

Coding Note - Consistency

data_path and **data_filename** will be deprecated. They are 'covers' for **detect_upload**, except they involve significantly more effort (two db calls). In addition I have discovered that they are being used in conjunction with **detect_upload** – so the work is being done three times except just once.

detect_upload() – returns the file path and original file name of an upload if any, and false otherwise. **data_path()** returns the file path if there has been an upload, and **data_filename()** returns the original file name of an upload if any. Only a single call to **detect_upload()** is required.

This is typically the code I have found:

```
$file_to_process = detect_upload();
if $file_to_process is not false then
    console("Upload file  {$file_to_process.FileName} ..
{$file_to_process.FilePath}");

$data_path = data_path(); // this will be null if you use the test file
$file_to_read = data_filename(); // this picks up the upload or data
offering

if $file_to_read is null begin
/* for testing, use in place test file, and say so */
$file_to_read = "RAPxmldlyexp20120810.xml";
$data_path = $file_to_read;
console("Using test file {$data_path}");
$file_lines_limit = 60; // for testing only load this number
console("Testing restricted load to {$file_lines_limit} records");
end
$match_date =
preg_match("/.*(?P<year>[\d]{4,4})(?P<month>[\d]{2,2})(?P<day>[\d]{2,2})\.xml/i", $file_to_read);
```

There is double testing of the ‘fact’ that there has been an upload (testing `file_to_process` and `file_to_read`).

I recommend this style:

```
$uploaded = detect_upload();
if $uploaded is false begin
    $uploaded.FilePath = "RAPxmldlyexp20120810.xml";
    $uploaded.FileName = $uploaded.FilePath;
    $file_lines_limit = 60;
    console("No upload detect, using test file {$uploaded.FileName} and
restricted lines {$file_lines_limit}");
End
else console("Upload detected {$uploaded.FileName} in
{$uploaded.FilePath}");

$match_date =
preg_match("/.*(?P<year>[\d]{4,4})(?P<month>[\d]{2,2})(?P<day>[\d]{2,2})\.xml/i", $uploaded.FileName);
```

From then on, do not have a variable `$file_to_read`, but use `$uploaded.FilePath` or if you want the original file name (as a unique key for example) use `$uploaded.FileName`.

This is not me being picky, there is a valid reason. While that sequence is 'valid' (but inefficient and messy) for development it is not valid for going live, where a "non-upload" should be treated as an error not an opportunity for testing. So rather than changing code between dev and live we could cause detect_upload to have a flag to allow an alternative/test filename to be presented back on return.