
Bs_ImageArchive:

make your objects editable

Andrej Arn
Sam Blume



blushoes

THIS IS A STEP-BY-STEP GUIDE TO CREATE AN EXAMPLE IMAGE ARCHIVE USING THE BS_IMAGEARCHIVE PLUGIN.

What you need:

- An existing installation of PHP/MySQL/Apache (or IIS) and BlueShoes.
- A website running BlueShoes, with an existing MySQL database.
- Some knowledge about PHP, understanding of how to use existing classes. Some knowledge of how to use SQL/MySQL.

Source Location:

Class: applications/imagearchive/Bs_ImageArchive.class.php

Example: applications/imagearchive/example/index.php (PHP code)

Example: applications/imagearchive/example/create.sql (SQL code)

Version 1.1, updated 2002-??

Create a new subdirectory in your webroot, in my case it's /examples/ImageArchive/.

Copy the file /applications/imagearchive/example/index.php into your new directory. If your webserver is set to read something else as default directory file, rename it (default.php or home.php, whatever).

We'll look at the source code shortly.

Create a subdirectory called "images", in my case the path is now /examples/imagearchive/images/. That directory will be your home for all images.

Now you may want to order your images in subfolders. In my case I created the following structure, do what you please.

```
images
  sam
    holidays
      usa1999
      austria2002
  rick
    biking
    cats
    private
```

We will use sessions and a CUG¹ (closed user group) to prevent access by unauthorized users.

Go to your MySQL prompt (or use phpMyAdmin²) and create the following table in your existing BlueShoes database:

```
CREATE TABLE `ExampleCugImageArchive` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `user` varchar(20) NOT NULL default '',
  `pass` varchar(20) NOT NULL default '',
  `isActive` tinyint(4) NOT NULL default '0',
  `startDatetime` datetime NOT NULL default "0000-00-00 00:00:00",
  `endDatetime` datetime NOT NULL default "0000-00-00 00:00:00",
  PRIMARY KEY (`ID`),
  KEY `user` (`user`)
) TYPE=MyISAM;
```

Feel free to rename the table to something else. Just make sure you remember that, cause we're going to use that name in the future.

Now let's add the users. In my case, I do:

```
INSERT INTO ExampleCugImageArchive (user, pass, isActive) VALUES ("sam",
"password", 1);
INSERT INTO ExampleCugImageArchive (user, pass, isActive) VALUES ("rick",
"password", 1);
INSERT INTO ExampleCugImageArchive (user, pass, isActive) VALUES
("visitor", "pictures" , 1);
```

now my table looks like:

```
mysql> select * from ExampleCugImageArchive;
```

ID	user	pass	isActive	startDatetime	endDatetime
1	sam	password	1	0000-00-00 00:00:00	0000-00-00 00:00:00
2	rick	password	1	0000-00-00 00:00:00	0000-00-00 00:00:00
3	visitor	pictures	1	0000-00-00 00:00:00	0000-00-00 00:00:00

3 rows in set (0.00 sec)

Let's protect the directories now.

Me, I create a file called .htBsFaAccess in the directories /images/rick/ and /images/sam/. The content of them is:

```
[access]
view    = login
add     = u:rick
edit    = u:rick
delete  = u:rick
```

(or 'sam' instead of 'rick' in the later.)

That means: everyone with a valid login may view the pictures, but only rick may do admin stuff. These settings are valid for this directory and all directories below that one (that don't overwrite the settings with their own .htBsFaAccess file).

This works like apache with its .htaccess files. We call them that way because apache has a default setting to not show files that start with ".ht".

If you're a windows user and the file explorer does not let you create files starting with a dot, just create them without the dot and then go to your command prompt (start – run – cmd – enter) and rename that file (ren htBsFaAccess .htBsFaAccess). That works.

Rick has a folder called "private". He does not want anyone to see the pics in there. So I create another .htBsFaAccess file, but this time I set 'view' to 'u:rick' as well.

Sam has a folder called 'open', he wants everyone to upload. So I set everything to 'login'.

Now let's look at the index.php file.

Lines 1-4

```
<?php
require_once($_SERVER["DOCUMENT_ROOT"] . "/global.conf.php");
require_once($APP['path']['core'] . 'auth/cug/Bs_CugDb.class.php');
require_once($APP['path']['applications'] . 'imagearchive/Bs_ImageArchive.class.php');
```

This includes the needed dependent BlueShoes files. It's the global configuration file, the Bs_CugDb class for user authentication and session handling, and the ImageArchive plugin itself.

Lines 6-11

```
$cug =& new Bs_CugDb('imageArchive');
$cug->userTableName = 'ExampleCugImageArchive';
$cug->logTableName = 'ExampleCugImageArchiveLoginLog';
$cug->redirectPage = 'index.php';
$cug->language = 'de';
$cug->formTarget = '_self';
```

Create a new instance and set the object vars. If you named your table different than “ExampleCugImageArchive”, change that. It’s wise to call the “log” table the same with the suffix “LoginLog”. This log table will log all (failed and successful) logins. The redirectPage is the page to link to once the user logged in successfully. If your page is not called index.php, change that here. Me, I set the language to German. You may use English “en” as well.

Lines 12-26

```
if (!$cug->letMeIn()) {
    if (!is_object($sDb)) {
        include_once($APP['path']['core'] . 'db/Bs_MySql.class.php');
        $sDb =& new Bs_MySql();
        $connId = $sDb->connect($APP['db']['main']);
        if ($sDb->isException($connId)) {
            //crap.
            $connId->stackTrace('', __FILE__, __LINE__, 'fatal');
            $connId->stackDump('alert');
            die('check your db, honey!');
        } else {
            $GLOBALS['bsDb'] = &$sDb;
        }
        $cug->setDbObject($sDb);
    }
}
```

The CUG class needs a connection to the db. If your global.conf.php file did not create one, it is created here.

Lines 27-38

```
$status = $cug->treatLoginForm();
if ($status === TRUE) {
    //oki doki, register some vars.
    $t = $cug->form->getFieldValue('username');
    $username = $t[0];
    $cug->bsSession->register('CugImageArchiveUser', $username);
} else {
    //display login form
    echo $status;
    exit;
}
}
```

Once the login was successful, we register the session var ‘CugImageArchiveUser’ and assign it the username (‘rick’, ‘sam’ or ‘visitor’).

Lines 41-57

```
$iArch =& new Bs_ImageArchive();
$iArch->serverBasePath = $APP['path']['site/www'] . 'examples/ImageArchive/images/';
$iArch->webBasePath = '/examples/ImageArchive/images/';
$iArch->thumbSettings = array(
    'prefix' => 'thumb_',
    'width' => 60,
    'height' => 60,
);
$iArch->overviewSettings = array(
    'tdWidth' => 65,
    'tdHeight' => 65,
    'tdColor' => 'silver',
    'cols' => 7,
    'rows' => 4,
);
$iArch->sessionUser = $bsSession->getVar('CugImageArchiveUser');
$iArch->currentDirectory = (isset($_REQUEST['currentDirectory'])) ?
$_REQUEST['currentDirectory'] : '';
```

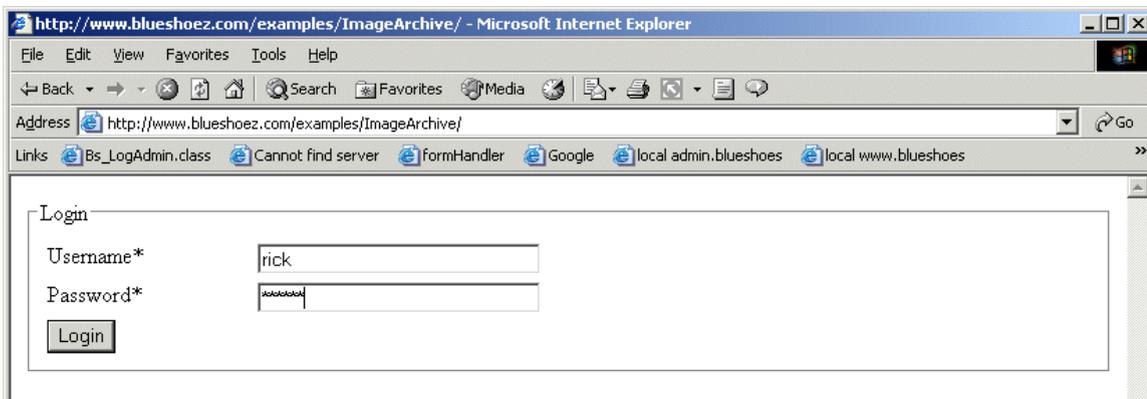
Set the serverBasePath and webBasePath for your server.

When pictures are uploaded (jpg) then thumbnails get created on the fly. The file names are prefixed with 'thumb_' and have the given size. Feel free to change that.

in the overview page there's an html table. Cells have the size of tdWidth and tdHeight. It should be a bit bigger than what you set for the thumbnails. In my case i have 7 pictures per row (7 cols) and 4 rows per page.

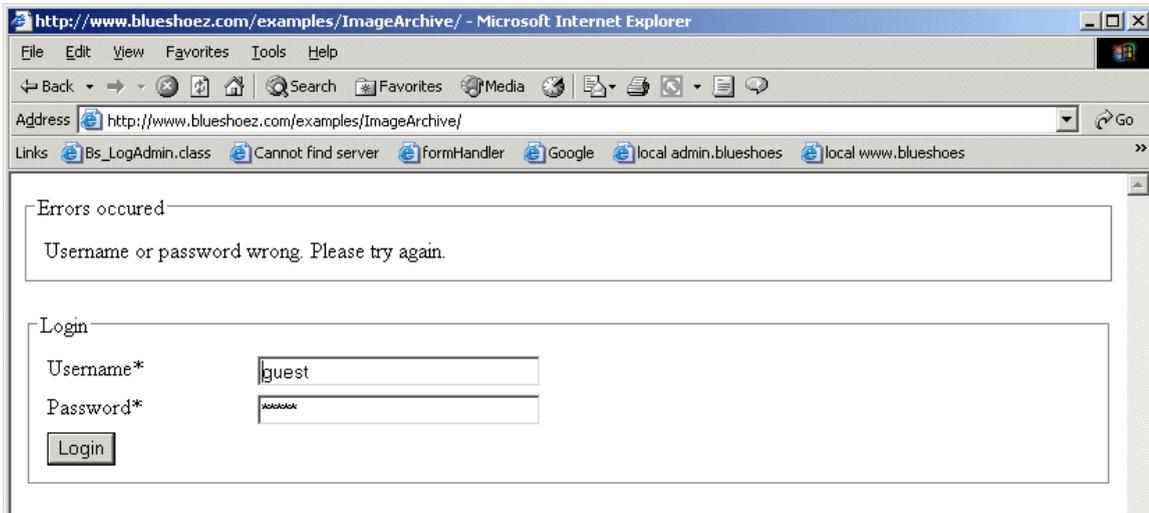
Lines 60 up to the end are mostly html. There's a css file (main.css) and an image (titleWebcamart.gif) which you can also find where you got index.php from. Just copy them over if you want to.

Now we're done, let's see if that works. Navigate your browser at your url, in my case that's <http://www.blueshoes.org/examples/ImageArchive/>

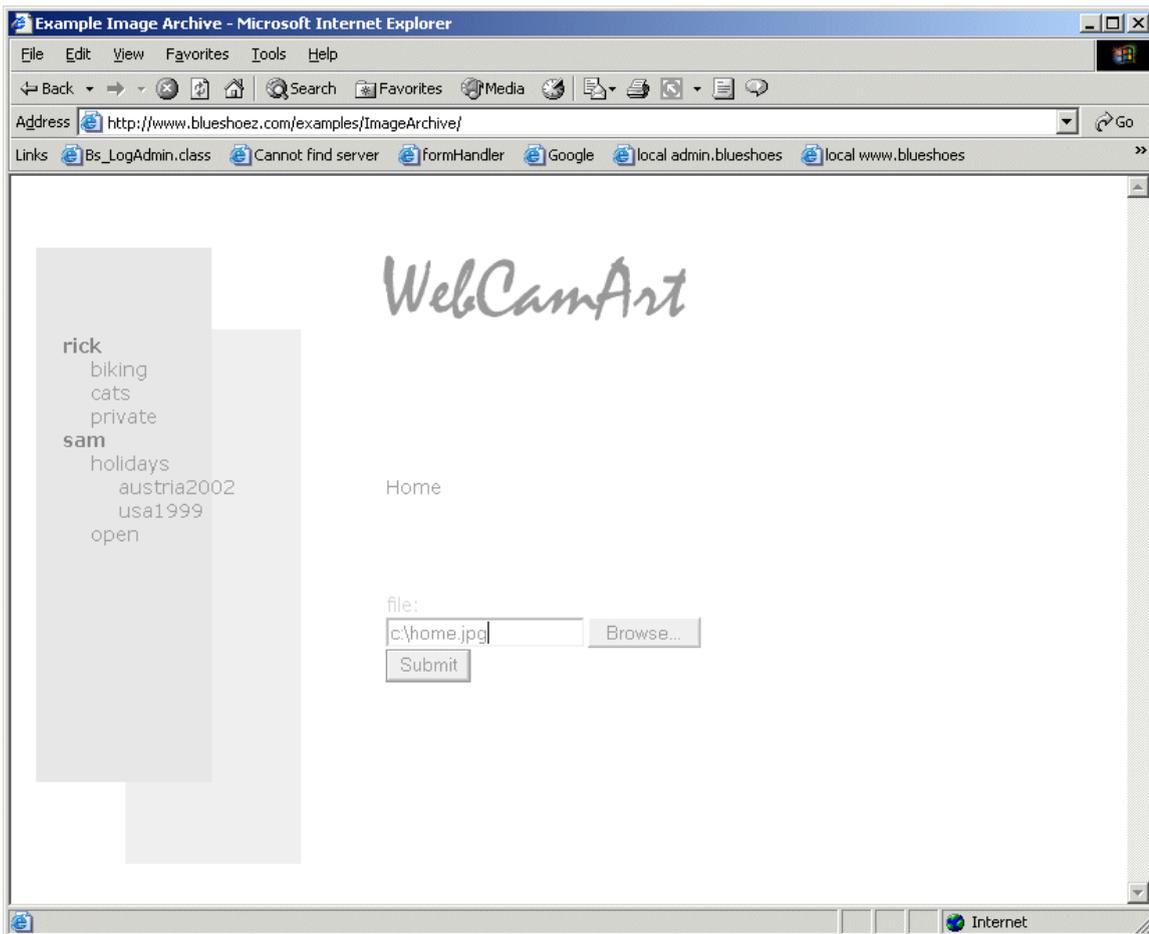


If you don't then check your path settings etc. Log in using your favorite user.

If your login failed, you'll see this one:

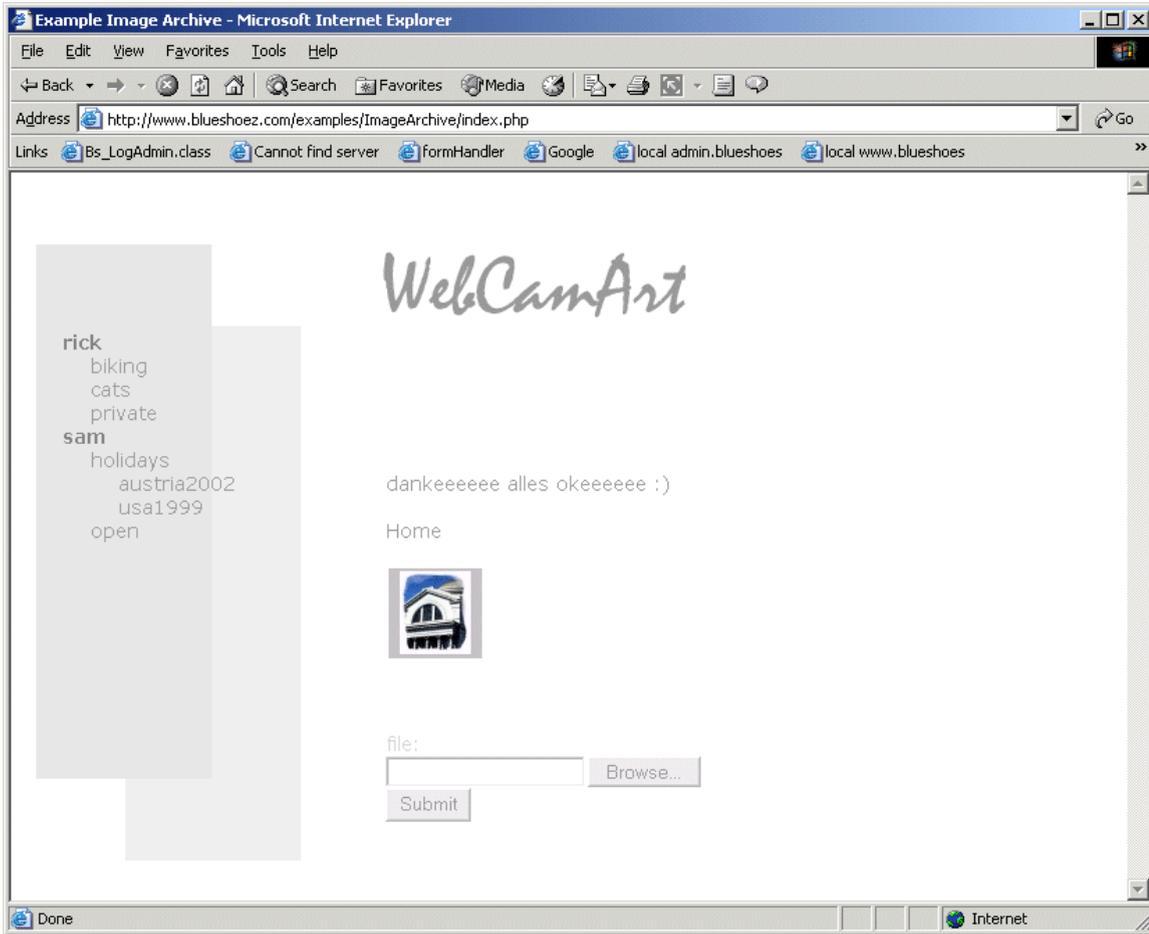


Otherwise this one:

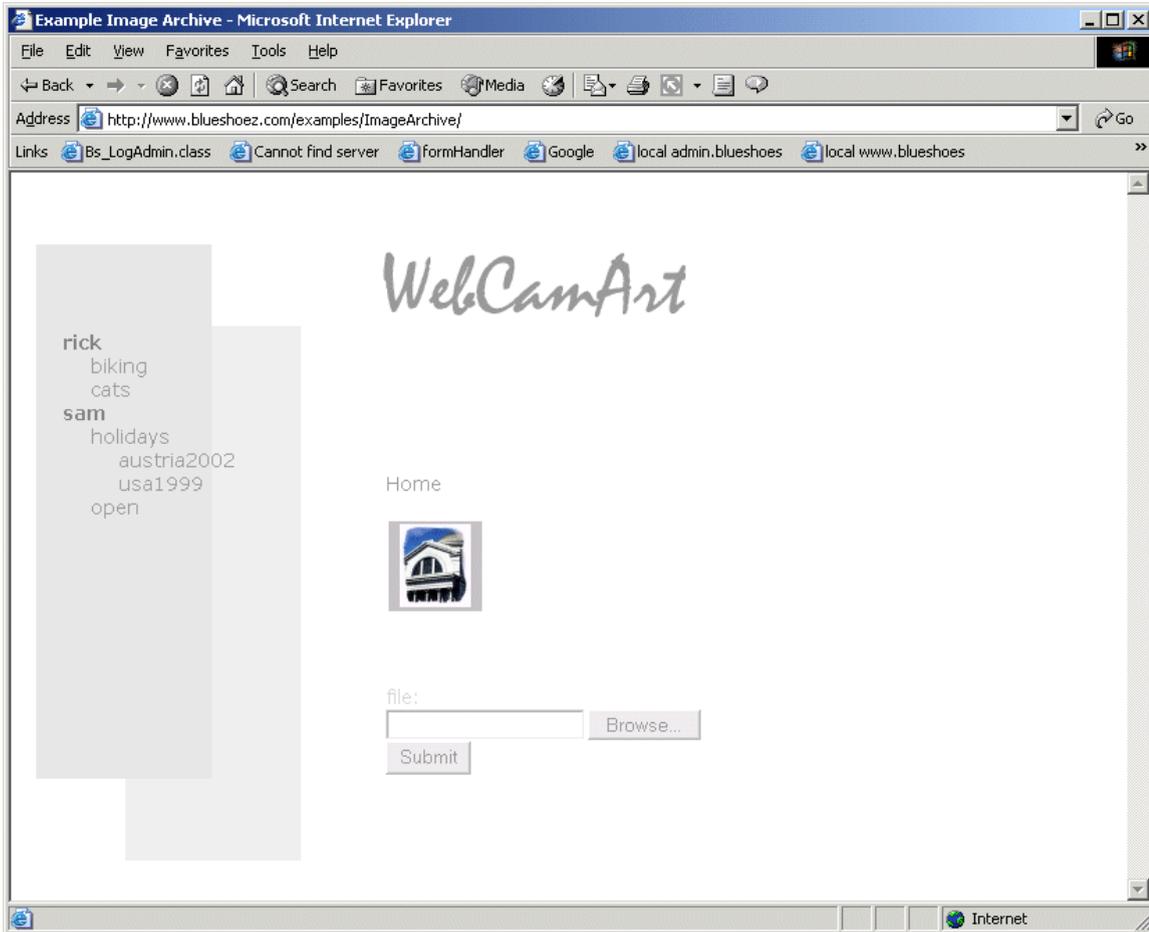


Let's upload a file (click browse, select your image, click submit).

Now you get a message that says it worked. At this time it's still hardcoded. Some language handling should be added here.



If you log in using our visitor user, you can see that the folder “private” from rick is not here. A logout button is missing. Close your browser and reconnect to your image archive.



References

- 1 Bs_Cug, see the classes in core/auth/cug/ and the HOWTO at <http://www.blueshoes.org/howtos/BSCugHowTo.pdf>
- 2 phpMyAdmin, a front-end for the MySQL database, see <http://www.phpmyadmin.net/>