Subject: Enum Type and 0 index.
Posted by johandewit on Mon, 13 Nov 2006 13:19:52 GMT
View Forum Message <> Reply to Message

File : includes/std.validation.class.inc
Funtion : validatefield

```
    if ($fieldspec['type'] == 'enum') {
        debugbreak();
        // get enum array for this field
        $enum = $this->caller->getValRep($fieldname);
        // if we already have the value do not replace it
        if (!in_array($fieldvalue, $enum)) {
            // replace index number with text value
            $fieldvalue = $enum[$fieldvalue];
        } // if
    } // if
```

Problem ; If $fieldvalue=0, where 0 = array_index, the function in_array returns TRUE instead of FALSE. The fieldarray is not replaced with the $enum[0] value, resulting in validation error.

See also http://be.php.net/manual/en/function.in-array.php#61491

Changing line 111 to if (!in_array($fieldvalue, $enum, true))
solves this, and the fieldvalue is rpelaced with $enum[0] as intended.

At the end of this function, the enum type is passed to the validateNumber, (default case branch) which always results in an error message.

I've attached a patch against version 1.17.0 which makes the enum field type working (in my situation).

Greetings

Johan

File Attachments

1) std.validation.class.inc.patch, downloaded 1842 times

Subject: Re: Enum Type and 0 index.
Posted by AJM on Mon, 13 Nov 2006 17:57:27 GMT
View Forum Message <> Reply to Message

I created a field with the type enum('one', 'two', 'three') and tested it with both NULL and NOT NULL. The field was displayed as a dropdown list, and in one test I obtained the values using

```
function _cm_getValRep ($item='', $where)
{
    if ($item == 'enum_test') {
        $array = $this->getEnum($item);
        return $array;
    } // if
}
```

and in another test using

```
function _cm_getValRep ($item='', $where)
{
    if ($item == 'enum_test') {
        $array['one']   = 'un';
        $array['two']   = 'deux';
        $array['three'] = 'trois';
        return $array;
    } // if
}
```

In neither test did I have the need to change line 111 to

```
if (!in_array($fieldvalue, $enum, true))
```

I did see, however, the need to insert

```
    case 'enum':
        break;
```

before

```
    default:
        // perform validation if field type is numeric (integer, decimal)
        $fieldvalue = $this->validateNumber($fieldname, $fieldvalue, $fieldspec);
} // switch
```

I also found and fixed a problem in file 'include.xml.php4/5.inc' which inserted a blank entry into the lookup array even if there was one already there. My changed files are attached.

File Attachments

1) radicore-2006-11-13-update.zip, downloaded 1883 times

Subject: Re: Enum Type and 0 index.
Posted by johandewit on Tue, 14 Nov 2006 10:51:50 GMT
View Forum Message <> Reply to Message

Id there a file missing in the because now i get a undefined reference for function

include.xml.php4.inc:1116: $id = removeTableSuffix($id);
include.xml.php5.inc:1114: $id = removeTableSuffix($id);

Greetings.

Btw. I got the error because the enum field in my case is hidden, and i was setting it in the _cm_pre_insertRecord function to the first value from yhe fieldspec['enumfield']['values'] array.

Greetings

Johan

Subject: Re: Enum Type and 0 index.
Posted by AJM on Tue, 14 Nov 2006 11:13:35 GMT
View Forum Message <> Reply to Message

Whoops, sorry about that. Attached is a new zip file which should fix those undefined references.

I cannot seem to reproduce that other error. Can you supply me with the contents of your ['enumfield']['values'] array?

File Attachments
1) radicore-2006-11-14-update.zip, downloaded 1884 times

Subject: Re: Enum Type and 0 index.
Posted by johandewit on Tue, 14 Nov 2006 12:57:19 GMT
View Forum Message <> Reply to Message

```
function _cm_getExtraData ($where, $fieldarray)
// Perform custom processing for the getExtraData method.
// $where = a string in SQL 'where' format.
// $fieldarray = the contents of $where as an array.
{
    $this->lookup_data['input_state'] = $this->fieldspec['input_state']['values'];
```

I believe above code is the source. Using above code is an indexec array, while your lookup array is an associative one.
Your code will work because the 'abnormal' behaviour of the in_array only occurs whenusing

indexed array, end only if index = '0'.

I've used above statement to fill up the lookup data because i do not have to change the code if the enumaration is changed in the database. This is done by the regeneration of the dict file.

I also uses the enum type here because depending on the contents of that field, I have to process the data in a specific and/or different way.

This interpretation of the enum type was also the reason to use the fieldspec['enumfield']['values'] directly in the std.validation.class.inc instead of the call

$enum = $this->caller->getValRep($fieldname)

because the fieldspec is passed as a parameter.

The getEnum on the other hand, retreives the enum values from the database.

I'm a little confused now about the prefered way to work with the enumaration type.

File Attachments

1) input.dict.inc, downloaded 1818 times

Subject: Re: Enum Type and 0 index.
Posted by AJM on Tue, 14 Nov 2006 13:56:12 GMT
View Forum Message <> Reply to Message

The correct way to load an ENUM field is to put the following code in the _cm_getValRep() method:

```
if ($item == 'enum_field') {
    $array = $this->getEnum($item);
} // if
```

This obtains the array of values from the database, but is indexed from 1 not 0.

I think I should change the export function so that when it creates the array of values in the ?.dict.inc file it forces the first index to be 1, as follows:

'values' => array(1 => 'entered', 'modified', 'submitted', 'approved', 'rejected', 'deleted')

The reason that I stopped using ENUM fields in my code is that Radicore has to support other databases, and ENUM fields are unique to MySQL.

Subject: Re: Enum Type and 0 index.
Posted by johandewit on Tue, 14 Nov 2006 14:25:14 GMT
View Forum Message <> Reply to Message

Thanks forr the explenation.

Does this mean you wil drop support for the enum fieldtype in the future ? If so, then it is better also not to use the enum type anymore, because it is mysql specific.

I will rethink the enum type usage, and it looks like this is a good FAQ entry. I will give it a try and send you my attemp.

Thanks for your help

Johan

Subject: Re: Enum Type and 0 index.
Posted by AJM on Tue, 14 Nov 2006 14:46:51 GMT
View Forum Message <> Reply to Message

If you know that your database will always be MySQL then you can continue to use ENUM fields. If you ever decide to use another DBMS which does not have ENUM fields then you will have a problem.

I do not plan to remove support for ENUM fields from Radicore so you can carry on using them. I stopped using them in my sample application as I needed a version that could also run on PostgreSQL. This means that I can switch from MySQL to PostgreSQL and back again just by changing one line in the CONFIG.INC file. No code anywhere else needs to change.

I shall update the FAQ to include information regarding ENUM fields.

Subject: Re: Enum Type and 0 index.
Posted by johandewit on Thu, 16 Nov 2006 08:52:47 GMT
View Forum Message <> Reply to Message

Hi,

Another minor thing due to the (in between)update.

The update_array() functions seems to be changed to update_array_indexed(0 and update_array_associative().

Now I got following error.

from eg. http://localhost/radicore/dict/column_upd.php

Fatal error: Call to undefined function array_update() in /tools/lampp/htdocs/radicore/includes/std.update1.inc on line 141

Concerning the enum field type, I think I,m going to follow your advice and remove the enum field type from the database and put it in my code. The code depends on the state of this field, and I think it is better if such stays in the code itself instead of the database.

Greetings and thanks a lot for your great support.

Johan

Subject: Re: Enum Type and 0 index.
Posted by AJM on Thu, 16 Nov 2006 10:52:10 GMT
View Forum Message <> Reply to Message

Here's a full set of my changes since the last release. I hope to finalise a new full release in the near future.

It is still OK to keep using ENUM fields as they will always be supported in Radicore.

File Attachments
1) radicore-2006-11-16-update.zip, downloaded 1857 times
