

I have read several discussion on this but I have not seen what I believe to be a good answer to the various possible conditions.

I believe I have a reasonably strong solution, but would love to hear from anyone who could improve on it.

The issues:

1. Delete the old file if there is a new one uploaded.
2. If the new file uploaded has the same name as the old one, DO NOT ACCIDENTALLY DELETE THE NEW ONE WHEN USING THE OVERWRITE METHOD FOR THE FILE NAME
3. Save the old file and do not update the database field if the upload fails

To do this, I added a file delete behavior for the existing file name in the database, then the upload behavior, and triggered both off the submit button.

Then the update transaction. **You may have to juggle the code around to get it in this order.**

Now, here is the trick. Enclose both the delete and the upload transactions in a PHP if statement that checks for whether or not there was an error in the file upload.

***(Note the value of the PHP \$\_FILES array for the error key IS 0 if there were no errors, and an empty file field actually produces an error code of 4)***

```
<?php
// Test for file upload that was successful through PHP
if (isset($_FILES["musicfile"]) && ($_FILES["musicfile"]["error"] === 0)) {
// If successful, do delete first so if the new filw has the same name it won't matter
// Enclose both delete and upload in this condition so if there was any error in the upload, it won
?>
<?php
$WA_DeleteFileResult1 = false;
if(isset($_POST["Update_x"])){
$WA_DeleteFileResult1 = WA_FileAssist_DeleteFile(".././musicfiles/", "".$row_WADAmusic['musicfile'])
}
?>
<?php
// WA_UploadResult1 Params Start
$WA_UploadResult1_Params = array();
```

```
// WA_UploadResult1_1 Start
$WA_UploadResult1_Params["WA_UploadResult1_1"] = array(
'UploadFolder' => "../musicfiles/",
'FileName' => "[FileName]",
'DefaultFileName' => "",
'ResizeType' => "0",
'ResizeWidth' => "0",
'ResizeHeight' => "0",
'ResizeFillColor' => "" );
// WA_UploadResult1_1 End
// WA_UploadResult1 Params End
?>
<?php
WA_DFP_SetupUploadStatusStruct("WA_UploadResult1");
if(isset($_POST["Update_x"])){
WA_DFP_UploadFiles("WA_UploadResult1", "musicfile", "0", "", "false", $WA_UploadResult1_Params)
}
?>
<?php
// End error check condition for WA behaviors
}
?>
```

Now set the update binding for the file name to the WA Server File Name (\$WA\_DFP\_UploadStatus["WA\_UploadResult1"]["serverFileName"]) with a default to the recordset value of the file name (the original file name)

```
<?php echo ((isset($WA_DFP_UploadStatus["WA_UploadResult1"]["serverFileName"]))?$WA_DF
```

If there is any upload error, the WA upload code will not be executed, so \$WA\_DFP\_UploadStatus["WA\_UploadResult1"]["serverFileName"] will not be set.

Now if no file is uploaded, or there is any error in the upload, nothing will be deleted or uploaded (copied from temp to the web site space), and the file name will remain unchanged.

If a new file is uploaded, the old will be deleted, the new uploaded, and the database updated correctly.

If the new upload has the same name as the old one, the old file will be deleted first, the new one uploaded, and the database name will still be correct.

*None of this will protect against collisions with the same file name from a different record, so always use caution with overwrite.*

*This method will also work fine with a renaming scheme like incrementing except you will never increment unless the file with the same name belongs to another record.*

*It would also work with a fixed file name like File\_RecordID.jpg*