and the “Newsletter Signup” database, and when there was a match, to hide the record in the Newsletter Signup database.

Here’s that code:

At the top of the page:

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

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

  list($newsletter_signupRecords, $newsletter_signupMetaData) = getRecords(array(
    'tableName'   => 'newsletter_signup',

  ));

?>
```

And in the body:

```
<?php foreach($newsletter_signupRecords as $record){
    $where = "email = '".mysql_escape($record['email'])."' AND approved=1";

    if(mysql_select_count_from('accounts',$where)){

        $query = "UPDATE `{$TABLE_PREFIX}newsletter_signup` SET hidden = 1 WHERE num ='".intval($record['num']) ;

       mysql_query($query) or die("MySQL Error:".mysql_error()."<br />\n");
    }

  }
  ?>
```

When I added reporting code into the body, my first approach was to add this where statement to the top of the page:

```
list($newsletter_signupRecords, $newsletter_signupMetaData) = getRecords(array(
    'tableName'   => 'newsletter_signup',
     'where'    => " hidden = '0' ",
    'orWhere'    => " hidden = '1' ",
  ));
```

And this code below the existing body code:

```
MEMBERS WHO HAVE NOT YET JOINED  <br />
<?php foreach ($newsletter_signupRecords as $record): ?><?
PHP if ($record['hidden'] == 0): ?>
<?php echo $record['last_name'] ?>, <?php echo $record['first_name'] ?>
<br />
<?php echo $record['email'] ?>
<br />
Date first signed up: <?php echo date("D jS, M  Y ", strtotime($record['createdDate'])) ?>
<hr>   <?php endif ?>
<?php endforeach ?>

<br /><br />MEMBERS WHO JOINED AFTER GETTING THE NEWSLETTER<br />
 <?php foreach ($newsletter_signupRecords as $record): ?><?
PHP if ($record['hidden'] == 1): ?>
<?php echo $record['last_name'] ?>, <?php echo $record['first_name'] ?>
<br />
<?php echo $record['email'] ?>
<br />
Date first signed up: <?php echo date("D jS, M  Y ", strtotime($record['createdDate'])) ?>
<br />
Date joined: <?php echo date("D jS, M  Y ", strtotime($record['updatedDate'])) ?>
<hr>   <?php endif ?>
<?php endforeach ?>
```

The reporting worked perfectly. The problem was that until the page was opened, and then refreshed, it did not reflect the current changes to the information.

Again, Jason came to the rescue. He suggested that the easiest way to insure that the information was correct, was to compile the report into 2 different variables while you're processing your records. Then all you have to do is output the results at the end. This way you won't have to do a second database call to update the report.

Here’s the new code that he suggested for the body:

```
<?php $notConverted="";
      $converted="";
?>

<?php foreach($newsletter_signupRecords as $record){

$where = "email = '".mysql_escape($record['email'])."' AND approved=1";

  if(mysql_select_count_from('accounts',$where)){

      $converted.=$record['last_name'].",".$record['first_name']."<br />".$record['email']."
<br />";
      $query = "UPDATE `{$TABLE_PREFIX}newsletter_signup` SET hidden = 1 WHERE num ='".intval($record['num']) ;

     mysql_query($query) or die("MySQL Error:".mysql_error()."<br />\n");
  }
  else{
    $notConverted.=$record['last_name'].",".$record['first_name']."<br />".$record['email']."
<br />";
```

```
  }
}
?>

Not Converted to Membership <br />
<?php echo $notConverted; ?>

Converted to Membership <br />
<?php echo $converted; ?>
```

IMPLEMENTING CAPTCHA ON THE WEBSITE MEMBERSHIP PLUGIN 1.04+ SIGNUP FORM - Aug 18th, 2011

NEW FOR MEMBERSHIP 1.04+

I wanted to use CAPTCHA validation on my membership signup form and although RE-CAPTCHA was pretty easy to implement, it
was a pretty intrusive addition to many web pages.

I discovered another alternative in a program called Captcha Creator. **http://www.captchacreator.com/** They're based in Romania and their support is a bit spottier then I'd like. (I've grown accustomed to the exceptional support offered by Interactive Tools) But now that I've been able to implement their program, I can format the Captcha display to fit my page design and it works like a charm. Again, their program is not free, and an unlimited use license costs $50.

Implementing the “out of the box” version of Captcha Creator on my membership signup form was throwing a bunch of “session already started” errors. The programmers at Captcha Creator revised the way the code was implemented on my page and also made some changes in the captchac_lib.php file and a few others.

*** After you've uploaded the “captcha” directory to your server, download the revised files from **http://www.thecmsbcookbook.com/downloads/captcha_creator.zip** and replace the ones in your “captcha” directory with those.

On my existing membership signup form I changed CAPTCHA CREATOR's Installation Step 2 code to better suit my design.

Instead of:

```
<table cellpadding=5 cellspacing=0 bgcolor="#E4F8E4">
<tr bgcolor="#AAD6AA">
<td colspan="2"><font color="#FFFFFF" face="Verdana" size="2">
<b>Image Verification</b></font></td>
</tr>
<tr>
<td style="padding: 2px;" width="10">
<img src="/captcha/captchac_code.php" id="captcha"></td>
<td valign="top">
<font color="#000000">Please enter the text from the image</font> &nbsp; <br /> <input type="text"
name="Turing" value="" maxlength="100" size="10">
[ <a href="#" onclick=" document.getElementById('captcha').src = document.getElementById('captcha').src + '?' + (new
Date()).getMilliseconds()">Refresh Image</a> ] [ <a href="/captcha/whatisturing.html"
onClick="window.open('/captcha/whatisturing.html','_blank','width=400, height=300, left=' + (screen.width-
450) + ',
top=100');return false;">What's This?</a> ]
</td>
</tr>
</table>
```

I used:

```
 <table>
<tr>
<td  valign="top"><span class="body-text-bold">CAPTCHA Image Verification</span>
<br /><br />
    <span class="body-
text">To help reduce spam, please type the characters in the image into the blank box before you
click on submit.<br />
    </span> <br />
    <a href="/captcha/whatisturing.html" onclick="window.open('/captcha/whatisturing.html','_blank','width=300,
height=300, left=' + (screen.width-450) + ', top=100');return false;">What is CAPTCHA?
</a></td>
<td>
<table cellpadding=5 cellspacing=0 >
<tr>
<td style="padding: 5px;" width="300" height="100">
<img src="/captcha/captchac_code.php" id="captcha"><br />
<a href="#" onclick=" document.getElementById('captcha').src = document.getElementById('captcha').src + '?' + (new
Date()).getMilliseconds()">Click to change the displayed characters</a><br /></td>
</tr>
<tr>
  <td style="padding: 5px;">
<input type="text" name="Turing" value="" maxlength="100" size="20" />
```

```
<input type="text" name="Turing" value="" maxlength="100" size="10" />
    <br />
    <br />    </td>
</tr>
</table></td></tr>
</table>
```

To make this all work in Membership 1.04+, at the top of the viewer I called the CAPTCHA program and checked for errors with:

```
<?php
include_once('captcha/captchac_lib.php');
session_write_close ();

if (@$_REQUEST['submit']) {
  $errorsAndAlerts = "";
    if (!@$_REQUEST['Turing'])    {$errorsAndAlerts .= "To help prevent spam, you must enter characters in the image into
the empty box below before submitting.<br /> <br /> \n";}
    else
    {
         $Turing_code = $_REQUEST["Turing"];
        if ( CheckCaptcha($Turing_code) !=1 )  {  $errorsAndAlerts .= "The characters you entered do not match those
displayed. Please try again.<br /> <br /> \n";}

    }
}
?>
```

And in the error checking code below I removed the clear errorsAndAlerts code line:

```
  $errorsAndAlerts = "";

So that it only appeared once at the top of the viewer code. (Otherwise it will clear the captcha error messages
__
```

Don't forget to use "save" or "submit" consistently throughout your viewer code. I think you can use either one but I've settled on submit.

__

*** LEGACY FOR MEMBERSHIP 1.03 AND EARLIER

In the error checking section of my page after:

```
 //error checking
$errorsAndAlerts = "";

if (!@$_REQUEST['username'])                { $errorsAndAlerts .= "You must choose a username!
<br />\n"; }
```

I added this code:

```
if (!@$_REQUEST['Turing'])    {$errorsAndAlerts .= "You must enter the CAPTCHA code!
<br />\n";}
else
{
     $Turing_code = $_REQUEST["Turing"];
    if ( CheckCaptcha($Turing_code) !=1 )  {  $errorsAndAlerts .= "The CAPTCHA code you entered does not match. Please try
again.<br />\n";}
}
```

For CAPTCHA CREATOR's Installation Step 3 I replaced this:

```
<?php
    include('captcha/captchac_lib.php');
    $Turing_code = $_REQUEST["Turing"];
    if ( CheckCaptcha($Turing_code) !=1 )
    {
```

```
    {
        echo "<b>
<font color=red>The Captcha Code you entered is invalid. Please press the Back button of your browser
and try again</font></b>";
    return 1;
    }
?>
```

With this:

```
<?php
include_once('captcha/captchac_lib.php');
session_write_close ();
?>
```

So the complete code at the top of my viewer, above the error checking section looks like this:

```
<?php
include_once('captcha/captchac_lib.php');
session_write_close ();
?>
<?php
require_once "cmsAdmin/lib/viewer_functions.php";
 ?>
 <?
php if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you
can access this page."); } ?>
```

(FOR V2.06 AND ABOVE SEE BELOW)
_

NOTE: If you've updated to V2.06 you've probably discovered that the error checking at the top of the lib/init.php file looks different, and pages using Captcha Creator are throwing errors again.

Well, thanks to CMSB User Djulia, the fix is in.

He discovered that when the function session_write_close (); is correctly written into the getRecords code at the top pf your page, the errors disappear, without the need to modify the lib/init.php file.

Calling the Captcha program first, then closing the session, then calling the CMSB viewer_function is the implementation that worked for me:

```
<?php
include_once('captcha/captchac_lib.php');
session_write_close ();
?>
<?php

require_once "cmsAdmin/lib/viewer_functions.php";


 list($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName'   => 'common_information',
    'where'       => whereRecordNumberInUrl(1),
    'limit'       => '1',
  ));
  $common_informationRecord = @$common_informationRecords[0]; // get first record

 ?>
<?php include_once "/path_to_your/spambot-email-protector.php" ?>
```

END LEGACY SECTION

IMPLEMENTING GOOGLE'S "I'M NOT A ROBOT", NO CAPTCHA, RECAPTCHA V3 IN A FORM - Aug 6th, 2022

With a lot of help from Greg Thomas and Dave Edis, here's how to implement the Google's new, "I'm Not A Robot", No CAPTCHA, reCAPTCHA in a CMSB Form.

You can download 2 sample viewers from **http://thecmsbcookbook.com/downloads/google-robot-captcha.zip**

Both incorporate the code required to implement reCAPTCHA.

One contains a very simple form, and demonstrates a bare bones implementation.
The other is the Sample User Signup Form viewer that's generated by the Website Membership Plugin. This viewer

The other is the Sample User Signup Form viewer that's generated by the Website Membership Plugin. This viewer demonstrates a more complex implementation with error checking.

The 5 step implementation for both viewers is the same.

1) Log in to **https://www.google.com/recaptcha/admin#list**
2) Register your web site. (if you want to get alerts from Google if they detect problems with your site, such as misconfiguration errors or an increase in suspicious traffic, leave the “get alerts” checkbox checked.)
3) A Site Key and Secret Key will be generated, copy them to a safe (secret) place.
4) Enter the 2 Keys as noted in the viewer code.
5) Enter your “Server Path” in the load viewer library code at the top of the viewer code.

You should be good to go.

Right now there are only a few “Themes” available to change the look of the check box. They are data-theme="dark"
and data-theme="light", and are changed in the <div> where you entered your site key.

You can learn more about Google’s No CAPTCHA reCAPTCHA at: **https://developers.google.com/recaptcha/**

IMPLEMENTING ASYNC TO DELAY THE LOADING REQUIREMENT OF THE GOOGLE RECAPTCHA V3 PLUGIN - Aug 6th, 2022

In August 2022, user JeffC was having a bit of an issue implementing async with the Google Recaptcha V3 plugin, which was slowing down the loading of his web pages when not deferred.

According to Daniel Louwe, Technical Lead at Interactive Tools,
It appears that the script provided in Google's docs is slightly inaccurate - at least for our usage here - so I've made a small change and it appears to be working with async. For anyone else reading, here is the final updated script:

```
  <script async src="https://www.google.com/recaptcha/api.js?render=<?php echo
urlencode(recaptcha_settings('site_key')); ?>"></script>

  <script>

    if(typeof grecaptcha === 'undefined') {
      grecaptcha = {};
    }
    grecaptcha.ready = function(cb){
      if(typeof grecaptcha.execute === 'undefined') {
        // window.__grecaptcha_cfg is a global variable that stores reCAPTCHA's
        // configuration. By default, any functions listed in its 'fns' property
        // are automatically executed when reCAPTCHA loads.
        const c = '___grecaptcha_cfg';
        window[c] = window[c] || {};
        (window[c]['fns'] = window[c]['fns']||[]).push(cb);
      } else {
        cb();
      }
    }

    grecaptcha.ready(function(){

        grecaptcha.execute('<?php echo urlencode(recaptcha_settings('site_key')); ?
>', {action: <?php echo
json_encode(recaptcha_settings('page')); ?>}).then(function(token) {

          document.getElementById("g-recaptcha-response").value = token;

        });

    });

  </script>
```

DEDUCTING A "REGISTRATION FEE" FROM THE FIRST PAYMENT ONLY - Feb 18th, 2011

My client wants to charge a registration fee to create a database record for new potential students. Then she wants to refund the registration fee (only once) if a student signs up and pays for a class.

Here’s the logic and the code required:

1) When the parent of a potential student wants to register, they first pay a small registration fee. (The registration fee value is pulled from a field in a single record editor called “common_information”) On submission of the application, a checkbox in the accounts database called “ registration fee refund” is automatically set to ‘1"
with this code which is added to the series of mysql querys in the signup form, under “mysql_query("INSERT INTO `{$TABLE_PREFIX}accounts` SET”:

```
registration_fee_refund     = '1',
```

2) When a parent goes to the pay tuition page, if the “registration_fee_refund” checkbox =1, the tuition fee displayed reflects the refund using this code:

First, a note telling the parent that their registration fee will be deducted from their tuition price.

```
<?PHP if ($CURRENT_USER['registration_fee_refund'] == 1): ?><br />
       *** Your ONE TIME registration fee refund is reflected in the tuition prices below. ***
<br /> <?PHP endif ?>
```

Then deduct the registration fee from the normal tuition fee with this code:

Step A) define a variable for the deposit amount using a 2 decimal place number format:

```
 <?
php $deposit_amount =  number_format($common_informationRecord['deposit_amount'],2); ?
>
```

Step B) If applicable, subtract the deposit amount from the normal tuition amount (pulled from a field in a multi record editor called “tuition_fees”.

```
<?php foreach ($tuition_feesRecords as $record): ?>
<?PHP if ($CURRENT_USER['registration_fee_refund'] == 1): ?><?
php echo number_format(($record['normal_tuition_amount'] -
$deposit_amount),2) ?><?php else:?><?
php echo number_format($record['normal_tuition_amount'] ,2) ?> <?PHP endif ?>
<?php endforeach ?>
```

3) After payment, the parent is automatically redirected to a “thank you” page. When the “thank you” page loads, if value of the refund registration fee field value is “1", the field is automatically reset to “0"

```
 <?php if ($CURRENT_USER ['registration_fee_refund'] == 1): ?>

     <?php mysqlStrictMode(false);
      $query = "UPDATE `{$TABLE_PREFIX}accounts` SET
       registration_fee_refund  = 0
               WHERE num = '".mysql_escape( $CURRENT_USER['num'] )."'";
      mysql_query($query) or die("MySQL Error:
<br />\n". htmlspecialchars(mysql_error()) . "\n");
      $userNum = mysql_insert_id();

   ?>

<?php endif ?>
```

SHOWING MEMBER VIDEOS ON A MEMBER’S PAGE - Apr 23rd, 2011

A client of mine who runs a dance school wanted to be able to create personalized videos for her students. She wanted each student who paid to have a video created to be able to see only their video.

Implementing this was made easier, because I was already using the “website membership” plugin and all students were also “members”.

NOTE:
I had created 2 new text fields in the “accounts” table, “last_name’ and “first_name” when I set up the CMSB installation, so that I could address each student by their name. I’ll use these fields for this recipe also.

Here’s how:

1) First I added a “videos_for_sale” check box field to the “accounts” user account section.

2) Next I created a multi-record section called Videos For Sale”. Under the Viewer URL tab, set the detail page URL to /videosdetail.php

Delete the “content” field and add 4 additional fields to the editor for a total of 5 input fields ("title" is the 5th)

A) An “upload” field called “video_uploads”, restricted to one mp4 or mov upload of unlimited size, with no thumbnails.
B) An “upload” field called “background_image” for an optional placeholder image while the video is loading.
C) A “text box” field called “description”
D) A pull down “list” field

The only field that’s a bit special is the list field. Instead of listing all the student’s names, I wanted to restrict the pull down to show only the last name of those students (members) that had a video created for them (a check mark in the “video_for_sale” check box in their account record).

Set the List Options pull down to “Get options from MySQL query”
replace the sample code in the box below to this:

```
SELECT num, last_name
  FROM '<?php echo $TABLE_PREFIX ?>accounts'
 WHERE video_for_sale='1'
```

This will populate the pull down list with the last_names from only those records with a check mark (1) in the “videos_for_sale” field in the “accounts” table.

After saving the new editor, check a few check boxes in the "accounts" records and create a few test records in the "videos_for_sale" editor and upload a few test videos (preferably in .mp4 format)

NOTE: You may have to change the maximum upload and maximum post sizes in your web server's PHP.ini file to allow for large uploads. (They're normally set to less than 10 Mb by default)

Now for the viewers.

On what I called the students page (students.php), include the following code. This will show a set of links to the video if one exists for that member:

In the head of this page, after the "load viewer library" call, add this "load records" call:

```
// load records
  list($videos_for_saleRecords, $videos_for_saleMetaData) = getRecords(array(
    'tableName'   => 'videos_for_sale',
  ));
```

Note: If you are pulling information from a single record editor on this page as well, you’ll have to remove the 'where'   => whereRecordNumberInUrl(1), line from the load records call.

And this error message call:

```
<?
php if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); } ?>
```

And in the body:

```
<!-- USER LOGIN FORM -->
<?php if (@$errorsAndAlerts): ?>
<div class="your_class">
<br />
<?php echo $errorsAndAlerts; ?>
<br />
</div>
<?php endif ?>

<?php if (!@$CURRENT_USER): ?>
<form action="?" method="post">
<input type="hidden" name="action" value="login" />

 <table width="400" border="0" align="center" cellpadding="2" cellspacing="0">
<tr>
<td class="your-class">Username</td>
<td><input type="text" name="username" value="<?
php echo htmlspecialchars(@$_REQUEST['username']); ?>" size="20" /></td>
</tr>
```

```
</tr>
<tr>
<td class="your_class">Password</td>
  <td><input type="password" name="password" value="<?
php echo htmlspecialchars(@$_REQUEST['password']); ?>" size="20"
/></td>
</tr>
<tr>
<td colspan="2" align="center">
<br />
<input type="submit" name="submit" value="Login" />
</td>
</tr>
<tr>
<td colspan="2" align="left">
<br /> <br />
<a class="your_class" href="<?
php echo $GLOBALS['WEBSITE_LOGIN_REMINDER_URL'] ?
>">FORGOT YOUR PASSWORD? CLICK HERE</a>
<br /> <br />
<span class="your_class">If you'd like to become a dance school member</span><br />
<a class="your_class" href="http://www.your_site.com/becomeamember.php">CLICK HERE TO SIGN UP</a>
</td>
</tr>
</table>
 </form>
<?php endif ?>
<!-- /USER LOGIN FORM -->

<!-- WELCOME MESSAGE-->
<?php if (@$CURRENT_USER): ?>
<span class="your_class">Welcome <?php echo $CURRENT_USER['first_name']; ?
>&nbsp;
<?php echo $CURRENT_USER['last_name']; ?>,</span>
<br /> <br />
<?php endif ?>
<!-- /WELCOME MESSAGE -->

<!-- VIDEO LINKS-->
<?php if (@$CURRENT_USER): ?>
<?php foreach ($videos_for_saleRecords as $record): ?><?
php if ($CURRENT_USER['num'] == $record['videos_for_sale']): ?>
<br />
<a class="your_class" href="<?php foreach ($record['video_upload'] as $upload): ?><?
php echo $upload['urlPath'] ?><?php
endforeach ?>">Download Your Personal<br />Training Video</a><br /> <br />
<a class="your_class"
href="http://www.your_site.com/videosdetail.php?<?php echo $record['num']; ?
>">Watch Your Personal<br />Training
Video</a>
<br /> <br />
<?php endif ?>
<!-- /VIDEO LINKS-->
```

And for the video detail page (videosdetail.php) where the student can view his or her video (using the free JWPlayer Video player described in the recipe “FREE JWPLAYER HANDLES BOTH HTML5 AND FLASH”).

In the head of the page, after the "load viewer" library call, add this "load records" call:

```
 // load records
  list($videos_for_saleRecords, $videos_for_saleMetaData) = getRecords(array(
    'tableName'   => 'videos_for_sale',
    'where'       => whereRecordNumberInUrl(1),
    'limit'       => '1',
  ));
  $videos_for_saleRecord = @$videos_for_saleRecords[0]; // get first record
```

And this error message call:

```
<?
php if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can
access this page."); } ?>
```

In the body, where you want the video to appear:

```
<!-- USER LOGIN FORM -->
  <?php if (@$errorsAndAlerts): ?>
    <div class="heading-text-13"><br />
      <?php echo $errorsAndAlerts; ?><br />
    </div>
  <?php endif ?>
```

```
<?php if (!@$CURRENT_USER): ?>

<form action="?" method="post">
  <input type="hidden" name="action" value="login" />

  <table width="400" border="0" align="center" cellpadding="2" cellspacing="0">
   <tr>
     <td  class="heading-text-13" colspan="2">You must be logged in to access this page.
</td></tr><tr>
    <td class="body-text-bold-9">Username</td>
    <td><input type="text" name="username" value="<?
php echo htmlspecialchars(@$_REQUEST['username']); ?>" size="20"
/></td>
</tr>
<tr>
<td class="body-text-bold-9">Password</td>
<td><input type="password" name="password" value="<?
php echo htmlspecialchars(@$_REQUEST['password']); ?>" size="20"
/></td>
</tr>
<tr>
<td colspan="2" align="center">
<br />
<input type="submit" name="submit" value="Login" />
 </td>
</tr>
<tr>
<td colspan="2" align="left">
<br /> <br />
<a class="special" href="<?php echo $GLOBALS['WEBSITE_LOGIN_REMINDER_URL'] ?
>">FORGOT YOUR PASSWORD? CLICK HERE</a>
<br /> <br />
<span class="body-text-9">If you'd like to become a dance school member</span>
<a class="special" href="http://www.your_site.com/becomeamember.php">CLICK HERE TO SIGN UP</a>
</td>
</tr>
</table>
</form>
<?php endif ?>

<!-- /USER LOGIN FORM -->

<!-- WELCOME MESSAGE -->
  <?php if (@$CURRENT_USER): ?>

<table width="70%" border="0" cellspacing="10" cellpadding="5">
<tr>
  <td valign="top" width="75%">
<table width="100%" border="0" align="center" cellpadding="5" cellspacing="0">
      <tr>
        <td colspan="2"><?php foreach ($videos_for_saleRecords as $record): ?><?
php if ($CURRENT_USER['num'] ==
$record['videos_for_sale']): ?>
          <span class="your_class">Welcome <?php echo $CURRENT_USER['first_name']; ?
>&nbsp;<?php echo
$CURRENT_USER['last_name']; ?>,</span>
<br />
<span class="your_class">Enjoy your video.</span>
<br /> <br />
<!-- /WELCOME MESSAGE -->
</td>
</tr>
<tr>
<td>
<!-- VIDEO DOWNLOAD LINK-->
<a class="your_class" href="<?php foreach ($record['video_upload'] as $upload): ?><?
php echo $upload['urlPath'] ?><?php
endforeach ?
>">Click here if you'd prefer to download<br />your personal training video</a>
<br />
<!-- /VIDEO DOWNLOAD LINK-->
     <?PHP endif ?>
<?php endforeach ?>
<br />
</td>
</tr>
</table>
 <?php endif ?>

<!-- DISPLAY THE VIDEO-->
 <?php if (@$CURRENT_USER): ?>

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

 <?php if ($CURRENT_USER['num'] == $record['videos_for_sale']): ?>
<table align="center" width="90%" border="0" cellpadding="5">
<tr>
<td>
<h2 align="center"><?php echo $record['title'] ?></h2>
```

```
 <h2 align="center"><?php echo $record['title'] ?></h2>
<br />
</td>
</td>
<tr>
<td align="center">

<div id="container">&nbsp;</div>
<script type="text/javascript">
jwplayer("container").setup({file: "http://www.your_site.com<?
php foreach ($record['video_upload'] as $upload): ?><?php
echo $upload['urlPath'] ?>
<?php endforeach ?>",
height: 420,
width: 510,
<?php if (@$record['background_image']): ?>image: "http://www.your_site.com<?
php foreach ($record['background_image'] as
$upload): ?><?php echo $upload['thumbUrlPath2'] ?>"<?php endforeach ?>, <?php endif ?
>
        autostart: true,
        modes: [
            { type: "html5" },
            { type: "flash", src: "player.swf" }
        ]

    });
</script>
 <!-- /DISPLAY THE VIDEO-->
<!-- DISPLAY THE VIDEO DESCRIPTION-->
<br /> <br />
</td>
</tr>
<tr>
<td class="your_class" align="left">
<?php echo $record['description'] ?>
</td>
</tr>
</table>
<!-- /DISPLAY THE VIDEO DESCRIPTION-->
<?php endif ?>
<?php endforeach ?>
<?php endif ?>
```

That should do it. You can style the pages any way you'd like.

CUSTOMIZING THE DATE FORMAT IN THE CMSB LIST VIEW - Apr 30th, 2011

This free plugin, by Robin Brayer from Interactive Tools is for all of you who would rather see “Saturday, April 30, 2011" instead of “2011-04-30 09:44:46" for dates that are displayed in your records list view.

Just copy the code below and paste it into a blank PHP document, save it as listViewDateFormat.php and upload it to your cmsAdmin/plugins folder. Then in the admin section, under plugins, activate your new plugin.

You can customize exactly how your date displays by changing the line $dateFormat = "l, F, d, Y"; in the plugin, following the guide in the Cookbook recipe called SHOW DATES ON YOUR PAGE

```
<?php


// register hooks
addFilter('listRow_displayValue', '_lvdf_changeDateFormat', null, 4);

function _lvdf_changeDateFormat($displayValue, $tableName, $fieldname, $record) {

  $returnValue = $displayValue;

  //Should we even consider this field?
  if(!($displayValueTimeStamp = @strtotime($displayValue))) {
    return $returnValue;
  }

// Determines the display format of the date field
  $dateFormat = "l, F, d, Y";

  $year = substr($displayValue, 0, 4);
  $firstHyphen = substr($displayValue, 4, 1);

  $month = substr($displayValue, 5, 2);
  $secondHyphen = substr($displayValue, 7, 1);

  $day = substr($displayValue, 8, 2);

  //echo $year . " " . $firstHyphen . " " . $month . " " . $secondHyphen . " " . $day;

  if(is_numeric($year) && is_numeric($month) && is_numeric($day) && ($firstHyphen == "-
") && ($secondHyphen == "-")) {
    $displayValueTimeStamp = strtotime($displayValue);
    if($displayValueTimeStamp > 1) {
      $returnValue = date($dateFormat, $displayValueTimeStamp);
    }
  }
  return $returnValue;
}

?>
```

REDIRECTING CURRENT USERS BASED ON THEIR MEMBER TYPE - Oct 5th, 2011

User John from JAM Graphics wanted to know if there was an easy way to redirect the retailer sales rep "members" to different parts of a website on login.

Robin Brayer from Interactive Tools came up with this simple answer:

If you add a "type" menu to the accounts section, that data will automatically be added to the CURRENT_USER variable ($CURRENT_USER['type']). You can use that to verify a users type at the top of a page.

On login, you can set your website membership "WEBSITE_LOGIN_POST_LOGIN_URL" to direct to something like "logindispatch.php". Then redirect users from there to the correct place. e.g. The code might look something like:

```
if($CURRENT_USER['type']=='retailer') {
  redirectBrowserToURL('retailerpage.php');
}
```

SENDING EMAILS TO "MEMBERS" - Feb 7th, 2012

When Christopherb wanted to be able to send emails to all members using CMS Builder, Jason Sauchuk from Interactive Tools had this suggestion.

According to Jason:

There's a function in CMS Builder called sendMessage() that lets you send emails.

What you can do is set up an HTML form that has 2 fields, one called "subject" and then one called "message". In the action of this form, you can have the form submit to itself (ie, action = "?").

At the top of your script you can use this code:

Code

```
if (@$_REQUEST['subject'] && @$_REQUEST['message'] && $CURRENT_USER['isAdmin']) {

  list($membersRecords, $membersMetaData) = getRecords(array(
    'tableName'        =>     'accounts',
    'allowSearch'      =>     false,
  ));

  $emailList = join(",", array_filter(array_pluck($memberRecords, 'email')));

  sendMessage(array(
    'to'      =>  "to@email.com",
    'from'    => "from@email.com",
    'subject' => @$_REQUEST['subject'],
    'text'    => @$_REQUEST['message'],
    'headers' => array(
           'BCC'  =>  $emailList,
         ),
  ));

}
```

With the code && $CURRENT_USER['isAdmin'] included, , this code will only execute if the current user is an administrator and both subject and message were given a value.

What the script does then is get all of the users in "accounts" and create a comma separated list of their emails. It then sends out an email with all of these emails in it's BCC field. You'll need to change the value of the "to" and "from" field to whatever emails you'd like to use.

SENDING EMAILS TO A SPECIFIC GROUP OF USERS - Aug 6th, 2012

If you're using an email form and sendmail to create and send emails, the manual way to add recipients to the list would be something like this:
NOTE:
I haven't checked this out so let me know if there are issues.

```
  'To'  => email1@example.com,email2@example.com
```

If all of your potential recipients have user accounts, you might also think about changing the To values to a variable ($emailList) that gets populated from user email addresses where there's a "notifications" check box checked in their record.

Something like this:

First add a check box field called notifications to your user account editor.

Then, create your list from the accounts records.

```
   <?php list($emailListRecords,) = getRecords(array(
     "tableName"         => "accounts",

     'where'      => "notifications = '1' AND hidden = '0'",
     'orWhere'      => "notifications = '1' AND hidden = '1'",
   ));

     $emailList = "";

     foreach( $emailListRecords as $record ) {

       $emailList.= $record['email'] .",";
    }
?>
```

Then use the variable in your sendmail code,

```
'To'      => $emailList,
```

Another way of listing all of the notification email addresses, say in a viewer, would be:

```
<?php // Get email list of those members with notifications on, and access to the project

          $accounts  = mysql_query_fetch_all_assoc("SELECT * FROM `{$TABLE_PREFIX}accounts` WHERE notifications = '1'");
     $emailList = join(",", array_pluck($accounts, 'email'));
    ?>

    <?php echo $emailList ;?>
```

ALPHABETICALLY DISPLAYING THE ENTRIES IN THE CREATEDBY SELECT USER PULL DOWN - Jan 3rd, 2014

Alphabetically displaying the entries in the createdBy select user pull down

To make it easier to re-

To make it easier to re
assign author access only listings that were originally created by admins, I wanted to force the entries createdBy select user pull down in a record to display in alphabetical order.

Dave Edis, senior developer at Interactive Tools came up with an easy, one line fix (and will be adding it to the code starting with CMSB versions above 2.53).

In cmsAdmin/lib/menu/default/common.php

Look for the code:

```
// show pulldown when editor clicks "change" beside "Created By" on edit page in CMS
function ajaxGetUsersAsPulldown() {
global $TABLE_PREFIX, $hasEditorAccess, $tableName;
if (!$hasEditorAccess) { return ''; } // must have section admin access

// get users with access to this section
$query = "SELECT u.num, u.username
FROM {$TABLE_PREFIX}accounts u
JOIN {$TABLE_PREFIX}_accesslist a ON u.num = a.userNum
WHERE a.accessLevel > 1 AND a.tableName IN ('all','$tableName')
$users = mysql_fetch($query);
```

And add:

```
ORDER BY username";
```

above the line, $users = mysql_fetch($query);

PICKING MEMBERSHIP RENEWAL DATE AND DURATION - Apr 20th, 2015

Instead of calculating an expiration date (expiresDate) based on the date that a member signed up, an organization wanted to calculate the expiration date and membership duration (in years) from a fixed month.

Further, they wanted anyone signing up after a particular cutoff month relative to the current month, to get an extra year added to their membership, so that their initial membership period was never shorter than one year.

Some of the other challenges were calculating a valid date for months that had 30 or 28 days vs those that had 31 days, and generating a valid MySQL date from the calculation.

With Claire Ryan's and Greg Thomas' from Interactive Tools help, we came up with the following solution.

In a single record editor (mine was called 'Organization Information') there are 2 list fields, a text field, and a check box . The list fields are called Start Month and Cutoff Month. Their values are:

```
1|January
2|February
3|March
4|April
5|May
6|June
7|July
8|August
9|September
10|October
11|November
12|December
```

NOTE: the format number|text allows the list to display the text in the pull down menu, while inserting the selected value in numbers.

The text field is called Renewal Duration and accepts 1 digit with a value from 1-9. With the description: Enter the digit corresponding to the Renewal Duration in years (1-9).

The check box (called 'Valid for One Year From Date Joined'), with the description: "If renewal period begins on date joined, check this box. Otherwise the 3 fields below will determine the renewal criteria."

Then define a few variables:

```
<?php
@$curr_month = date("n") ;
@$start_month      = str_pad(@$organization_informationRecord['start_month'], 2, "0", STR_PAD_LEFT);
@$cutoff_month = @$organization_informationRecord['cutoff_month'] ;
@$renewal_duration = @$organization_informationRecord['renewal_duration'] ;
@$currentYear = date('Y');
?>
```

After that, perform the calculations required and check for valid last day of the month :

```
 <?php
@$renewalYear = (intval(@$curr_month) >= intval(@$cutoff_month))? @$currentYear + (@$renewal_duration) :@ $currentYear +
(@$renewal_duration - 1) ;
@$expiryMonth = ($organization_informationRecord['start_month'] -1);
  if(@$expiryMonth < 10) {
    @$expiryMonth = "0".@$expiryMonth;
  }
  if(@$expiryMonth == 00) {
    @$expiryMonth = 12;
  }
   @$testEndDate = @$renewalYear. '-'.@$expiryMonth.'-01';
  @$lastDayofMonth = date('t', strtotime(@$testEndDate));
    if(@$expiryMonth == 02) {
    @$lastDayofMonth = 28;
  }
  if(@$expiryMonth == 04 || @$expiryMonth == 06 || @$expiryMonth == 09 || @$expiryMonth == 11) {
    @$lastDayofMonth = 30;
}
if(@$expiryMonth == 01 || @$expiryMonth == 03 || @$expiryMonth == 05 || @$expiryMonth == 07 || @$expiryMonth == 08 ||
@$expiryMonth == 10 || @$expiryMonth == 12 ) {
    @$lastDayofMonth = 31;
}
?>
```

And then create a valid MySQL date from the result:

```
<?php $adjustedMembershipExpiryDate = $renewalYear. '-'.$expiryMonth.'-
'.$lastDayofMonth.' 00:00:00'; ?>
```

You can use the following to test your progress.

```
Current Month is: <?php echo date("n"); ?><br />
Test End Date is: <?php echo @$testEndDate ; ?><br />
Start Month is: <?php echo @$start_month ; ?><br />
Expiry Month is: <?php echo @$expiryMonth ; ?><br />
Last Day of Month: <?php echo @$lastDayofMonth ?><br />
Cutoff Month is: <?php echo @$cutoff_month ?><br />
Renewal Duration is:<?php echo @$organization_informationRecord['renewal_duration'] ?
> years<br />
Current Year is: <?php echo date('Y'); ?><br />
Renewal Year is: <?php echo @$currentYear + (@$renewal_duration - 1); ?><br />
Calculated $adjustedMembershipExpiryDate is: <?
php echo @$adjustedMembershipExpiryDate ?> <br />

<?php if(@$curr_month >= @$cutoff_month): ?>
It's <?php echo @$cutoff_month ?> or after, so use renewal_duration  (<?
php echo @$renewal_duration ?>) as year interval
added to current year.
<?php else:?>
It's before <?php echo @$cutoff_month ?>, so use renewal_duration -1 (<?
php echo @$renewal_duration -1 ?>) as year
interval added to current year.
<?php endif ?>
```

The final step is to insert the $adjustedMembershipExpiryDate into the mysql_query("INSERT INTO list. NOTE: the apostrophes are so that the variable is recognized correctly.

```
 expiresDate      = '$adjustedMembershipExpiryDate',
```

Then, to allow for the, “membership from date joined” option, I surrounded the calculation code and the mysql_query("INSERT INTO code with:

```
<?php if ( $organization_informationRecord['valid_for_one_year_from_date_joined'] = 0 ||
$organization_informationRecord['valid_for_one_year_from_date_joined'] = '' ):?>
... date calculation code and your  mysql_query("INSERT INTO code, including  expiresDate      =
'$adjustedMembershipExpiryDate', ...
 <?
php elseif (@$organization_informationRecord['valid_for_one_year_from_date_joined'] == '1'):?
>
...Your original mysql_query("INSERT INTO code, including expiresDate      = (NOW() + INTERVAL 1 YEAR),...
<?php endif ?>
```

MORE EXPIRATION DATE CODING OPTIONS - Oct 6th, 2016

Automatically updating an annual membership can get complicated because some organizations still prefer to use a specific renewal date, and others allow the actual date that a member has signed up as the base date and annually update from there.

The situation becomes more complex when a member has skipped a year and then decides to renew.

Here’s some really simple MySQL update code as well as some more sophisticated coding examples thanks to Dave Edis, Senior developer at interactive Tools.

Example 1
This will simply add one year to the current expiration date:

```
<?php
 mysqlStrictMode(false);
 mysql_query("UPDATE `{$TABLE_PREFIX}accounts`
       SET  expiresDate = expiresDate + INTERVAL 1 YEAR
   WHERE num = '".mysql_escape( $CURRENT_USER['num'] )."'")
      or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
       $userNum = mysql_insert_id();

?>
```

Example 2
This will simply update the expiration date to one year from the actual date the membership is renewed:

```
<?php
 mysqlStrictMode(false);
 mysql_query("UPDATE `{$TABLE_PREFIX}accounts`
       SET  expiresDate      = (NOW() + INTERVAL 1 YEAR)
   WHERE num = '".mysql_escape( $CURRENT_USER['num'] )."'")
      or die("MySQL Error:\n". htmlspecialchars(mysql_error()) . "\n");
       $userNum = mysql_insert_id();

?>
```

Example 3 (thanks to Dave Edis)
This will update a member's expiration date to 1 year from the current expiration date.
But, if the expiration date does not contain the previous year, then the new expiration date will be the current year +1 (so that the record for a member who has skipped a renewal year does not get updated to last year).

```
<?php
 mysqlStrictMode(false);

$lastYear    = date('Y', strtotime("-1 year"));
$expiryYear  = date('Y', strtotime($CURRENT_USER['expiresDate']));
$currentYear = date('Y');
if ($expiryYear != $lastYear) { $newExpiryYear = $currentYear + 1; }
else                          { $newExpiryYear = $expiryYear + 1; }

// Update expiry to greater of: 1 year from now, 1 year from old expires date
$updateSet = "expiresDate=DATE_FORMAT(expiresDate,'$newExpiryYear-%m-%d %T')";
$updateWhere = "num = '".mysql_escape( $CURRENT_USER['num'] )."'";
$updateQuery = "UPDATE `{$TABLE_PREFIX}accounts` SET $updateSet WHERE $updateWhere";
mysql_query($updateQuery) or die("MySQL Error: ". htmlspecialchars(mysql_error()) . "\n");
?>
```

Example 4 (thanks to Dave Edis)
This will update the expiration date to the greater of: 1 year from now, or 1 year from the old expiration date:

```
<?php
 mysqlStrictMode(false);

$updateSet   = "expiresDate = GREATEST(NOW(), expiresDate) + INTERVAL 1 YEAR";
$updateWhere = "num = '".mysql_escape( $CURRENT_USER['num'] )."'";
$updateQuery = "UPDATE `{$TABLE_PREFIX}accounts` SET $updateSet WHERE $updateWhere";
mysql_query($updateQuery) or die("MySQL Error: ". htmlspecialchars(mysql_error()) . "\n");
?>
```

CRON JOBS

CREATING A CRON JOB - Apr 10th, 2019

Cron is a time-based job scheduler in Unix-like computer operating systems. The name cron comes from the word "chronos", Greek for "time". Crontab jobs allow users to schedule when a particular script is executed based on matching month, day, hour, and minute.

I'm using a crontab job or "cron job" to schedule when a PHP "membership renewal reminder" script is run.

The PHP script, which is run once a day, compares the expiresDate with the current date for each member. If the number of days until expiration matches a preset number, the member is sent an email reminding them that their membership is about to expire.

Crontab jobs are pretty particular in how they're set up, and ISPs differ in the way that you access and input the information to set up a cron job, so you may have to get in touch with their tech support to get things operating as they should. My ISP (IXWebHosting) sets up cron jobs in the ftp management section of their interface.

There's specific syntax for entering the scheduling information into a cron job request. At the end of this recipe, you'll find the acceptable parameters from my web host IXWebHosting, taken straight from their "help" screen.

NOTE: You'll have to insure that the script that you're trying to run with the conjob has write permission

CREATING AND RUNNING A CRON JOB

There are 3 pieces of information necessary for creating and running a cron job
1) the scheduling information
2) the path to your server's php interpreter (obtained from your web host), and
3) the path to the php file to be executed

I needed to run my script once a day at 2 minutes after midnight, so my scheduling parameters are:
Minute: 2, Hour: 0, Day of month: 1-31, Month: 1-12, Day of week: 1-7

After you've set up all your scheduling parameters, either put both the path to your server's php interpreter, and the path to your PHP script in your crontab command line as in this example (Note: The path to your server's php interpreter will be different, so check with your web host, and don't forget the space between the 2 segments.):

*** Note: if you have issues with sending mail using cron jobs, see the recipe under Bugs and Fixes called: "Cron issues sending mail with CMSB 3.50 or earlier, and PHP 7.2 after a Bluehost server upgrade"

```
/hsphere/shared/php5/bin/php -q /your_server_path/mailtest.php
```

or, put the path to your server's php interpreter at the very top of your php script with no blank spaces or lines before it, like this (note: The path to your server's PHP interpreter will be different, so check with your web host.):

```
#!/hsphere/shared/php5/bin/php-cgi -q
```

And insert only the path to your php script in the command line, like this:

```
/your_server_path/mailtest.php
```

TESTING SENDMAIL
To test if your sendmail is working you can create a .php file called mailtest.php with this simple php sendmail test script. When you refresh the page it should send the email.

```
<?php
$to = "your_email@your_host.com";
$from = "you@sendmail_test.com";
$subject = "Testing Sendmail";
$txt = "Testing Sendmail from PHP Script";
mail($to,$subject,$txt,"From: $from", "-f$from");
?>
```

NOTE: You'll need to make sure that this file has a permission of 755 (read, write, execute) or the file may not run when called by the cron manager (your cron daemon confirmation email will show an "access denied" error).

TESTING YOUR CRON JOB
Once you're sure that sendmail is working, you can use the script to test your cronjob.

CRON JOB CONFIRMATION EMAILS
More than likely, your Cron Daemon will send a confirmation email to an address you choose and you can use the information in that to debug your cron job and your scripts

Once you're sure everything is working, you can send the daemon emails to a non existent email address so that they will not clutter up your inbox.

CRON ISSUES SENDING MAIL WITH CMSB 3.50 OR EARLIER, AND PHP 7.2 AFTER A BLUEHOST SERVER UPGRADE - May 28th, 2019

About a month ago, I moved an account to a new server at Bluehost, and I started having major problems with sending emails using a cron job.

After weeks of playing around, many days waiting for Bluehost's senior level support, and hours of Daniel Loewe and Dave Edis' time, The issue of why emails were never received when using a mailarray or an email template, is solved. (The problem didn't affect phpmail)

Bluehost narrowed it down to Message-ID issues similar to: <65812fc901465e02d09689bdaf6a902f@swift.generated>.

Seems, according to Dave Edis, the senior guru at Interactive Tools, (as well as I understand it) if there's no $_SERVER['SERVER_NAME'] set. See: **https://forge.typo3.org/issues/24686**, a header is generated by the swift mail program.

Daniel worked up a fix, and at least part of it will be rolled out in a future version of CMSB

Here's the fix that Daniel created.

1) Add your domain name to the end of your cron job (after a space, and no http or www)
2) Add the following code to the top of the cron.php file (above "//load viewer library" code) (i put mine after the load viewer library code by mistake and it still worked)

```
 if (empty($_SERVER['HTTP_HOST']) && isset( $_SERVER['argv'][1])) {
 $_SERVER['SERVER_NAME'] = $_SERVER['argv']
[1]; // set server name for SwiftMailer Message-ID in cron tasks //
(https://forge.typo3.org/issues/24686)
}
```

3) Add the same code to any page that addresses CMSB's sendMessage() function from outside the background task framework (like mailarray or email template triggering scripts)

Daniel said, "The extra code should only be necessary when using CMSB's sendMessage() function. it's worth noting that this fix will be in cron.php in a future CMSB version (either 3.51 or 3.52), so after that, it will only be applicable for users setting up cron jobs that use sendMessage() outside of CMSB's background tasks or users on older versions."

Hope this save someone a bit of aggravation....

SENDING MEMBERSHIP REMINDER EMAILS USING A CRON JOB - Apr 10th, 2019

SENDING THE MEMBERSHIP RENEWAL REMINDER EMAILS
And once you've got your cron job working, you can substitute the PHP script that sends reminder emails to your members.

It uses the mail function built in to CMS Builder which seems to be more robust then using sendmail directly.

*** Note: if you have issues with sending mail using cron jobs, see the recipe under Bugs and Fixes called: "Cron issues sending mail with CMSB 3.50 or earlier, and PHP 7.2 after a Bluehost server upgrade"

For reference, the basic implementation for the mail function is:

```
$message = 'This is a test message';
$mailArray = array(
'to' => 'example@example.com',
'from' => 'example@example.com',
'subject' => 'This is a subject',
'html' => $message
);
$errors = sendMessage($mailArray);
```

THE MEMBERSHIP RENEWAL REMINDER SCRIPT
Here's the PHP script that I run once a day to send a renewal reminder emails to members:
A) when their membership is 30 days from expiration, and
B) when their membership is one week from expiration.

To allow my clients to change both the reminder schedule and the text of the reminder emails that are sent, I created 2 text fields and 2 text boxes in a single record editor I named "Common Information". The text fields are named first_reminder_day and second_reminder_day, the text boxes are named first_reminder_message and second_reminder_message.

The script allows for html code to be added to the email so that the emails can be styled. It also includes code from the recipe: "INCLUDING VARIABLES IN THE CONTENTS OF A TEXT BOX" so that the current annual renewal fee can be pulled from another table called "become_a_member" and included in the "reminder message" fields.

NOTE: don't forget to change the path to your server's PHP interpreter and to revise the paths to your server and any links. Also, make sure that this file has a permission of 755 (read, write, execute) or the file may not run when called by the cronjob manager.

```
#!/hsphere/shared/php5/bin/php-cgi -q
<?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($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName'   => 'common_information',
    'allowSearch' => '0',
    'limit'       => '1',
  ));
  $common_informationRecord = @$common_informationRecords[0]; // get first record

  list($accountsRecords, $accountsMetaData) = getRecords(array(
    'tableName'   => 'accounts',
  ));
  list($become_a_memberRecords, $become_a_memberMetaData) = getRecords(array(
    'tableName'   => 'become_a_member',
     'allowSearch' => '0',
    'limit'       => '1',
  ));
   $become_a_memberRecord = @$become_a_memberRecords[0]; // get first record
  ?>
 <?php foreach($accountsRecords as $record) : ?>

<?php
  $secondsOld = time() - strtotime($record['expiresDate']); // seconds to or since expiration date
  $daysOld    = intval($secondsOld/60/60/24); // converts seconts to days
  $first_reminder_day = $common_informationRecord['first_reminder_day'] ;
  $second_reminder_day = $common_informationRecord['second_reminder_day'] ;
  // replace the variable *regular_dues_amount* with the annual dues
  $first_reminder_message =
str_replace('*regular_dues_amount*',$become_a_memberRecord['regular_dues_amount'],$common_informationRecord['first_reminder_message']);
  $second_reminder_message =
str_replace('*regular_dues_amount*',$become_a_memberRecord['regular_dues_amount'],$common_informationRecord['second_reminder_message']);
  $member = $record['first_name'] ;

?>


 <?php ob_start(); // start capturing output ?>

      <?php $first_reminder_message = wordwrap( $first_reminder_message, 70); ?>
      <?php echo $first_reminder_message ?>
      <?php $output = ob_get_clean(); // stop capturing output  ?>

       <?php ob_start(); // start capturing output ?>

      <?php $second_reminder_message = wordwrap( $second_reminder_message, 70); ?

>
      <?php echo $second_reminder_message ?>
      <?php $output2 = ob_get_clean(); // stop capturing output  ?>


<?
php if((!@$record['neverExpires'] && @$record['lifetime_member'] == '0') && $record['remove_me'] == '0' && ($daysOld
== -$first_reminder_day)) :?>

 <?php   $the_to = $record['email']; ?>

 <?php

 $message = <<<EOF
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type content='text/html; charset=utf-8" />
<style type="text/css">
.special {font-family:Arial; color:  rgb(227,224,219); font-size: 1.0em; font-weight: bold; text-
decoration: underline;}
.body-text {font-family:Arial; font-size: 1.0em;}
.body-text-yellow {font-family:Arial; color: rgb(255,255,0); font-size: 1.0em;}
.heading-text-yellow {font-family:Arial; color: rgb(255,255,0); font-size: 1.3em;}
</style>
</head>
<body bgcolor="#7C7164">
<table style='background-color: #7C7164;' width='100%' align='center' border='0' >
             <tr>
    <td align='left' >
    <table style='background-color: #7C7164;' width='70%' align='center' border='0' >
             <tr>
    <td align='left' >
 <div align='left'><img src='http://www.artistsofpalmbeachcounty.org/images/APBC-
LOGO.png' width='800' height='183'
style='border:hidden'/></div>
<br /> <br />
 <div align='left' class='body-text'>
  Hello $member,
  <br /> <br />

 $output \r\n
<br /> <br />
  Best,
  <br /> <br />
 The Membership Committee <br />
 Artists of Palm Beach County\r\n</div>
  </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
  </body>
</html>
EOF;
 $the_from  = "membership@artistsofpalmbeachcounty.org";
 $the_subject = "Your Current Artists of Palm Beach County Membership will expire in $first_reminder_day days";

$mailArray = array(
'to' => $the_to,
'from' => $the_from,
'subject' => $the_subject,
'html'        => $message
);
$errors =  sendMessage($mailArray);
// delay execution by .2 sec
usleep(200000);
// reset max_execution_time to 30 sec

set_time_limit(30);
?>

<?php endif ;?>

 <?
php if((!@$record['neverExpires'] && @$record['lifetime_member'] == '0') && $record['remove_me'] == '0' && ($daysOld
== -$second_reminder_day)) :?>
<?php   $the_to = $record['email']; ?>
 <?php

 $message = <<<EOF
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type content='text/html; charset=utf-8" />
<style type="text/css">
.special {font-family:Arial; color:  rgb(227,224,219); font-size: 1.0em; font-weight: bold; text-
decoration: underline;}
.body-text {font-family:Arial; font-size: 1.0em;}
.body-text-yellow {font-family:Arial; color: rgb(255,255,0); font-size: 1.0em;}
.heading-text-yellow {font-family:Arial; color: rgb(255,255,0); font-size: 1.3em;}
</style>
</head>
<body bgcolor="#7C7164">
<table style='background-color: #7C7164;' width='100%' align='center' border='0' >
              <tr>
    <td align='left' >
    <table style='background-color: #7C7164;' width='70%' align='center' border='0' >
              <tr>
    <td align='left' >
 <div align='left'><img src='http://www.artistsofpalmbeachcounty.org/images/APBC-
LOGO.png' width='800' height='183'
style='border:hidden'/></div>
<br /> <br />
 <div align='left' class='body-text'>
  Hello $member,
  <br /> <br />

 $output2 \r\n
<br /> <br />
  Best,
  <br /> <br />
 The Membership Committee <br />
 Artists of Palm Beach County\r\n</div>
  </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
  </body>
</html>
EOF;
 $the_from  = "membership@artistsofpalmbeachcounty.org";
 $the_subject = "Your Current Artists of Palm Beach County Membership will expire in $second_reminder_day days";

$mailArray = array(
'to' => $the_to,
'from' => $the_from,
'subject' => $the_subject,
'html'         => $message
);
$errors =  sendMessage($mailArray);
// delay execution by .2 sec
usleep(200000);
// reset max_execution_time to 30 sec
set_time_limit(30);
?>


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

THE CONTENTS OF MY REMINDER MESSAGE TEXT BOXES

We wanted to let you know that your membership will expire in about a week.

To continue enjoying the benefits of membership, please <a href='**http://www.your_site.com/paypal.php**'   / >CLICK HERE</a> to renew for another year for only $*regular_dues_amount* using PayPal.

and

We wanted to let you know that your membership will expire in about a month.

To continue enjoying the benefits of membership, please <a href='**http://www.your_site.com/paypal.php**'   / >CLICK HERE</a> to renew for another year for only $*regular_dues_amount* using PayPal.

GOING FURTHER
For situations where the sending of repetitive emails need to be avoided, Dave Edis from Interactive Tools suggested setting up check boxes that the PHP script writes to when an email is sent and then including a compare in the if statements to determine the check boxes value (1=sent, 0=not sent). he reminded that you'd also need to change the value of those check boxes to "0" on renewal so the process could repeat.

If you're getting error messages in your Cron Daemon emails like these:

```
Notice: Undefined index:  SCRIPT_FILENAME in
/hsphere/local/home/apbcweb/artistsofpalmbeachcounty.org/cmsAdmin/plugins/createPDF/createPDF.php on line 22

Notice: Undefined index:  HTTP_HOST in
/hsphere/local/home/apbcweb/artistsofpalmbeachcounty.org/cmsAdmin/plugins/createPDF/createPDF.php on line 23

Notice: Undefined index:  SCRIPT_NAME in
/hsphere/local/home/apbcweb/artistsofpalmbeachcounty.org/cmsAdmin/plugins/createPDF/createPDF.php on line 23

Notice: Undefined variable: _SESSION in
/hsphere/local/home/apbcweb/artistsofpalmbeachcounty.org/cmsAdmin/lib/login_functions.php on line 88
```

Dave Edis offers a solution:

He says:

For the Undefined notices, a workaround might be to edit /lib/init.php and add this line just above "// define constants"

```
error_reporting(E_ALL & ~E_NOTICE); // show all errors but notices

// define constants
```

Cron Tab Help Document (IXWebhosting.com)

The format of a cron command is very much the V7 standard, with a number of upward-compatible extensions. Each line has
five time and date fields, followed by a user name if this is the system crontab file, followed by a command. Commands are executed by cron(8) when the minute, hour, and month of year fields match the current time, and when at least one of the two day fields (day of month, or day of week) match the current time (see ``Note'' below). Note that this means that non-existent times, such as "missing hours" during daylight savings conversion, will never match, causing jobs scheduled during the "missing times" not to be run. Similarly, times that occur more than once (again, during daylight savings conversion) will cause matching jobs to be run twice.

cron(8) examines cron entries once every minute.

The time and date fields are:

field	allowed values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

A field may be an asterisk (*), which always stands for ``first-last''.

Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive. For example, 8-11 for an ``hours'' entry specifies execution at hours 8, 9, 10 and 11.
Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples:
``1,2,5,9'', ``0-4,8-12''.

Step values can be used in conjunction with ranges. Following a range with ``/'' specifies skips of the number's value through the range. For example, ``0-23/2'' can be used in the hours field to specify command execution every other hour
(the alternative in the V7 standard is ``0,2,4,6,8,10,12,14,16,18,20,22''). Steps are also permitted after an asterisk, so if you want to say ``every two hours'', just use ``*/2''.

Names can also be used for the ``month'' and ``day of week'' fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.

The ``sixth'' field (the rest of the line) specifies the command to be run. The entire command portion of the line, up to a newline or % character, will be executed by /bin/sh or by the shell specified in the SHELL variable of the cronfile. Percent-signs (%) in the command, unless escaped with backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input.

Note: The day of a command's execution can be specified by two fields -
- day of month, and day of week. If both fields
are restricted (ie, aren't *), the command will be run when either field matches the current time. For example, ``30 4 1,15 * 5'' would cause a command to be run at 4:30 am on the 1st and 15th of each month, plus every Friday.

SENDING INFORMATION UPDATE REMINDER EMAILS USING A CRON JOB - Feb 12th, 2023

The information in an on-line directory is only as good as that input by the organizations that are listed.
For that reason, my client wanted to hide any listing that has not been updated in either 6 month or a year intervals.
They also wanted to send emails to each listing organization 30 days before their update was due, another 10 days before the update was due, and when their listing was automatically hidden form the search results.

This was accomplished by using a reminder email PHP script that is called once a day by a cron job.
My Cron job, executed at 5AM every day, looked like this:

```
php -q /home3/mrqsygmy/public_html/reminder_email.php
```

The actual hiding of listings is directly implemented on the search page which is documented at the end of this recipe.

THE INFORMATION UPDATE REMINDER SCRIPT
The script requires the following fields to be created in a single record editor (mine is called 'common_information'):
an 'are mandatory reminders active' checkbox called 'reminder' (checked = Yes, unchecked = No)
a mandatory update duration checkbox called ''mandatory_update_duration' (checked = one year, unchecked = 6 months)
a first reminder day text field called 'first_reminder_day' (for 30 days or your interval)
a second reminder day text field called 'second_reminder_day' (for 10 days or your interval)
a 'first reminder message' text box field called 'first_reminder_message' (for 30 days or your interval)

a 'second reminder message' text box field called 'second_reminder_message' (for 10 days or your interval)
a 'thiird reminder message 'text box field called 'third_reminder_message' (for hidden from directory message)
NOTE: *** It's important that your message includes that they need to log in to and RE-SAVE their account record.***
a 'send test emails only' checkbox called 'send_test_emails_only' (checked = Yes (test email), unchecked = No (live)
a 'test email' text field called 'test_email' (for the email address where you'd like to recieve tests)
a main box background color' text field called 'main_box_background_color' (for formatting the color of the emails)

and a 'test date' date field in the Account record called 'test_date' (the live script depends on the field 'updatedDate' to determine when emails are due. The 'test_email' field in the account record allows for testing
and a 'Hide This Account From Directory Searches and Email Lists' checkbox field in the Account record called 'notAdmin'

Here’s the PHP script that I run once a day to send a information update reminder emails to organizations:

```
?php $libraryPath = 'cmsAdmin/lib/viewer_functions.php';
  $dirsToCheck = array('/home3/mrqsygmy/public_html/dbtproviders/','','../','../../','../../../');
  foreach ($dirsToCheck as $dir) { if (@include_once("$dir$libraryPath")) { break; }}
  if (!function_exists('getRecords')) { die("Couldn't load viewer library, check filepath in sourcecode."); }
 if (empty($_SERVER['HTTP_HOST']) && isset( $_SERVER['argv'][1])) {
$_SERVER['SERVER_NAME'] = $_SERVER['argv']
[1]; // set server name for SwiftMailer Message-ID in cron tasks
(https://forge.typo3.org/issues/24686)
}
  // load records
       list($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName'   => 'common_information',
     'where'       => '',
     'allowSearch' => false,
    'limit'       => '1',
  ));
  $common_informationRecord = @$common_informationRecords[0]; // get first record
<?php  list($accountsRecords, $accountsMetaData) = getRecords(array(
    'tableName'   => 'accounts',
  ));
?>
<?php $color = $common_informationRecord['main_box_background_color'] ?>
<?php $currentUnixTime = strtotime(date('M jS, Y')); ?>
<?php // $timeGone = 0 ?>
<?php @$first_reminder_day = $common_informationRecord['first_reminder_day'] ;?>
<?php @$first_reminder_day_two =  @$first_reminder_day * 86400 ?>
<?
php @$second_reminder_day = $common_informationRecord['second_reminder_day'] ;?>
<?php @$second_reminder_day_two =  @$second_reminder_day * 86400 ?>
<?php // $timeGone = 0 ?>
<?php $emailCountTwoFour = 0 ?>
<?php $emailCountOneThree = 0 ?>
<?php foreach($accountsRecords as $record) : ?>
<?php $timeGone = 0 ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +6 month") ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])) . " +6 month") ?><?php endif ?>
```

```
<?php $sixMonthsFirst = ($sixMonths - $first_reminder_day_two) ?>
<?php $sixMonthsSecond = $sixMonths - $second_reminder_day_two ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +1 year") ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])) . " +1 year") ?><?php endif ?>
<?php $twelveMonthsFirst = ($twelveMonths - $first_reminder_day_two) ?>
<?php $twelveMonthsSecond = $twelveMonths - $second_reminder_day_two ?>
<?php if ($sixMonthsFirst == $currentUnixTime ):?>

<?php $timeGone = 1 ?>
<?php endif ?>
<?php if ($sixMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 2 ?>
<?php endif ?>
<?php // elseif ($common_informationRecord['mandatory_update_duration']=='1'):?>
<?php if ($twelveMonthsFirst == $currentUnixTime ):?>
<?php $timeGone = 3 ?>
<?php endif ?>
<?php if ($twelveMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 4 ?>
<?php endif ?>
<?php  // endif?>
<?php if(!@$record['isAdmin'] && $common_informationRecord['reminder'] == 1): // live ?>
<?php  if($timeGone==1 || $timeGone==3):  ?>
<?php
  @$first_reminder_message = $common_informationRecord['first_reminder_message']; // 10 calendar days
  @$second_reminder_message = $common_informationRecord['second_reminder_message']; // 30 calendar days
   @$practiceName = $record['practice_name'] ;
  @$firstName = $record['contact_first_name'] ;
    @$lastName = $record['contact_last_name'] ;?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $expires = date("M jS, Y",
strtotime($record['test_date'])) ;?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0 ||
$common_informationRecord['send_test_emails_only'] == ''):?><?
php $expires = date("M jS, Y",
strtotime($record['updatedDate'])) ;?><?php endif ?>
<?
php // ************** START SET PROVIDER VARIABLES AND OUTPUT **************************  ?
>
<?php ob_start(); // start capturing output ?>
<?php $first_reminder_message = wordwrap( $first_reminder_message, 70); ?>
<?php echo $first_reminder_message ?>
<?php $output = ob_get_clean(); // stop capturing output  ?>
<?php ob_start(); // start capturing output ?>
<?php $second_reminder_message = wordwrap( $second_reminder_message, 70); ?>
<?php echo $second_reminder_message ?>
<?php $output2 = ob_get_clean(); // stop capturing output  ?>
<?
php //************** END SET PROVIDER VARIABLES AND OUTPUT ************************** ?
>
<?php  // **************DETERMINE PROVIDER SATUS ****************** ?>
<?php // Send Test or Real emails? ?>
<?
php if ($common_informationRecord['send_test_emails_only'] == 0 || $common_informationRecord['send_test_emails_only']
== ''):?>
<?php   $the_to = $record['email']; // live?>
<?php elseif ($common_informationRecord['send_test_emails_only'] == 1):?>
<?php @$the_to = $common_informationRecord['test_email']; ?>
<?php endif ?>
<?php
$message = <<<EOF
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type content='text/html; charset=utf-8" />
<style type="text/css">
.special {font-family:Arial; color:  rgb(0,0,0); font-size: 1.0em; font-weight: bold; text-
decoration: underline;}
.text_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.0em;}
.heading_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.3em;}
.heading_font_blue {font-family:Arial; color: rgb(0,0,255); font-size: 1.3em;}
</style>
</head>
<body bgcolor="#$color">
<table style='background-color: #$color;' width='100%' align='center' border='0' >
             <tr>
    <td align='left' >

    <table style='background-color: #$color;' width='70%' align='center' border='0' >
             <tr>
    <td align='left' >
 <div align='left'>
<img src='http://www.dbtproviders.com/images/DBT_email_masthead.png' style='border:hidden;
width:489px; height:88px;'/></div>
 <div align='left' class='text_font'>
  Hello $firstName $lastName,
 $output \r\n
  Best,

  The DBT Provider Directory Team
\r\n\r\n</div>
  </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
  </body>
</html>
EOF;

 $the_from  = "no-reply@dbtproviders.com";
 $the_subject = "A DBT Provider profile mandatory update for $practiceName is due about $first_reminder_day days from
now. You can update your profile at https://dbtproviders.com/provider_profile.php";

$mailArray = array(
 'to' => $the_to,
'from' => $the_from,
'subject' => $the_subject,
'html'        => $message
);
$errors =  sendMessage($mailArray);
// delay execution by .2 sec
usleep(200000);
// reset max_execution_time to 30 sec
set_time_limit(30);
?>
<?php endif ;?>
<?php endif // end test timegone 1 or 3?>
<?php  endforeach?>
<?php foreach($accountsRecords as $record) : ?>
<?php $timeGone = 0 ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +6 month")?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])) . " +6 month")?><?php endif ?>
<?php $sixMonthsFirst = ($sixMonths - $first_reminder_day_two) ?>
<?php $sixMonthsSecond = $sixMonths - $second_reminder_day_two ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +1 year") ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])) . " +1 year") ?><?php endif ?>
<?php $twelveMonthsFirst = ($twelveMonths - $first_reminder_day_two) ?>
<?php $twelveMonthsSecond = $twelveMonths - $second_reminder_day_two ?>
<?php if ($sixMonthsFirst == $currentUnixTime ):?>
<?php $timeGone = 1 ?>
<?php $timeGone = 5 ?>
<?php endif ?>

<?php if ($sixMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 2 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($twelveMonthsFirst == $currentUnixTime ):?>
<?php $timeGone = 3 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($twelveMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 4 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if(!@$record['isAdmin']  && $common_informationRecord['reminder'] == 1):?>
<?php if($timeGone==2 || $timeGone==4):?>
<?php
  @$first_reminder_message = $common_informationRecord['first_reminder_message']; // 10 calendar days
  @$second_reminder_message = $common_informationRecord['second_reminder_message']; // 30 calendar days
   @$practiceName = $record['practice_name'] ;
  @$firstName = $record['contact_first_name'] ;
    @$lastName = $record['contact_last_name'] ;?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $expires = date("M jS, Y",
strtotime($record['test_date'])) ;?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0 ||
$common_informationRecord['send_test_emails_only'] == ''):?><?
php $expires = date("M jS, Y",
strtotime($record['updatedDate'])) ;?><?php endif ?>
<?
php // ************** START SET PROVIDER VARIABLES AND OUTPUT **************************  ?
```

```
>
<?php ob_start(); // start capturing output ?>
<?php $first_reminder_message = wordwrap( $first_reminder_message, 70); ?>
<?php echo $first_reminder_message ?>
<?php $output = ob_get_clean(); // stop capturing output  ?>
<?php ob_start(); // start capturing output ?>
<?php $second_reminder_message = wordwrap( $second_reminder_message, 70); ?>
<?php echo $second_reminder_message ?>
<?php $output2 = ob_get_clean(); // stop capturing output  ?>
<?
php //************** END SET PROVIDER VARIABLES AND OUTPUT ************************** ?
>
<?php  // **************DETERMINE PROVIDER SATUS ******************* ?>
<?php // Send Test or Real emails? ?>
<?
php if ($common_informationRecord['send_test_emails_only'] == 0 || $common_informationRecord['send_test_emails_only']
== ''):?>
<?php   $the_toa = $record['email']; // live?>
<?php elseif ($common_informationRecord['send_test_emails_only'] == 1):?>
<?php @$the_toa = $common_informationRecord['test_email'];  ?>
<?php endif ?>
<?php
 $message2 = <<<EOF
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type content='text/html; charset=utf-8" />
<style type="text/css">
.special {font-family:Arial; color:  rgb(0,0,0); font-size: 1.0em; font-weight: bold; text-
decoration: underline;}
.text_font {font-family:Arial; font-size: 1.0em;}
.text_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.0em;}
.heading_font_red {font-family:Arial; color: rgb(255,0,0); font-size: 1.3em;}
.heading_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.3em;}
.heading_font_blue {font-family:Arial; color: rgb(0,0,255); font-size: 1.3em;}
</style>
</head>
<body bgcolor="#$color">
<table style='background-color: #$color;' width='100%' align='center' border='0' >
              <tr>

    <td align='left' >
    <table style='background-color: #$color;' width='70%' align='center' border='0' >
              <tr>
    <td align='left' >
 <div align='left'>
<img src='http://www.dbtproviders.com/images/DBT_email_masthead.png' style='border:hidden;
width:489px; height:88px;'/></div>
 <div align='left' class='text_font'>
  Hello $firstName $lastName,

<span class='heading_font_blue'>This is your final courtesy reminder.</span>

  <span class='heading_font'>You must verify/update the information in your DBT Provider profile within the next 10
days, or your listing will be hidden from the DBT Provider Directory.Your information is never deleted, and you can
reinstate your listing at any time by logging in to you provider profile at
https://dbtproviders.com/provider_profile.php and updating your information. </span>
 $output2 \r\n
  Best,

 The DBT Provider Directory Team
\r\n
 </div>
  </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
  </body>
</html>
EOF;

$the_from  = "no-reply@dbtproviders.com";
 $the_subject = "A DBT Provider profile for $practiceName is due for a mandatory update about $second_reminder_day days
from now. You can update your profile at https://dbtproviders.com/provider_profile.php";

$mailArray = array(
'to' => $the_toa,
'from' => $the_from,
'subject' => $the_subject,
'html'        => $message2
);
$errors =  sendMessage($mailArray);
// delay execution by .2 sec
usleep(200000);
// reset max_execution_time to 30 sec
set_time_limit(30);
?>
<?php endif ?>
<?php endif // end test timegone 2 or 4?>
```

```
<?php endforeach ?>
<?php foreach($accountsRecords as $record) : ?>
<?php $timeGone = 0 ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +6 month") ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])) . " +6 month") ?><?php endif ?>
<?php $sixMonthsFirst = ($sixMonths - $first_reminder_day_two) ?>
<?php $sixMonthsSecond = $sixMonths - $second_reminder_day_two ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])) . " +1 year")?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0 ||
$common_informationRecord['send_test_emails_only'] == ''):?><?
php $sixMonths = strtotime(date("M jS, Y",

strtotime($record['updatedDate'])) . " +6 month") ?><?php endif ?>
<?php $twelveMonthsFirst = ($twelveMonths - $first_reminder_day_two) ?>
<?php $twelveMonthsSecond = $twelveMonths - $second_reminder_day_two ?>
<?php if ($sixMonthsFirst == $currentUnixTime ):?>
<?php $timeGone = 1 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($sixMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 2 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($twelveMonthsFirst == $currentUnixTime ):?>
<?php $timeGone = 3 ?>

<?php endif ?>
<?php if ($twelveMonthsSecond == $currentUnixTime ):?>
<?php $timeGone = 4 ?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($sixMonths == $currentUnixTime ):?>
<?php $timeGone = 5 ?>
<?php endif ?>
<?php if ($twelveMonths == $currentUnixTime ):?>
<?php $timeGone = 6 ?>
<?php endif ?>
<?php  // endif?>
<?php if(!@$record['isAdmin']  && $common_informationRecord['reminder'] == 1):?>
<?php if($timeGone==5 || $timeGone==6):?>
<?php
  @$first_reminder_message = $common_informationRecord['first_reminder_message']; // 10 calendar days
  @$second_reminder_message = $common_informationRecord['second_reminder_message']; // 30 calendar days
   @$third_reminder_message = $common_informationRecord['third_reminder_message']; // listing hidden
   @$practiceName = $record['practice_name'] ;
  @$firstName = $record['contact_first_name'] ;
    @$lastName = $record['contact_last_name'] ;?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $expires = date("M jS, Y",
strtotime($record['test_date']));?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0 ||
$common_informationRecord['send_test_emails_only'] == ''):?><?
php $expires = date("M jS, Y",
strtotime($record['updatedDate']));?><?php endif ?>
<?
php // ************** START SET PROVIDER VARIABLES AND OUTPUT **************************  ?
>
<?php ob_start(); // start capturing output ?>
<?php $first_reminder_message = wordwrap( $first_reminder_message, 70); ?>
<?php echo $first_reminder_message ?>
<?php $output = ob_get_clean(); // stop capturing output  ?>
<?php ob_start(); // start capturing output ?>
<?php $second_reminder_message = wordwrap( $second_reminder_message, 70); ?>
<?php echo $second_reminder_message ?>
<?php $output2 = ob_get_clean(); // stop capturing output  ?>
<?php ob_start(); // start capturing output ?>
<?php $third_reminder_message = wordwrap( $third_reminder_message, 70); ?>
<?php echo $third_reminder_message ?>
<?php $output3 = ob_get_clean(); // stop capturing output  ?>
<?
php //************** END SET PROVIDER VARIABLES AND OUTPUT ************************** ?
>
<?php  // **************DETERMINE PROVIDER SATUS ******************* ?>
<?php // Send Test or Real emails? ?>
<?
php if ($common_informationRecord['send_test_emails_only'] == 0 || $common_informationRecord['send_test_emails_only']
== ''):?>
<?php   $the_tob = $record['email']; // live?>
<?php elseif ($common_informationRecord['send_test_emails_only'] == 1):?>
<?php @$the_tob = $common_informationRecord['test_email'];  ?>

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

```
 $message3 = <<<EOF
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type content='text/html; charset=utf-8" />
<style type="text/css">
.special {font-family:Arial; color:  rgb(0,0,0); font-size: 1.0em; font-weight: bold; text-decoration: underline;}
.text_font {font-family:Arial; font-size: 1.0em;}
.text_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.0em;}
.heading_font_red {font-family:Arial; color: rgb(255,0,0); font-size: 1.3em;}
.heading_font {font-family:Arial; color: rgb(0,0,0); font-size: 1.3em;}
.heading_font_blue {font-family:Arial; color: rgb(0,0,255); font-size: 1.3em;}
</style>
</head>
<body bgcolor="#$color">
<table style='background-color: #$color;' width='100%' align='center' border='0' >
              <tr>
    <td align='left' >
    <table style='background-color: #$color;' width='70%' align='center' border='0' >
              <tr>
    <td align='left' >
 <div align='left'>
<img src='http://www.dbtproviders.com/images/DBT_email_masthead.png' style='border:hidden;
width:489px; height:88px;'/></div>
 <div align='left' class='text_font'>
  Hello $firstName $lastName,

<span class='heading_font_blue'>Your Directory listing has been hidden from the search results in the DBT Provider
Directory.</span>

  <span class='heading_font'>Until you verify/update the information in your DBT Provider profile your listing will
remain hidden from the DBT Provider Directory.Your information is never deleted, and you can reinstate your listing at
any time by logging in to you provider profile at https://dbtproviders.com/provider_profile.php and updating your
information.</span>
 $output3 \r\n
  Best,

 The DBT Provider Directory Team
\r\n
 </div>
  </td>
    </tr>
    </table>
    </td>
    </tr>
    </table>
  </body>
</html>
EOF;

$the_from  = "no-reply@dbtproviders.com";
  $the_subject = "The DBT Provider Directory Listing for $practiceName has been hidden from the Directory until  update
your profile at https://dbtproviders.com/provider_profile.php";

$mailArray = array(
'to' => $the_tob,
'from' => $the_from,
'subject' => $the_subject,
'html'         => $message3
);
$errors =  sendMessage($mailArray);
// delay execution by .2 sec
usleep(200000);
// reset max_execution_time to 30 sec
set_time_limit(30);
?>

<?php endif ?>
<?php endif // end test timegone 2 or 4?>
<?php endforeach ?>
<?php
    $crontext = "Reminder Run at ".date("r")."\n" ;
    $folder = substr($_SERVER['SCRIPT_FILENAME'],0,strrpos($_SERVER['SCRIPT_FILENAME'],"/")+1);
    $filename = $folder."cron_test.txt" ;
    $fp = fopen($filename,"a") or die("Open error!");
    fwrite($fp, $crontext) or die("Write error!");
    fclose($fp);
    echo "Wrote to ".$filename."\n\n" ;
    ?>
```

The actual hiding of the listings is accomplished by the following code on the search page:

```
<?php // load records from 'accounts'
  list($accountsRecords, $accountsMetaData) = getRecords(array(
    'tableName'   => 'accounts',
    'loadUploads' => true,
    'allowSearch' => true,
  ));

// load records from 'common_information'
      list($common_informationRecords, $common_informationMetaData) = getRecords(array(
    'tableName'   => 'common_information',
     'where'      => '',
     'allowSearch' => false,
    'limit'       => '1',
  ));
  $common_informationRecord = @$common_informationRecords[0]; // get first record
  ?>
<?php $currentUnixTime = strtotime(date('M jS, Y')); ?>

<?php // if(@$_REQUEST['save']):?>
<?php foreach($accountsRecords as $record) : ?>
<?php $timeGone = 0 ?>
<?php $sixMonths = '0' ?>
<?php $twelveMonths = '0' ?>
<?php if(!@$record['isAdmin'] && $common_informationRecord['reminder'] == 1): // live ?>

<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])). " +6 month" ) ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $sixMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])). " +6 month" ) ?>
<?php endif ?>
<?php if ($common_informationRecord['send_test_emails_only'] == 1 ):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['test_date'])). " +1 year" ) ?><?
php elseif ($common_informationRecord['send_test_emails_only'] == 0
|| $common_informationRecord['send_test_emails_only'] == ''):?><?
php $twelveMonths = strtotime(date("M jS, Y",
strtotime($record['updatedDate'])). " +1 year" ) ?><?php endif ?>
<?php endif ?>

<?php  if ($common_informationRecord['mandatory_update_duration']=='1'): // 12 months?
>
<?php if ($twelveMonths >= $currentUnixTime ):?>
<?php $timeGone = 2 ?>
<?php endif ?>
<?php endif ?>

<?php  if ($common_informationRecord['mandatory_update_duration']=='0' ||
$common_informationRecord['mandatory_update_duration']==''): // 6 months?>
<?php if ($sixMonths >= $currentUnixTime ):?>
<?php $timeGone = 1 ?>
<?php endif ?>
<?php endif ?>

<?
php if(!$record['isAdmin'] && !$record['notAdmin'] == '1' && ( $timeGone == 1) || ( $timeGone  == 2) &&
(!empty((int)$record['test_date'] ) || !empty((int)$record['updatedDate'] ))):?>
<?php echo htmlencode($record['practice_name']) ?>

Test Date =  <?
php echo date("M jS, Y ", strtotime($record['test_date'])) // hide and remove  for live ?>
Current Date = <?php echo date("M jS, Y")  // hide and remove  for live ?>
Time Gone = <?php echo $timeGone  // hide and remove  for live ?>
<?php  endif ?>
<?php endforeach ?>

<?php // endif ?>
```

EXPLOITSCANNER SCHEDULING USING A CRON JOB - May 30th, 2012

After some fits and starts trying to get the script to work in a shared hosting environment, here’s what I discovered...

Most shared hosting plans do not allow command line access, so scanning from the web seems like the only choice.

Most hosts time out their php execution script at 90 seconds so it's likely that your script will time out, as noted in the readme file.

However, if you can run your exploitScanner as a cron job, you may be able to run it using the cron manager command line and get around the timeout issue.

Here’s how it worked for me at IXWebHosting. (there's more info on setting up cron jobs and scheduling in the recipe CREATING A CRON JOB)

At the very top of the xs.php script, I inserted the line (NOTE: Your path will probably be different)

#!/hsphere/shared/php5/bin/php -q

to call the php interpreter

On the cron manager command line I used: (NOTE: Enter your own path information here)

```
/path_to_your_server/xs.php -p /path_to_your_server/ -l xs.log
```

To call the xs.php script, tell it to start in the site's root directory, and to create a log file called xs.log in the root directory

Don't be fooled into thinking that scanning is a quick process, my meager site took between 25 and 30 minutes to execute and then create the cron report and log file.

So, you might want to do your testing on a directory that has only a few files and change to the root after you get everything working.

The only change that I made in the xs.php file other than the php interpreter call is to change the time zone.

GOING FURTHER

Dave Edis from Interactive Tools offered: To suppress the listing of the number of scanned files in each directory when the scanner is run from a cronjob you can just comment this line:

```
 print "Files Scanned: " .number_format($OPTIONS['filesScanned']). "\r";
```

the -p flag can be followed either by the full path to the directory that you want to scan, or by a "." (Without the quotes). The "." means root directory. Don't forget to leave a space between the -p and the ".", IE: -p .

The -l flag only requires the file name of the log file after a space, IE: -l your_log_file.log

Your log file might look something like this.

```
Exploit Scanner v1.01 - Scans filesystem for web exploit patterns
--------------------------------------------------------------------------------
Matched Patterns: php(36), js(7), htaccess(6), filepath(2)
Scan started: Fri, 25 May 2012 09:35:01 -0400
Root dir: /hsphere/local/home/xyz/your_domain.com
Log file: your_log_file.log

/hsphere/local/home/xyz/your_domain.com/your_file.php (Score: 11)
  - PHP Code Match (Score 10): PHP gzinflate(base64_decode()) function pairing previously seen in exploit code (matched string: gzinflate(base64_decode($stuff2))));)
  - PHP Code Match (Score 1): PHP variable being called as a function (matched string: " . $mysql_error()

Scanned: 235 dirs, 4,308 files (1 risks found)
Execute time: 127 minutes, 19 seconds

--------------------------------------------------------------------------------
* Note: Please send any files with false-positives or undetected-exploits to us.
```

An explanation of the various entries:

```
>Matched Patterns: php(36), js(7), htaccess(6), filepath(2)
```

These are the number of patterns being checked for different kinds of files. Note that often a single pattern will match multiple exploits.

```
>Scan started: Thu, 24 May 2012 16:41:49 -0400
```

When the scan was started, useful if output is going to an email or log.

```
>Root dir: /hsphere/local/home/xyz/your_domain.com
```

The path being scanned, also useful for reference when output is going to an email or log.

If you scan from the command line you can output to a log file by adding -l yourlogfile.log in which case the filename
would be reflected here.

```
>Log file: your_log_file.log
```

```
>/hsphere/local/home/xyz/your_domain.com/your_file.php (Score: 11)
```

The path to a scanned file with possible exploit vulnerability and the total score of those vulnerabilities

```
> - PHP Code Match (Score 10): PHP gzinflate(base64_decode()) function pairing previously seen in exploit code (matched string: gzinflate(base64_decode($stuff2))));)
>  - PHP Code Match (Score 1): PHP variable being called as a function (matched string: " . $mysql_error()
```

These are the patterns that were discovered.

```
> Scanned: 235 dirs, 4,308 files (1 risks found)
 > Execute time: 127 minutes, 19 seconds
```

The number of directories and files scanned, and the time that it took for the entire scan.

WEBSITE MEMBERSHIP PLUGIN - USER EXAMPLES

SETTING UP PRIVATE AREAS FOR USERS - Aug 21st, 2011

User gsfriend said: I'm setting up a site for a custom home builder and they want a private area where customers can login and see pictures of their home during the construction process and we as a place they can access documents that relate to that customer only.

My question is, can the membership plugin be easily configured to display UNIQUE content to the current member logged in? In my case, each customer would have their own unique files that I would only want them to have access to, not other clients.

Dave at Interactive Tools came back with:

“What the plugin does is make a global available on every page called $CURRENT_USER that has the record of the currently logged in user.

So a simple implementation would be to have a section called "homes" with a field called "ownerUserNum" which was a pulldown of customer names (with user num stored as the value in the database).”

Then in your homes viewer you could have something like:

```
'where' => " ownerUserNum = {$CURRENT_USER['num']} ",
```

And only homes assigned to the currently logged in user would be shown.

Dave suggested setting up a multi record section called “Homes” with a pulldown field showing all users to the "Homes" section like this:

Go to: Admin > Section Editors > Homes > Add Field
Field Label: Owner
Field Name: ownerUserNum
Field Type: list

Display As: Pulldown
List Options: Get options from database (advanced)
Section Tablename: Accounts
Values Field: num
Labels Field: fullname

Gsfriend added fields for Title, Name, Content, File Uploads for Images, etc., to the section.

The resulting viewer code was:

```
<?php header('Content-type: text/html; charset=utf-8'); ?>
<?
php if (!@$GLOBALS['WEBSITE_MEMBERSHIP_PLUGIN']) { die("You must activate the Website Membership plugin before you can access this page."); } ?>
<?php if (!$CURRENT_USER) { websiteLogin_redirectToLogin(); } ?>

<?php
  require_once "/home/ekfriend/public_html/cms/lib/viewer_functions.php";

  list($homeownersRecords, $homeownersMetaData) = getRecords(array(
    'tableName'   => 'homeowners',
'where' => " (ownerUserNum = '{$CURRENT_USER['num']}') OR '{$CURRENT_USER['isAdmin']}' ",
    'limit'       => '1',
  ));
  $homeownersRecord = @$homeownersRecords[0]; // get first record

  list($settingsRecords, $settingsMetaData) = getRecords(array(
    'tableName'   => 'settings',
    'limit'       => '1',
  ));
```

If you wanted to also allow admins to view the records, you’d change the where statement to:

```
'where' => " (ownerUserNum = '{$CURRENT_USER['num']}') OR '{$CURRENT_USER['isAdmin']}' ",
```

According to Jason Sauchuk of Interactive Tools:

If you're searching as part of a multi-
select list, you need to use the LIKE operator and put tab characters before and
after your search option. Try this:

```
'where' => "ownerUserNum LIKE '%\t". intval($CURRENT_USER['num'])."\t%'",
```

LIST RECORDS ASSIGNED TO CURRENT USER ONLY - Aug 6th, 2010

S2media wanted to create a page where a project manager can log in to see only the projects that they're assigned to.

He went on to say, "When a project is created in a section called "Projects", I have a drop down of project managers that I can assign it to. The drop down pulls in a field "id" which is a field I added to the Accounts section. There's also a field called project_status which can be Quote, Active, Hold or Completed.

Both Jason Sauchuk and Dave Edis from Interactive Tools got involved in this one and they came up with the following:

```
list($projectsRecords, $projectsMetaData) = getRecords(array(
    'tableName'   => 'projects',
,
'where' => 'project_status="Active" AND project_manager= "'. $CURRENT_USER['id'] .'"' ,
```

While testing, you can add these to the code:

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

<?php showme($projectsRecords); ?>
```

LIST ONLY RECORDS WHERE CURRENT USER IS AUTHOR - Jul 24th, 2011

On a membership site the client wanted to display only those records that were authored by the current user from a table called "subscription_pages".

The first step was to limit the records returned to only those where the Current user was the Author with a where clause in the get records call:

```
list($subscription_pagesRecords, $subscription_pagesMetaData) = getRecords(array(
    'tableName'   => 'subscription_pages',
    'where' => 'createdByUserNum = "'. $CURRENT_USER['num'] .'"' ,



      ));
```

Then for an added touch, I welcomed the user by name and added their subscription level to the greeting.

```
Welcome <?php echo mysql_escape($CURRENT_USER['first_name']); ?> <?
php echo mysql_escape($CURRENT_USER['last_name']);
?>, <?php if ($CURRENT_USER['allowedFolders'] == "entry_level"): ?>Entry<?
PHP endif ?><?php if
($CURRENT_USER['allowedFolders'] == "silver"): ?>Silver<?PHP endif ?><?
php if ($CURRENT_USER['allowedFolders'] ==
"gold"): ?>Gold<?PHP endif ?><?
php if ($CURRENT_USER['allowedFolders'] == "platinum"): ?>Platinum<?PHP endif ?
> level
subscriber.
```

Then where I wanted to display the user's choices:

```
 <?php foreach ($subscription_pagesRecords as $record): ?>
   <?php if ($record['createdByUserNum']== @$CURRENT_USER['num']): ?>

<!-- Records Created By User #:  <?php echo $record['createdByUserNum'] ?
> Exist -  Current User is # : <?php echo
@$CURRENT_USER['num'] ?><br />-->

 <span class="medium">TO VIEW ALL OF YOUR LISTINGS - </span>
<a href="master_listing.php"><span
class="medium"><u><b>CLICK HERE</b></u></span></a>
   <br /> <br />
     <span class="medium">TO ADD LISTINGS OR MODIFY EXISTING LISTINGS - </span>
```

```
<a href="cmsAdmin/admin.php"><span
class="medium"><u><b>CLICK HERE</b></u></span></a>
  <br /> <br />
   <?php endif ?><?php break ; ?><?php endforeach ?>
```

And if the current user had not created any records:

```
 <?php if (!$subscription_pagesRecords): ?>
    <span class="medium">YOU HAVEN'T CREATED ANY LISTINGS YET. </span><a
href="http://96.0.19.10/cmsAdmin/admin.php"><span class="medium"><u>
<b>CLICK HERE</b></u></span></a>
    <span class="medium">TO GET STARTED.</span>
<br /> <br />
<?php endif ?>
```

LIST RECORDS WHERE AUTHOR OF RECORDS IN TABLE 1 RECORDS MATCH AUTHOR OF RECORDS IN TABLE 2 - Jul 24th, 2011

If a member-user uploaded a custom masthead, I needed to display that masthead in a detail viewer. However, I wanted to display each user's masthead only on records that they had created.

Here's how...

I already had a multi-record editor called subscription_pages that contained all of the records created by each member-user.

In order to add custom mastheads:

1) Create a multi-record editor called "custom_branding" with only one record allowed per user. The only field in this editor is an upload field for their custom masthead.

2) To the standard records calls:

```
 list($subscription_pagesRecords, $subscription_pagesMetaData) = getRecords(array(
    'tableName'   => 'subscription_pages',
    'where'       => whereRecordNumberInUrl(1),
    'limit'       => '1',

  ));
  $subscription_pagesRecord = @$subscription_pagesRecords[0]; // get first record
```

Add the following to define the variable $owner (add this below the subscription_pages call or you'll get a undefined variable error)

```
 $owner = $master_subscription_pagesRecord['createdByUserNum'];
```

Then change the where statement to restrict the records called to only those where the author's of the custom branding record (the masthead) matches the subscription_pages records

```
 // load records
  list($custom_brandingRecords, $custom_brandingMetaData) = getRecords(array(
    'tableName'   => 'custom_branding',
'where' => 'createdByUserNum = "'. $owner .'"' ,
    'limit'       => '1',
  ));
  $custom_brandingRecord = @$custom_brandingRecords[0]; // get first record
```

Finally in the body where you want to display the custom masthead:

```
<?
php if ($custom_brandingRecord['masthead'] && ($custom_brandingRecord['createdByUserNum'] ==
$master_subscription_pagesRecord['createdByUserNum'])) : ?><?
php foreach ($custom_brandingRecord['masthead'] as
$upload): ?><img src="<?php echo $upload['thumbUrlPath'] ?>" width="<?
php echo $upload['thumbWidth'] ?>" height="<?php
echo $upload['thumbHeight'] ?>" alt="" /><?php endforeach ?>
<?php else : ?>
<img src="your_standard_masthead.jpg" />
<?PHP endif ?>
```

For testing purposes, you can add:

```
Custom Branding owner: <?php echo $custom_brandingRecord['createdByUserNum'] ?
> <br /> Subscription Page owner: <?php
echo $master_subscription_pagesRecord['createdByUserNum'] ?>
```

CHECKING FOR PASSWORD STRENGTH - Aug 3rd, 2013

Thanks to user Zick and Greg Thomas from Interactive Tools for the following Password Strength checking suggestions for the Website Membership Plugin.

The code should be inserted after the NewPassword Checking code. You can change the password length to any number of characters that you feel is appropriate.

For uniformity, you might also want to change the initial password generated in the “generate password” section of the user-
signup.php form to that number of characters matches. (See the recipe called CHANGING PASSWORD LENGTH)

```
    if    (strlen($_REQUEST['newPassword1']) < 11 )                              { $errorsAndAlerts .=
"Please enter 12 or more characters for your New Password\n"; }
    if    (!preg_match("#[a-
z]+#", $_REQUEST['newPassword1']) )                        { $errorsAndAlerts .=
"Password must include at least one letter!\n"; }
    if    (!preg_match("#[A-
Z]+#", $_REQUEST['newPassword1']) )                        { $errorsAndAlerts .=
"Password must include at least one CAPS!\n"; }
    if    (!preg_match("#\W+#", $_REQUEST['newPassword1']) )                       { $errorsAndAlerts .=
"Password must include at least one symbol!\n"; }
    $errorsAndAlerts .= getNewPasswordErrors(@$_REQUEST['newPassword1']); // only works in Version 2.52+)
_
```

According to Greg, the getNewPasswordErrors function added in Version 2.52, and implemented in Version 1.10 of the Membership Plugin, will return a string of errors, including checking if the password starts with spaces, and if the password is in the 1000 most popular passwords. If there are no errors it will return an empty string.

If you’re using a prior version just comment out (or remove) the getNewPasswordErrors code.

HINT: Check the file /lib/login_password_blacklist.txt

If you've got some passwords that you think should be included, just add them to the file and they won't be accepted.

CHANGING PASSWORD LENGTH - Aug 27th, 2014

When a new user signs up for an account, a randomly generated 17 character temporary password is sent to them. That's pretty secure, but if it's a bit unwieldy for your situation, you can easily change the password length to any amount of characters.

In llder versions of the Website Membership plugin, in the user-signup.php file, look for the code:

```
$_REQUEST['password'] = substr(md5(uniqid(rand(), true)), 15);
```

or in newer versions Look in the Website Membership plugin itself, for:

```
$password = substr(md5(uniqid(mt_rand(), true)), 15);
```

According to Chris Waddell of Interactive Tools "The 15 in the code skips the first 15 characters of the 32 characters returned by md5(), resulting in a 17 character string."

To set the length to 5 characters, try this:

```
$_REQUEST['password'] = substr(md5(uniqid(rand(), true)), 0, 27); // example output: c5560
```

or in newer versions

```
$password = substr(md5(uniqid(mt_rand(), true)), 0, 27); // example output: c5560
```

You can replace the 5 with any number you'd like up to 32.

QR CODES

QRCODEGENERATOR PLUGIN - Dec 29th, 2018

Now for something completely different, and useful...

Robin Brayer of Interactive Tools has created a free QRCodeGenerator plugin that creates QR (Quick Response) codes from URLs, text, Contact Information, email addresses and more.

You can download the latest version from the user submitted page at:

http://www.interactivetools.com/add-ons/?user_submitted=1

For those of you that are not up to speed on the many uses of QR codes, you can find out more on the CMSB Forum at:

http://www.interactivetools.com/iforum/Products_C2/CMS_Builder%3A_Plugins_%26_Add-ons_F40/QR_Code_Generator_P87895/

Or by Googling QR Codes

Implementation is really simple:

Just upload the QRCodeGenerator plugin to your cmsAdmin/plugins folder and then activate it through the admin>plugins area if the admin interface.

There are examples included that show how to create QR codes from many types of information.

You can read more about QR Codes at:

http://code.google.com/apis/chart/docs/gallery/qr_codes.html

http://code.google.com/apis/chart/docs/gallery/qr_codes.html

and an interesting blog at:

http://www.socialmediaexaminer.com/how-qr-codes-can-grow-your-business/

CREATING QR CODES FOR ALL THE RECORDS IN A SECTION - May 5th, 2011

I had a created a multi-record section with a record for each of an artist's client's works.

The artist wanted to be able to generate a single page, three column viewer that showed the QR codes for the URL of the detail page for each record in that section, along with some identifying information.

The QR images would then be used to create QR stickers for wall labels at an art exhibition.

The records all had an upload field called "image" and a "Title" field for the name of the work.

Here's the code I used for the top of my viewer:

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

  list($your_tablesRecords, $your_tableMetaData) = getRecords(array(
    'tableName'   => 'your_table',
  ));


?>
<!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 here's the code for the body:

```
<div align="center"><h2>HOME PAGE QR CODE</h2><?php
  echo qrCode(array(
    'type' => 'url',
    'size'   => '150', // height & width in pixels
    'url'  => 'http://your_site.com/'
  ));
?></div>
<br /> <br />
<div align="center"><h2>INDIVIDUAL IMAGE QR CODES<br /> <br />
<table align="center" width="1200">
  <tr><?php foreach ($your_tableRecords as $record): ?>
 <?php foreach ($record['image'] as $upload): ?>
 <?php $num = $record['num'] ?>

    <td align="center" valign="bottom"><img src="<?php echo $upload['thumbUrlPath'] ?
>" width="<?php echo
$upload['thumbWidth'] ?>" height="<?php echo $upload['thumbHeight']?
>" />&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<?php
  echo qrCode(array(
    'type' => 'url',
    'size'   => '150', // height & width in pixels
    'url'  => "http:/your_site.com/your_detail_page.php?$num"
  ));
?><br /><?php echo $record['title'] ?><br /><hr /><br />
</td>
<?php $maxCols=3; if (@++$count % $maxCols == 0): ?></tr><tr><?php endif; ?>
<?php endforeach ?>
<?php endforeach ?>
</tr>
</table>
```

Here's a trick from Robin Brayer that helped me get this to work.

To append the record number to the URL you first need to create a variable.

I used:

```
<?php $num = $record['num'] ?>
```

Then you can append that variable to the URL in two ways.

1) Then you can concatenate the variable to the URL like this

```
'url'  => 'http://www.elleschorrphotography.com/gallery5detail.php?' . $num
```

or,

2) If you use double quotes instead of single quotes, you can add the variable directly to the string, like this:

```
'url'  => "http://www.elleschorrphotography.com/gallery5detail.php?$num"
```

BUGS AND FIXES

DON'T LIKE "SORRY, NO RECORDS WERE FOUND!" TO APPEAR AFTER CLEARING THE DEVELOPER, LOG? WELL, YOU'RE NOT ALONE. - Nov 11th, 2022

Unless you're Canadian, the word 'Sorry. probably appears strange when used in some cases.

For me it was that when I cleared the developer log table, the message "Sorry, no records were found!" came up instead of something like, "Yay, no records were found!"

According to Dave Edis, Senior Developer at interactivetools.com: It's funny, saying "Sorry" is a bit of a Canadian cultural stereotype. Even the transit buses say "Sorry" when they're full or not in service!
https://dailyhive.com/vancouver/canadian-buses-apologize-sorry-photos

However, if you edit /lib/languages/en.php you can provide a different translation even for the English version. Just search for "Sorry":

```
    'Sorry, no records were found!' => 'Sorry, no records were found!',
    'Sorry, the <b>search</b> returned no results!' => 'Sorry, the <b>search</b> returned no results!',
```

Just replace the second string with whatever text you want."

He was, "Sorry about that, ;-)"

PATCHING CORE PROGRAM FILE BREAKS CMSB ADMIN PAGE FORMATTING - Jul 9th, 2020

I had to patch the code in a core program file (lib/common.php) for some specific functionality, but no matter what I tried I came up with a trashed Admin page formatting.

What was even more strange is that when I undid the code changes, and re-uploaded what should have been the unmodified file, the badly formatted admin pages remained.

Both Greg Thomas from Interactive Tools and I were at a loss to explain the issue until CMSB user Steve99 came up with the solution.

He said: "In reference to "but the page is very strangely formatted"... I've seen that happen before when patching a file. It ended up being the DreamWeaver html editor that corrupted the file.

Try replacing the modified file with the original packaged one, then perform the edits in another editor program such as Notepad++ or Sublime Text and upload.

I was originally using DreamWeaver and when I used Notepad++ instead, the page formatting remained correct.

Now that I think about it, Dreamweaver also sometimes caused 'cannot send header' errors by inserting a hidden blank character before the opening PHP tag.

Moral... Use another text editor in these rare situations.

FILE UPLOAD ISSUES WITH THE NEW HTML UPLOADER AND FILE SIZE ISSUES IN

V3.50 AND OTHER VERSIONS - Dec 16th, 2019

File Upload issues with the html uploader in V3.50 and possibly other versions

Issue 1) (from version 3.15 - Not tested in 3.50) .mov files could not be uploaded without throwing a 'forbidden file type' error.

Daniel Loewe, a programmer from Interactive Tools said that was a related to an ongoing issue that we are aware of and currently looking at solutions for. He suggested adding the file type quicktime (the whole word) as an allowed filetype. That solved the error and I could upload a small test .mov file.

Issue 2) (from version 3.50) Attempts to upload large files (300mb) of any kind would throw an error similar to '414 Request-URI Too Large" error from nginx/1.14.1.1'. or 'The requested URL's length exceeds the capacity limit for this server'. Initially, I thought this to be related to the upload_max_filesize and post_max_size settings in my php.ini file(s). (I actually found 2 php.ini files, a global one in the site root directory and the other in my cmsAdmin directory. (You may have to ask your webhost to change the values in the Global php.ini file for your site)

I changed both the values in both files to 999M (999Mb) and thought that might solve the issue.

When I checked the values in the Admin>General tab they still showed an upload_max_filesize of 102M, and when I checked the php.ini, using the link in the Admin>General tab under Server Info, the values were 102M local and 999M global.

Still, trying to upload a large (300mb) file threw the nginx error.

Again Daniel came to the rescue.

He said:

"That nginx error is reported when a URL is requested that is too long for the server's configuration. In this case, this appears to be a bug in how the HTML5 uploader is handling some errors. That's something that we'll have to look into as a secondary issue, but hopefully sorting out your configuration problems will also clear this error up.

For PHP configuration, CMSB comes with a number of different default configuration files, as different server setups will check different files/locations. These files have different values set to help locate which one is being loaded. 102MB indicates that it's checking the .user.ini file in your cmsAdmin directory for it's values. Try changing the values there."

I change both of the values to 999M in the cmsAdmin>.user.ini file, and now large uploads are working as expected with no errors.

Hope this helps someone.

HEADER BAR AND HTML SEPARATOR STYLING FIX (PRIOR TO V3.07) - Oct 24th, 2016

From CMSB version 3.0 through 3.06 styling in Header Bars has shown as html instead of actually styling the Header Bar.

Dave Edis, Senior Programmer at InteractiveTools has offered a simple fix until Version 3.07 is released.

He said:

In /lib/admin_functions.php search for "to display even" and replace this line:

```
$attrs['label'] = $attrs['label'] !== "" ? htmlencode($attrs['label']): '&nbsp;'; // default to &nbsp; to display even
with empty string
```

With this line:

```
$attrs['label'] = $attrs['label'] !== "" ? $attrs['label'] : '&nbsp;'; // default to &nbsp; to display even with empty
string
```

In /lib/menus/database/editTable_functions.php

To display up to 50 characters of text in an html separator in the field list instead of just --html--

In /lib/menus/database/editTable_functions.php

replace this block:

```
if (@$field['type'] == 'separator') {
  if ( @$field['separatorHeader']) { echo "<div class='text-center' style='line-height: 200%'>
<b>---
{$field['separatorHeader']} ---</b></div>\n"; }
  else                              { echo "<div class='text-center'>--- {$field['separatorType']} ---
</div>"; }
}
```

With this block:

```
if (@$field['type'] == 'separator') {
  $htmlAsText = htmlencode(mb_strimwidth(strip_tags($field['separatorHTML']), 0, 50, "..."));
  if     ($field['separatorType'] == 'header bar') { echo "<div class='text-center' style='line-
height: 200%'><b>---
{$field['separatorHeader']} ---</b></div>\n"; }
  elseif ($field['separatorType'] == 'html')       { echo "<div class='text-center' style='line-
height: 200%'><b>---
$htmlAsText ---</b></div>\n"; }
  else                                             { echo "<div class='text-center'>--
- {$field['separatorType']}
---</div>"; }
}
```

Dave said: We'll include that in V 3.07 and also enable the "Field Label" field so you can manually specify a field label to be displayed in the field editor as well if desired.

CMSB SOMETIMES LOGS ME IN AUTOMATICALLY - Aug 6th, 2010

It's not a bug. According to Dave Edis: Starting with about version 1.32, “We've switch the way login credentials are stored so they are no longer removed when the browser is closed and are stored for up to 24 hours. It used to be stored until the browser was closed.”

STRANGE ERROR MESSAGES - Aug 6th, 2010

Ever consistently get a:

Warning: preg_match(): Unknown modifier '\' in /yourserverpath/cmsAdmin/lib/menus/default/save.php on line 132

when you try to save a record? Well, that’s a bug that has been fixed in the latest CMSB v1.34 and above.

It occurs when a non-escaped /appears in one of your “disallowed characters” fields. The / is ending that regexp early and confusing it. (Search for regex in the cookbook to learn more about regular expressions).

If you don’t want to upgrade right now, you can fix the problem by backslashing the / to let the regex expression know it's meant to be a character. As in:

```
!@#$%^&*()=_":;<>?\/\~`-
```

DEFAULT VALUE TEXT NOT COPIED TO RECORDS - Aug 6th, 2010

This was a bug in the first release version of CMSB1.34 which has been fixed.

GETTING ERRORS AFTER RESTORING A DATABASE? - Dec 31st, 2012

Dave Edis from Interactive Tools offered this possible fix. He said:

"Generally you need everything to match up, so the /data/schema/ folder (or the fields under the field editor) should be the same as when the backup was made).

If you are getting errors because there is a field in the schema (created in the field editor) that doesn't exist in the MySQL Database you can just click on "Section Editors" and it will automatically check and re-create any fields that are missing from the MySQL."

CHAPTER 6 - SUBSCRIBER CONTRIBUTED RECIPES

CREATE A 10 STAR RATING SYSTEM IN CMSB - Dec 29th, 2018

It’s become quite common for hotels, attractions and other listings to include both reviews and star rating systems in their listings.

Here’s how to create a star rating system for your site, using CMSB. It’s here courtesy of CMSB user thenetgirl, who created it for a rental property client.

EDITORS
First, you’ll need to create 2 multi-record editors. They’re called “Vacation Rentals” and “Ratings”.

* You can create preformatted editors by upload the ,ini.php files in the download package below to your cmsAdmin/data/schemaPresets folder and then creating the editors from the "select presets" pulldown menu

The Vacation Rentals editor contains the listing information records for the individual properties.

(Each record contains listing information for a single property. The “Property Code” field value in the “Vacation Rentals” editor is used in a pull down list to populate the “Property Code” field in the “Ratings” editor.)

The Ratings editor contains the customer submitted ratings for the individual properties.
(There can be multiple “Ratings” records for the same property. The “votes” are averaged and displayed in the appropriate viewers. The VacationRentalsDetail viewer shows only the overall rating for the property and the MoreRatings viewer shows detailed ratings for that property.

VIEWERS
There are 3 viewers that you’ll need to create: VacationRentalsList.php, VacationRentalsDetail.php, and MoreRatings.php.

The Vacation Rentals viewers are standard list and detail pages for the property listings. Code to display the "Overall Rating" and a link to the MoreRatings detailed ratings viewer are added to the VacationRentalsDetail page

The MoreRatings viewer is the one that shows the average of the ratings submitted for the particular property. Code to display the detailed ratings is added to this viewer.

IMAGES
You’ll also have to create the “star” images to be displayed as the ratings. I’ve created some simple circle gifs with the numbers 1-9 for this demonstration. (They are hard coded to be found in an images/stars folder in the directory that contains your viewer files, so if you change the location, don’t forget to change the viewer coding.)

Rather the listing all the code here, you can download the ini.php* files the sample viewer files, and a folder containing the 9 sample star gifs from:

http://thecmsbcookbook.com/downloads/ratings.zip

** DON’T FORGET TO CHANGE THE "LOAD VIEWER LIBRARY - $dirsToCheck" PATH IN THE SAMPLE VIEWERS TO YOUR OWN SERVER PATH.

Before you can see the rating system in operation, you’ll need to create some “property listings” records in the Vacation Rentals editor and then some Ratings records for those properties in the Ratings editor.

Once you’ve created a working system, you can customize the components for your specific needs.

You can also use the membership plugin to allow previously registered guests to post their own ratings and comments.

AN EVENTS CALENDAR USING CMSB - Sep 11th, 2020

Small change October 11, 2018

After testing with PHP7 and CMSB 3.15, I had to move the header line <?php header('Content-

type: text/html;
charset=utf-8'); ?> in caledar5.php to after the 'load viewer library' call, or I'd get the errors:

```
E_USER_ERROR: removePrefixedCookie: Can't remove cookie(loginsession), headers already sent! Output started in
/home3/ellescho/public_html/thecmsbcookbook/calendar5.php line 1.

/home3/ellescho/public_html/thecmsbcookbook/cmsAdmin/lib/common.php (line 757)
http://www.thecmsbcookbook.com/calendar5.php
```

AND

```
E_WARNING: session_start(): Cannot send session cache limiter - headers already sent (output started at
/home3/ellescho/public_html/thecmsbcookbook/calendar5.php:1)

/home3/ellescho/public_html/thecmsbcookbook/cmsAdmin/lib/init.php (line 713)
http://www.thecmsbcookbook.com/calendar5.php
```

This recipe has been updated to reflect the latest code changes as of 2/2/2011
The zip file was recompiled on April 9, 2011
If you were having difficulties with single events, please re-download the zip file.

Zickey, who bills himself as “just a guy who uses CMS Builder” put together this really cool calendar application and has been kind enough shared both the code and instructions with us. He asks that if you utilize the Zickey calendar, you read and agree to the terms in the terms.txt document, which basically say that if you use the calendar, don’t blame him for anything that doesn’t work the way you expect it to.

To implement the Zickey calendar, you’ll need:

A multi record editor called “events”
A viewer called “calendar5.php”
A detail viewer called “events”detail.php”
The “calendar5.css” file

You can download all the required viewer files, the.css file, the events.ini.php file*, the terms.txt file, and a copy of Zickey’s help file, instructions.rtf from:

http://www.thecmsbcookbook.com/downloads/zickey_calendar5.zip

You can see an operating calendar example at

http://www.thecmsbcookbook.com/calendar5.php

The calendar contains sample events spanning September and October 2020.

***To make the calendar load the event entries from your site, the calendar5.php $dirsToCheck = array code on line 5 must be updated with your own path. You'll also need to modify the eventsDetail.php file the same way. If needed, you can find the path by opening the events editor in the CMS Builder "Code Generator".

Note: The eventsDetail.php viewer is for example only and taken directly from the CMSB code generator.

*** to create your own events editor, upload the events,ini.php file to your cmsAdmin/data/schemaPresets folder and then create the editor from the "select presets" pulldown menu
.
Setup is extremely simple:

1) create the “events” editor and create some test events.
2) Change the “l $dirsToCheck = array” paths and upload the calendar.php, and eventsDetail.php viewers to your server
3) Upload the calendar5.css file to your css directory

That’s it, the calendar should be fully functional.

The calendar uses a “maxwords” function to limit the amount of words shown on the calendar in the optional event description field. If you’re unfamiliar with the maxwords function you can learn more about it in the recipe, “LIMITING THE NUMBER OF WORDS SHOWN IN A PAGE”.

**** The Zickey calendar is a collaborative project, so if you make any useful changes, or add any functionality, please post your changes and additions to:

http://www.interactivetools.com/iforum/Products_C2/CMS_Builder_F35/P82903

As always, after you’ve got the example working, you can add any required fields and style the pages the way you’d like

1.21 seconds