Remository 3.52.8

Posted by chrissy6930 - 2009/10/21 18:29

---

hi there,

I have installed Remository on my test session now

Joomla Multi Sites version: 1.2.10 (Latest available: 1.2.10)
Patches definition version: 1.2.13 (Latest available: 1.2.13)

after adjusting the config I went to tools and selected my dummy site

here I selected to install Remository from master and ticked the checkbox to propagate to children

then I went to the two slaves and created component type menu items

I submitted a file with 1 image each on either of the slave sites

iwent and checked the files in filelist which looked ok. I can see the file with the appropriate title, the image I added and the download is working

however when going to fileinfo I noticed that each of the files displays 2 image, i.e. all the images with ID 1 are being displayed on either slave site

checking the db shows that table slv000_downloads_files & slv001_downloads_files hold the correct data - both files come with fileID = 1

checking the file structure shows that both images are in ../components/com_remository_files/file_image_1

also despite having disabled database storage of files I cannot find them anywhere on the server

could you please advice what I'm doing wrong?

==========

Re:Remository 3.52.8

Posted by edwin2win - 2009/10/21 20:00

No idea.
The only solution that I see would be to debug on your environment and try identifying the peace of code that produce those 2 infos.
Once identified, it will be more easier to understand the origin of this issue.

If you want that we debug on your environment, this is possible with our billable support that you can order on
http://www.jms2win.com/download?page=shop.product_details&flypage=flypage.tpl&product_id=35&category_id=1

==========

Re:Remository 3.52.8
Posted by chrissy6930 - 2009/10/22 10:34

I will start with images::

when saving a submission there is first
the savefile_controller which is being called (com_remository/c-classes)
which in turn is calling the savethumb_controller (c-classes)
which is calling function addImage in remositoryThumbnails (p-classes)

```
public function addImage ($physicalFile, $fileid, $legend='', $remositoryFile=false) {
 $interface = remositoryInterface::getInstance();
 if ($nextfree = $this->getNextFree()) {
  $filepath = $interface->getCfg('absolute_path').$this->filepath;
  if (!is_dir($filepath)) mkdir($filepath);
  $file_to_fix = $filepath.time().$physicalFile->proper_name;
  if ($remositoryFile) $physicalFile->copyToFileSystem($file_to_fix);
  else $physicalFile->fileToFile($file_to_fix, 0, false, false);
  $file_ext = remositoryAbstract::lastPart($file_to_fix,'.');
  if ($file_ext != 'png' AND $file_ext != 'jpg' AND $file_ext != 'jpeg' AND $file_ext != 'gif') {
   if (!$remositoryFile) echo _DOWN_THUMB_WRONG_TYPE;
  }
  elseif (file_exists($file_to_fix)) {
   $large_image = $filepath.'img_'.$fileid.'_'.$nextfree.'.'.$file_ext;
   $small_image = $filepath.'th_'.$fileid.'_'.$nextfree.'.'.$file_ext;
   $repository = remositoryRepository::getInstance();
   $this->imgresize($file_to_fix, $large_image, $repository->Large_Image_Width,
$repository->Large_Image_Height, true);
   $physicalimage = new remositoryPhysicalFile();
   $physicalimage->setData($large_image);
   $physicalimage->setPerms();
   $this->imgresize($file_to_fix, $small_image, $repository->Small_Image_Width,
$repository->Small_Image_Height, true);
   $physicalimage->setData($small_image);
   $physicalimage->setPerms();
   unlink ($file_to_fix);
   if ($legend) {
       $legend_path = $filepath.'txt_'.$fileid.'_'.$nextfree.'.'.$file_ext;
       if ($fp = fopen($legend_path, 'wb')) {
           fputs($fp, $legend);
           fclose($fp);
       }
   }
  }
  else echo "._ERR1.";
 }
}
```

there are some other snippets in p-classes/remositoryThumbnails that might be relevant for this matter:

```
public function __construct ($file) {
 $interface = remositoryInterface::getInstance();
 srand((double)microtime()*1000000);
 $this->fileid=$file->id;
 $repository = remositoryRepository::getInstance();
 $this->maxcount = $repository->Max_Thumbnails;
 $this->allow_large = $repository->Allow_Large_Images;
 $this->filepath = remositoryThumbnails::baseFilePath().remositoryThumbnails::dirPattern().$file->id.'/';
 if ($this->maxcount) $this->findFiles();
 elseif ($file->screenurl) {
  $this->thumb_URLs = $file->screenurl;
  $this->img_URLs = $file->screenurl;
  $filepath = str_replace($interface->getCfg('live_site'), $interface->getCfg('absolute_path'),
$file->screenurl);
  $this->thumb_paths = $filepath;
  $this->img_paths = $filepath;
  $this->count = 1;
 }
 else $this->count = 0;
}

public static function baseFilePath () {
 return '/components/com_remository_files/';
}

public static function dirPattern () {
 return 'file_image_';
}
and

private function findFiles () {
 $interface = remositoryInterface::getInstance();
 $thumb_pattern = 'th_'.$this->fileid.'_';
 $dir = new remositoryDirectory($interface->getCfg('absolute_path').$this->filepath);
 $thumbfiles = $dir->listFiles($thumb_pattern);
 $this->count = 0;
 foreach ($thumbfiles as $i=>$thumb) {
  $numberandext = remositoryAbstract::lastPart($thumb,'_');
  $k = intval(remositoryAbstract::allButLast($numberandext,'.'));
  $marker = 'X';
  $thumbpath = $this->filepath.$thumb;
  $this->thumb_paths = $interface->getCfg('absolute_path').$thumbpath;
  $this->thumb_URLs = $interface->getCfg('live_site').$thumbpath;
  $imagefile = str_replace('th_','img_',$thumbpath);
  $textfile = $interface->getCfg('absolute_path').str_replace('th_', 'txt_', $thumbpath);
  if (file_exists($interface->getCfg('absolute_path').$imagefile)) {
   $this->img_paths = $interface->getCfg('absolute_path').$imagefile;
   $this->img_URLs = $interface->getCfg('live_site').$imagefile;
  }
  else {
   $this->img_paths = $this->thumb_paths;
   $this->img_URLs = $this->thumb_URLs;
```

```
  }
  if (file_exists($textfile)) {
      if ($fp = fopen($textfile, 'rb')) {
          $this->img_texts = fgets($fp);
          fclose($fp);
      }
  }
  $this->count = $i+1;
 }
 $freecount = 0;
 unset($this->allfree);
 for ($i=1; $imaxcount AND ($this->count + $freecount) maxcount; $i++) {
  if (!isset($marker)) {
   $this->allfree = substr($i+100,1);
   $freecount++;
  }
 }
 $this->freecount = $freecount;
}
```

I will be posting code for saving file in a separate post

==========