Subject: Error messages not being returned to previous screen
Posted by ljkbrost on Fri, 06 Nov 2009 05:39:54 GMT
View Forum Message <> Reply to Message

When the task 'table(upd4)export.php' gets called while attempting to export the table contents to PHP, error messages are not displayed.

I've created a subsystem but failed to create the subsystem directory in the file system. When I 'Export to PHP' the radicore framework detects that it could not create the file, records an error in $this->errors[] but this does not get displayed back to the calling screen.

So it looks like it worked but in reality, it did not.

Tracing this problem has been a real pain. At first I thought the comparison in std.update4.inc was failing:

```
if ($dbobject->errors) {
    // some sort of error - return to previous script
  $batch_errors = scriptPrevious($dbobject->getErrors, $messages, null,
$dbobject->getInstruction());
    return $batch_errors;
} // if
```

But through my 10 or 15 comparisons of $dbobject->errors versus $dbobject->getErrors() using count(), sizeof(), empty(), is_set()_, etc... I found that the function getErrors() has a problem. When you call getErrors() it is a destructive function.

```
    function getErrors ()
    // return array of error messages
    {
        $errors = $this->errors;
        $this->errors = array();

        if (!is_array($errors)) {
            // convert string into an array
            $errors = (array)$errors;
        } // if

        return $errors;

    } // getErrors
```

When you call this function more than once the $this->errors will get purged. It should really look like:

```
    function getErrors ()
    // return array of error messages
    {
        $errors = $this->errors;
        $this->errors = array();

        if (!is_array($errors)) {
            // convert string into an array
            $this->errors = (array)$errors;
        } // if
        else {
          $this->errors = $errors;
        }

        return $this->errors;

    } // getErrors
```

While the above fix does what it should do. It does not fix the problem. You still don't get the error message sent to the previous page.

Doing a quick audit of the various files in the includes directory, there are 8 other instances of the getErrors() functions defined. The other instances are a simple and just return $this->errors. They are not designed to arrays of error strings. It may be time to review the other functions and their uses to ensure the differences make sense.

So back to the original code:

```
if ($dbobject->errors) {
    // some sort of error - return to previous script
  $batch_errors = scriptPrevious($dbobject->getErrors, $messages, null,
$dbobject->getInstruction());
    return $batch_errors;
} // if
```

Looking at the code the call to $dbobject->getErrors does not have brackets for the function call. Doing a print_r($dbobject->getErrors) and print_r($dbobject->getErrors()) gives two different results. The first returns nothing the second returns what you would expect.

Changing the code to:

```
if ($dbobject->errors) {
```

```
    // some sort of error - return to previous script
  $batch_errors = scriptPrevious($dbobject->getErrors(), $messages, null,
$dbobject->getInstruction());
    return $batch_errors;
} // if
```

Causes the error message to get propogated back to the previous screen and everything appears to be fine.

Doing a quick search through the code base, this is the only area where the call is $dbobject->getErrors. Everywhere else it appears that it is $dbobject->getErrors(). I would suggest this gets fixed in std.update4.inc.

Cheers,

Kyle Brost
---
www.softelephant.com