[WORKING] mosDBTable class

Last Updated Monday, 25 December 2006

Last reviewed: Not reviewed

phpDocumentor

Abstract database table class. This is the parent class to all database derived objects. Customisation will generally not involve tampering with this class.
Each table of the Joomla database has a class of type mosDBTable associated with it. The parent class defines a set of default functions that will suffice for the majority of tables, but which may be overridden if required.
If you are creating a component which requires its own table then you
are strongly advised to use the mosDBTable class so as to take advantage
of this default functionality.
For example, suppose you have a table called myTable, containing just three fields: $id, $title and $published. Then you could define a class called, perhaps, myTable, as follows:/**

```
* myTable database table class
* @package MyComponent
*/
class myTable extends mosDBTable {
 /** @var int Primary key */
 var $id=null;
 /** @var string */
 var $title=null;
 /** @var boolean */
 var $published=null;

 /**
 * @param database A database connector object
 */
 function myTable( &$db ) {
  $this->mosDBTable( '#__myTable', 'id', $db );
 }

}
```

Notice that you MUST define exactly the same attributes in the class as
the fields in the table schema definition. It is a common programming
error to add a new field to the table schema and forget to add the same
field as an attribute in the class definition.Â A symptom typical of this is when changes made to a database variable in program code are never saved to the physical database record.Â Also make sure that you set each variable to null otherwise problems can arise in some versions of PHP 4.1.x.

You could then declare an instance of this table as follows:

```
$myTbl = new myTable( $database );
```

Having declared an instance of the table the default functions allow
for most requirements. For example, updating a row can be done like this:

```
// Declare an instance of the myTable class.
$myTbl = new myTable( $database );

// Load record with primary key id=17.
$uid = 17;
$myTbl->load( $uid );

// Update some fields in the table row.
$myTbl->set( 'title', 'The title' );
$myTbl->set( 'published', true );

// Save the updated record.
$myTbl->store();
```

More commonly the load will take place in one call to Joomla, where a
form is constructed which the user fills in with the updated fields.

The user then submits the form which calls a different routine in the component which will bind the $_POST array to the table object and then store the object in the database table.
Joomla version

Unknown

Defined in

libraries/joomla/database/mysql.php
(includes/database.php prior to Joomla 1.1)
Functions

mosDBTableConstructor for the mosDBTable class.
bind
Copy the contents of an array into the database table object.
canDelete
Checks whether it is safe to delete a record from the table by checking whether dependancies exist for this record in the database schema.
checkChecks that current object attributes are valid.

checkin
Generic check-in function.
checkout
Generic check-out function.
delete
Generic delete function.
filter
Filters all public properties of the class to remove malicious code or unwanted HTML tags.
get
Returns the value of a class attribute.
getError
Returns the most recent error message.
getPublicProperties
Returns an array of public properties in the class.
hit
Generic hit count update.
isCheckedOut
Tests if the current item is checked out. This method was introduced in Joomla 4.5.3.
load
Generic load object from database table row.
move
Generic move function.
publish_array
Generic publish/unpublish function.
save
Generic save function.
set
Sets the value of a class attribute.
store
Inserts a new row or updates an existing row in the database table.
toXML
Exports the current object in XML.
updateOrder
Compacts the ordering sequence of the selected records.

Please report any errors on this page to the Developer Documentation Forum.