Subject: Update 4
Posted by danpoleary on Thu, 28 Jun 2007 17:02:53 GMT
View Forum Message <> Reply to Message

A little lost,

I have a 3 tasks based on tablename(upd4).php, that I would like to use to update multiple records. I have read the "update 4" in the dialog-types, which says perform the modifications within the table class, but am not sure how to proceed.

The three I have are tablename_task1(upd4).php, tablename_task2(upd4).php and tablename_task3(upd4).php. I want to be able to detect which one is called inside the table class, and update certain fields accordingly. Any examples of using update 4, that I can refer to?

Thanks in advance,

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 28 Jun 2007 21:59:26 GMT
View Forum Message <> Reply to Message

If you have several different tasks which need to update the same database table, but perform different processing, you need to take a look at
http://www.tonymarston.net/php-mysql/infrastructure-faq.html #faq49

If the amount of differences in the code is significant then I would recommend using a different subclass for each task. In this way the code for each task would be completely separate and easier to maintain.

Subject: Re: Update 4
Posted by danpoleary on Fri, 29 Jun 2007 13:36:00 GMT
View Forum Message <> Reply to Message

Thanks,

Those functions are what I was looking for. But where in the table class would I call them?

Basically, I want to set the combination of 5 boolean fields to a certain state depending on which of the 3 tasks was called for all records selected in the previous screen.

Now the second problem, is how would I provide a popup within this sequence, to allow the user to pick the content for one field from a dropdown keyed from another table?

The UPD4 is one of a few, that does not have examples I can find anywhere on either site.

Thanks,
Dan

Subject: Re: Update 4
Posted by AJM on Fri, 29 Jun 2007 14:24:31 GMT
View Forum Message <> Reply to Message

If you want to find a working example of any transaction pattern the the best place to look is Home->Menu System->List Pattern, choose the pattern you want, then press the 'List Task' button. This will show you all the tasks in the system which follow that pattern.

If you try this with UPD4 you will see the following tasks:
- dict_database(export)
- dict_table(export)
- mnu_subsystem(build)
- mnu_subsystem(export)

If you want to see which methods are used by a particular patter then you need to look at http://www.tonymarston.net/php-mysql/dialog-types.html, pick a pattern, then scroll down till you see the heading 'Object Calls'. If you do this for UPD4 you will see the following sequence of method calls:
- initialise()
- getData()
- startTransaction()
- updateMultiple()
- commit()

Each one of these is a hyperlink that will take you to http://www.tonymarston.net/php-mysql/functions-and-variables .html for a more detailed explanation as some of these methods may call other methods.

Once you realise which methods are called in which sequence it should be easy to identify where your code should go.

If you want to introduce a popup form in this sequence so that the user can choose a value then you should not be using the UPD4 pattern as this does not allow any dialog with the user. If you want any user dialog before the update you should be using the UPD1 pattern

Subject: Re: Update 4
Posted by danpoleary on Fri, 29 Jun 2007 15:05:06 GMT
View Forum Message <> Reply to Message

Hi,

Thanks!

All of that leads back to where my problem lies. I need to set these booleans of selected records (100's at a time) but I also need one key field set to the correct value from a foreign table. Update 1, does not allow me to update all the records at once.

I have all the code done and working for setting the booleans (thanks for the examples) but now wish to find a way to pre-select the key that is also to be set for all those records. Can I force a call to a popup that will list the foreign tables values, which upon selection will return the key. Then I could just $rowdata['detination_key'] = $returned_value;

Thanks

Subject: Re: Update 4
Posted by AJM on Fri, 29 Jun 2007 16:42:24 GMT
View Forum Message <> Reply to Message

If you need to perform the same update on multiple records it sounds like you need to use the updateSelection() method instead of the updateRecord() method. The updateSelection() method uses two arguments, $selection and $replace, and will issue an SQL query in the format "UPDATE <table> SET $replace WHERE $selection". If you also want to call a popup to obtain a key from a foreign table to include in the update then take a look at the UPDATE 2 pattern.

Subject: Re: Update 4
Posted by danpoleary on Fri, 29 Jun 2007 18:54:40 GMT
View Forum Message <> Reply to Message

Thank you,

I will give that a try tonight. Part of the problem, is that this is made up of three tables. One table called "files" has two parents, one called "sources" and one called "destination". The "files" table has the "sources" key populated already.

From this, I have the sources table displayed, I then pick a source, and call the task (update_multiple) Multi2, which brings up a list of all files from that source. Right now to change the records (100's) is very painful one at a time. so that is why I wanted to update multiple, setting the booleans and the pre-selected destination key.

Still learning the framework and liking it quite a bit. Keep up the good work.

dan

Subject: Re: Update 4
Posted by AJM on Fri, 29 Jun 2007 19:32:25 GMT
View Forum Message <> Reply to Message

When you bring up the list of files from the chosen source, is it your intention to process all of them, or just a selection of them? Are you moving them to a different destination, or is other processing involved?

When you are attempting something which is rather complicated it may appear that there is not a pattern which fits the job 100%, but in that case you must try the nearest match then insert custom code which can override the default behaviour of that pattern. Every different pattern executes a different set of methods in a predeterined sequence. All of the methods which begin with '_cm_' are customisable, which means that you can copy the empty stub from 'std.table.class.inc' into your table class, then insert your own code which will be executed when that method is called. You should find a '_cm_pre_' and '_cm_post_' method either side of each standard method, so you have the opportunity to execute your own code before and after each standard method.

Once you become more familiar with the framework you should realise that the ability to have your own code executed in several different places along the processing flow is a really powerful tool. The biggest difficulty is being able to pick the most appropriate pattern, then knowing what code to put where. It gets easier with experience (or so I'm told)

Subject: Re: Update 4
Posted by danpoleary on Tue, 03 Jul 2007 15:47:14 GMT
View Forum Message <> Reply to Message

Hi,

Tried a few things, but not sure if my final logic will work.

If I am on the sources table screen, and select (lets say source1), then click "update multiple files". The update multiple screen, will provide a list of all files, from the files table, from that source.

I then select the files I want to set to "publish", then click on "publish selected". At this point, in the publish selected script (inside the custom table class that extends the "files" table class) I then use a call to scriptNext, to call a popup that lets me select the "destination" key from the destination table.

After I call scriptPrevious, and now back inside my publish script, I set the boolean value of publish to "true" and the destination key value to the the return value.

Will this logic work, or am I heading in the wrong direction?

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Tue, 03 Jul 2007 17:14:20 GMT
View Forum Message <> Reply to Message

You could always try it and see, but my personal approach would be different. After selecting the source I would activate a MULTI4 screen which would show me the source directory at the top (outer/parent) with the files within that directory at the bottom (inner/child). The interesting aspect to the MULTI4 is that it allows updates to both the parent and child entities, so in the source area you could also select the destination.

After selecting entries from the files (inner/child) area I would then have a navigation button which used an UPDATE4 to perform a specific action on the selected files.

This is made possible by the following tricks you can use within custom methods:

- in order to be able to have a "destination" field in the "source" area you can have code in the _cm_changeConfig() method which alters the contents of $this->fieldspec. You could have this as a popup or a dropdown depending on the number of possible entries.

- in order to make the contents of "destination" available in the "files" area as well as the "source" area when a button is pressed you would also need to add it to $this->fieldspec of the "files" class.

- when you press a navigation button the default behaviour is to pass the values from the primary keys of the selected records to the child screen, which I think in this case would be "source" and "filename". You can use the _cm_getPkeyNames() method to add "destination" to the list of primary key fields so that its value can be included in the selection string which is passed to the UPDATE4 script, which then has all the information it needs to perform the necessary processing.

How does that sound?

Subject: Re: Update 4
Posted by danpoleary on Wed, 04 Jul 2007 14:47:08 GMT
View Forum Message <> Reply to Message

Hi Tony,

Yes, I looked through what you mentioned, but have a few questions.

You say to use the _cm_changeConfig function. What would be needed to list the other table(destination) as a drop down within the table(sources). What I have so far is:

```
 $this->fieldspec['table_id_snr']['control']       = 'popup';
 $this->fieldspec['table_id_snr']['task_id']       = 'destination(popup)snr';
 $this->fieldspec['table_id_snr']['foreign_field'] = 'table_id_snr';  <===== should this be the field ID from the destination table?
 unset($this->fieldspec['table_id_snr']['noedit']);
 return $fieldarray;
```

Is that what you were referring to?

Then, how would would I access that data within the call to update4? You mention _cm_getPkeyNames, but how would that work, when the current value of the destination key field in the files table is NULL?

I looked for example in some of the other code, but not sure if I have the correct idea here.

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Wed, 04 Jul 2007 16:42:50 GMT
View Forum Message <> Reply to Message

All information about popups is avalable in the FAQ:

What is a popup?
How to incorporate a 'popup' control into a form
How does the processing of a 'popup' form actually work?
How can you switch a popup between accepting single and multiple selections?

The value for 'foreign_field' can be any value from the foreign table - either the primary key, or any non-key field. It's up to you what you want displayed.

If you add 'destination' to $this->fieldspec in the 'files' object as well as the 'source' object, then the value for 'destination' should be given to both objects when a submit button is pressed. Once the value is avalaible in $this->fieldarray it can be made to appear in the selection string by adding it to the array in _cm_getPkeyNames.

Subject: Re: Update 4
Posted by AJM on Wed, 04 Jul 2007 17:50:21 GMT
View Forum Message <> Reply to Message

I've just run a test and discovered that the value from the outer entity is not being loaded into the inner entity, so I've made a little change to the array_update_indexed() function inside file include.general.inc which solves the problem.

File Attachments
1) include.general.inc, downloaded 2196 times

Subject: Re: Update 4
Posted by danpoleary on Thu, 05 Jul 2007 13:56:33 GMT
View Forum Message <> Reply to Message

Hi Tony,

Before you sent that update, I did get something to work. When I clicked on the multi4 after choosing a source, I get the screen with the source data editable at the top as expected, but the field for the destination_key/desc does not appear.

When I replaced my include.general.inc with the updated one, everything stopped working, blank screen on login. I then reverted the include file, and only incorporated the changed function. Now I can log in, but my call to multi4 brings up a blank screen.

In the popup faq's above, do I need to update the dict file for the source table? If I do, will that not appear on the standard source list1 screen all the time?

and, should I add the field description to the screen.inc file used for my multi4?

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 05 Jul 2007 14:31:48 GMT
View Forum Message <> Reply to Message

I do not understand why your MULTI4 starts off with a blank screen as the change I made to the array_update_indexed() function is only used after the form has been displayed and a button is pressed.

The field for destination_key/desc will not appear in the 'source' area unless it has an entry in the $fieldspec array within the 'source' object which identifies the type of HTML control which is to be used. It must also have an entry in $fieldarray which can either be blank or set to an initial value.

You do not need to create a relationship between 'source' and 'destination' in the dictionary unless there is a true relationship. The only reason for reading the foreign table in a relationship is to obtain a non-key description field which is to be displayed in the popup control instead of the foreign key. If you are happy to show the foreign key then you do not need any reference to a foreign table. If you do wish to read the foreign table to obtain a non-key field, and a true relationship has not been defined in the dictionary, you can always insert code in the _cm_getForeignData() method to read the table manually and insert the non-key field into $fieldarray.

Subject: Re: Update 4
Posted by danpoleary on Thu, 05 Jul 2007 15:00:08 GMT
View Forum Message <> Reply to Message

Hi,

Found my mistake, and fixed the blank screen problem (ftp problem).

Anyway, I did the following:

1. created a new class (sources_destination) derived from the original (sources) class.

2. added the function _cm_changeConfig to this new class:
Quote:
```
function _cm_changeConfig ($where, $fieldarray)
  {
      $this->fieldspec['publish_key'] = array('type' => 'string',
                              'size' => 1024,
                              'required' => 'y',
                              'control' => 'popup',
                              'task_id' => 'destination_list(popup)',
                              'foreign_field' => 'Collection_destination');
    return $fieldarray;


  }
```

3. In my filelist(multi4).php, I use (sources_destination) as the outer table. Should this stay as sources?

I did all this first, to ensure that after I pick a source, and select multi4, that I get the screen as described with the extra field displayed in the outer. What I get though, is just the list of files! I think I got myself a little tangled...

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 05 Jul 2007 15:30:54 GMT
View Forum Message <> Reply to Message

If you have changed your 'filelist(multi4).php' script to use 'sources_destination' as the outer table instead of 'source' then have you also changed your screen structure file so that the entry for the outer table also reads 'sources_destination'? If you have not then the XSL sylesheet is still looking for data with the 'source' tablename, and as it cannot find anything with that name it cannot display anything, thus leaving the whole area blank.

Subject: Re: Update 4
Posted by danpoleary on Thu, 05 Jul 2007 16:03:46 GMT
View Forum Message <> Reply to Message

OOps,

New I missed something, thanks. Now that I got the outer back, I still do not get the new field that I added to the _cm_changeConfig function. Should I add a line like the following to the filelist.multi4.screen.inc file:

$structure['outer']['fields'][] = array('publish_key' => 'Publish Key');

I tried that, and still the field does not show up in the outer.

I tried to match up what you said about the function above, with the notes in the popup faq. So far no luck.

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 05 Jul 2007 16:35:49 GMT
View Forum Message <> Reply to Message

In order to display a field with a particular HTML control you must have an entry in $this->fieldspec which identifies which control to use. You must also have an entry in the screen structure file which identifies where that field is to be displayed. You must also have an entry in $fieldarray, even if it is blank, otherwise the XSL stylesheet will ignore that entry. Remember that the names must be identical.

Subject: Re: Update 4
Posted by danpoleary on Thu, 05 Jul 2007 16:48:33 GMT
View Forum Message <> Reply to Message

I got the first two already, in my sources_destination.class.inc, I have:

Quote:
```
function _cm_changeConfig ($where, $fieldarray)
  {
      $this->fieldspec['publish_key'] = array('type' => 'string',
                              'size' => 1024,
                              'required' => 'y',
                              'control' => 'popup',
                              'task_id' => 'collection_list(popup1)',
                              'foreign_field' => 'publish_key');
    return $fieldarray;
```

}

In the filelist.multi4.screen.inc, I have:

Quote:$structure['outer']['fields'][] = array('source_key' => 'Source Key');
$structure['outer']['fields'][] = array('source_url' => 'Source Url');
$structure['outer']['fields'][] = array('contact' => 'Contact');
$structure['outer']['fields'][] = array('phone' => 'Phone');
$structure['outer']['fields'][] = array('email' => 'Email');
$structure['outer']['fields'][] = array('region' => 'Region');
$structure['outer']['fields'][] = array('source_root' => 'Source Root');
$structure['outer']['fields'][] = array('parent_source_key' => 'Parent Source Key');
$structure['outer']['fields'][] = array('publish_key' => 'Publish Key');

But where does the $fieldarray come into play?

I looked through the code, and the FAQ's, and am not sure where this should be applied.

Subject: Re: Update 4
Posted by AJM on Thu, 05 Jul 2007 17:12:43 GMT
View Forum Message <> Reply to Message

$fieldarray (or $this->fieldarray) is the array of data which is exported to the XML file which is used by the XSL stylesheet to build the HTML document. It displays the values for named fields in specified places, so if there is nothing in $fieldarray for a field called 'publish_key' then it will be ignored by the XSL stylesheet.

The contents of $fieldarray is initially populated by a call to getData() - except for input screens where getInitialData() is used instead - so you need to insert a value for 'publish_key' in _cm_post_getData() so that the XSL stylesheet has something to work with.

Subject: Re: Update 4
Posted by danpoleary on Thu, 05 Jul 2007 20:20:23 GMT
View Forum Message <> Reply to Message

Yes,

Got it, I forgot to add the $fieldarray['publish_key'] = ''; into the _cm_changeConfig function. Once I added that, it works great!

Now to clean up, turn off edit on the other outer fields, and add the button to call update4.

Thanks a million for pointing me in the right direction.

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 05 Jul 2007 20:59:12 GMT
View Forum Message <> Reply to Message

No problem. Glad you got it sorted out.

Subject: Re: Update 4
Posted by danpoleary on Fri, 06 Jul 2007 15:56:40 GMT
View Forum Message <> Reply to Message

Hi, Thanks again,

I now have two of the update4 buttons running. The last one is where I am now stuck.

When you say above, to also add the destination field to $this->fieldspec of the "files" class, how is that done? I obviously do not need a new dropdown or popup, I just need the value selected passed to the update4 script.

You then mention _cm_getPkeyNames(). In various examples, I have not found how to pass the value into the update4 script using this function. Do you declare this function in the inner class of the multi4, or in the extended class used as part of the update4?

Thanks,

Dan

Subject: Re: Update 4
Posted by danpoleary on Fri, 06 Jul 2007 16:51:12 GMT
View Forum Message <> Reply to Message

Hi Tony,

This is what I have tried.

in my multi4, I have the outer class called "sources_destination" which extends sources class, and

includes the _cm_changeConfig function that sets up the foreign field 'sel_publish_key'. This works, when I select the sources key, and then the multi4 button, I get the outer of "sources" with the 'sel_publish_key' from the destination table displayed as a dropdown. I can select the destination key.

I then added the _cm_getPkeyNames function to the "sources_destination" class. The function contains:

```
$pkey_array[] = 'sel_publish_key';
 return $pkey_array;
```

Is this correct, I do not see the field in the class used in the update4.

Dan

Subject: Re: Update 4
Posted by AJM on Fri, 06 Jul 2007 17:12:17 GMT
View Forum Message <> Reply to Message

The MULTI4 pattern has an outer (top) entity which shows a single occurrence and an inner (bottom) entity which shows multiple occurrences. You have a popup in the outer entity which supplies a value for 'sel_publish_key'. When you select occurences of the inner entity and press the navigation button to activate the UPDATE4 function you want the value for 'sel_publish_key' included in the selection string which is passed to that function.

As the selection string will be extracted from the inner entity you need to do the following:

include 'sel_publish_key' in the $fieldspec array of the inner entity so that the value selected in the outer entity will also appear in the POST array for the inner entity.
add 'sel_publish_key' to the pkey array in the _cm_getPkeyNames() method within the inner entity so that it can be included in the selection string. Without this only those fields in the primary key will be used.

Subject: Re: Update 4
Posted by danpoleary on Mon, 09 Jul 2007 15:42:08 GMT
View Forum Message <> Reply to Message

Hi Tony,

I have the following in the inner class:

Quote:
```
 function _cm_changeConfig ($where, $fieldarray)
  {
```

```
  $fieldarray['sel_publish_key'] = '';
  $this->fieldspec['sel_publish_key'] = array('type' => 'string',
                                              'size' => 32);
   return $fieldarray;
 }

   function _cm_pre_updateRecord($rowdata)
  {
        // Dan O'Leary, for task specific actions.
        // modify the rowdata boolean fields publish,autopublish and the publish_key
        //$rowdata['publish'] = TRUE;
        $rowdata['autopublish'] = TRUE;

        // set the actual publish_key value to the value of the sel_publish_key
        $rowdata['publish_key'] = $rowdata['sel_publish_key'];

        return $rowdata;
  }

  function _cm_getInitialData ($fieldarray)
  // Perform custom processing for the getInitialData method.
  // $fieldarray contains data from the initial $where clause.
  {

     $fieldarray['sel_publish_key'] = $_SESSION['san_filelist(multi4)']['sel_publish_key'];

     return $fieldarray;

  } // _cm_getInitialData
```

But I still cannot see the sel_publish_key from the outer class.

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Mon, 09 Jul 2007 16:14:43 GMT
View Forum Message <> Reply to Message

What is this line?

$fieldarray['sel_publish_key'] = $_SESSION['san_filelist(multi4)']['sel_publish_key'];

I have never suggested that you use anything like this.

The contents of 'sel_publish_key' in the outer entity will not appear in the inner entity until AFTER the navigation button has been pressed. The contents of the $_POST array will then contain a single occurrence from the outer entity, plus one or more occurrences from the inner entity. A field in the $_POST array for the outer entity will not be merged with the data for the inner entity UNLESS a field with that name is found in the inner entity's $fieldspec array.

In order to get the value for 'sel_publish_key' included in the selection string which is passed from the inner entity to the UPDATE4 function you will need to add 'sel_publish_key' to $pkey_array in the _cm_getPkeyNames() method of the inner entity.

Subject: Re: Update 4
Posted by danpoleary on Mon, 09 Jul 2007 19:16:14 GMT
View Forum Message <> Reply to Message

Hi Tony,

Sorry, I based that call on something on say in the existing code.

I now have the following in the inner entity (filelist_auto.class.inc file) which extends the filelist.class.inc. This extended class is used in both the multi4, and in the update4.

Quote:
```
    function _cm_pre_updateRecord($rowdata)
  {
        $rowdata['autopublish'] = TRUE;

        // what was select in the popup during _cm_initialize
        $rowdata['publish_key'] = $rowdata['sel_publish_key'];
        return $rowdata;
  }

 function _cm_changeConfig ($where, $fieldarray)
  {
   $fieldarray['sel_publish_key'] = '';
   $this->fieldspec['sel_publish_key'] = array('type' => 'string',
                                               'size' => 32);
    return $fieldarray;

  }

   function _cm_getInitialData ($fieldarray)
   // Perform custom processing for the getInitialData method.
   // $fieldarray contains data from the initial $where clause.
   {
      $fieldarray['sel_publish_key'] = $_POST['sel_publish_key'];

      return $fieldarray;
```

```
} // _cm_getInitialData

function _cm_getPkeyNames ($pkey, $task_id, $pattern_id)
{
    $pkey_array[] = 'sel_publish_key';
    return $pkey_array;
}
```

Where else could I try? I can see the post of the sel_publish_key, but I am not sure in which function to set it!

I have looked for many examples, but have yet to find one that works in this multi4, update4 fashion.

Thanks,

dan

Subject: Re: Update 4
Posted by AJM on Mon, 09 Jul 2007 20:11:36 GMT
View Forum Message <> Reply to Message

Step through with your debugger to see what happens when you press the navigation button. You should see the contents of the $_POST array being merged wth both the outer and inner entities, then the $selection string is built using the details of the selected entries before being passed to the UPDATE4 function.

Subject: Re: Update 4
Posted by danpoleary on Tue, 10 Jul 2007 00:21:05 GMT
View Forum Message <> Reply to Message

Hmmm,

Debug would be great, but our firewalls prevent outbound connections to the komodo running on my system.

I will try to move the whole environment to a local system.

In the meantime, does the code I have look ok so far for the inner entity?

Thanks,

dan

Subject: Re: Update 4
Posted by AJM on Tue, 10 Jul 2007 08:58:28 GMT
View Forum Message <> Reply to Message

No. The code in _cm_getInitialData() is redundant as you do not access the $_POST array directly in that manner. The contents of the $_POST array will be inserted into the inner entity's $fieldarray by code that is in the MULTI4 controller - that is what the change I made to the array_update_indexed() function is for.

Subject: Re: Update 4
Posted by danpoleary on Tue, 10 Jul 2007 16:16:59 GMT
View Forum Message <> Reply to Message

I did some debugging, and the _cm_getPkeyNames method inside my inner entity in the multi4 is never called.

For extra details, I wrote out the contents of the various attributes in the custom methods, and none contain the sel_publish_key other than the _cm_changeConfig methods $fieldarray after I add it.

When looking at my code above for the inner entity, is that not the correct way of adding the sel_publish_key to the inner? and is my _cm_getPkeyNames method not correct? (I did change the attribute name from $pkey to $pkey_array)

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Tue, 10 Jul 2007 17:54:19 GMT
View Forum Message <> Reply to Message

I have just tried modifying a MULTI4 transaction to reproduce the solution I have described, and it works exactly as I expected. I tested it using Radicore version 1.25.0 plus the updated version of 'include.general.inc' which I attached to an earlier post in this thread.

To make sure you are not missing something simple this is what I did:

- the outer entity of the MULTI4 function contains a field called 'sel_publish_key' which appears on the screen and is amendable by the user. This means that when a button is pressed the $_POST array will contain the current value for 'sel_publish_key'.

- when I select entries from the inner entity of the MULTI4 function and press a navigation button,

the contents of the $_POST array is then merged with the contents of both the outer and inner entities before the button is processed.

- the inner entity of the MULTI4 function does not contain a field called 'sel_publish_key', so its value will not, by default, be added to $fieldarray within the inner entity. I can change this behaviour by adding the following code to the inner entity:

```
function _cm_changeConfig ($where, $fieldarray)
{
   $this->fieldspec['sel_publish_key'] = array('type' => 'string', 'nondb' => 'y');

   return $fieldarray;

} // _cm_changeConfig
```

- when the navigation button is pressed the following lines in 'std.multi4.inc' will be executed:

```
116:    $inner_post = getPostArray($_POST, $dbinner->getFieldSpec());
117:    $inner_data = array_update_indexed($inner_data, $inner_post);
```

You need to inspect this with your debugger to ensure that $_POST contains a value for 'sel_publish_key', and that $inner_data is updated so that every row now contains the value for 'sel_publish_key'.

- the next line in 'std.multi4.inc' calls the childform() function, which will pass control to the form associated with the navigation button which has been pressed. The processing should go through the code appearing at line 179:

```
   if (isset($post['select'])) {
      // convert selection into SQL where format
      $pkey_array = $dbobject->getPkeyArray(null, $task_array);
      $selection = selection2where($pkey_array, $post['select']);
   } else {
```

The call to getPeyArray() contains the call to _cm_getPkeyNames() which should contain the following:

```
function _cm_getPkeyNames ($pkey_array, $task_id, $pattern_id)
{
   $pkey_array[] = 'sel_publish_key';

   return $pkey_array;

} // _cm_getPkeyNames
```

This ensures that when the $selection string is built it will contain the value for 'sel_publish_key'.

This string will then be passed to whatever function was activated by the navigation button.

All I had to do to get this to work was include 'sel_publish_key' in the _cm_changeConfig() and _cm_getPkeyNames() methods of the inner entity, so unless you are doing something really peculiar it should be just as easy for you.

Subject: Re: Update 4
Posted by danpoleary on Wed, 11 Jul 2007 15:15:42 GMT
View Forum Message <> Reply to Message

Hi Tony,

Thanks,

I rebuilt the whole thing. The following describes what I have right now.

In the multi4 , I access an extended outer class of sources_destination which extends sources, by adding the sel_publish_key field, and allows me to select a value.

for the inner class (filelist), I have exactly what you have above, and that part works well, except my debugger will not allow me to see the contents of $inner_data (komodo problem)

$_POST does contain the value I selected for that key.

When the update4 is processes, I can see the value being passed in, but I get this error:
Quote:2007-07-11 11:11:15

Fatal Error: MySQL error: 1054 - Unknown column 'sel_publish_key' in 'where clause' (# 1054).

SQL query: SELECT SQL_CALC_FOUND_ROWS * FROM filelist WHERE (unique_sequence='1323' AND sel_publish_key='IFC') ORDER BY filename asc

Error in line 407 of file 'C:\apache\htdocs\search\includes\dml.mysqli.class.inc'.

Host Info: localhost via TCP/IP
Server Version: 5.0.27-community-nt
Client Info: 5.0.22, Client Encoding: latin1

Script: /search/search/filelist_autopublish(upd4).php

User Id: MGR

Remote Address: 127.0.0.1

Request URI: /search/search/filelist_autopublish(upd4).php?session_name=m enu2

So how do I get rid of the sel_publish_key from within the select statement of the filelist for this method?

Thanks, I am much closer, not sure why it did not work before, but it may have been because I was 4 versions behind of Radicore.

Dan

Subject: Re: Update 4
Posted by AJM on Wed, 11 Jul 2007 16:08:39 GMT
View Forum Message <> Reply to Message

This means that the table class you are using in the UPDATE4 function contains a reference to 'sel_publish_key' in its $fieldspec array, and this reference must have been added manually in that class.

If you remove this reference then 'sel_publish_key' will automatically be removed from the $where string during the processing of the initialise() method.

Presuming that you want to use the value for 'sel_publish_key' in the updateMultiple() method but not the getData() method you will have to save the value as a member variable and remove it from the $where clause before $DML->getData() is performed. You can do this either in the _cm_initialise() method or the _cm_pre_getData() method. Here is an example:

```
function _cm_pre_getData ($where, $where_array, $fieldarray=null)
{
    $this->sel_publish_key = $where_array['sel_publish_key'];
    unset($where_array['sel_publish_key']);
    $where = array2where($where_array);

    return $where;

} // _cm_pre_getData
```

Subject: Re: Update 4
Posted by danpoleary on Wed, 11 Jul 2007 16:32:48 GMT
View Forum Message <> Reply to Message

Yes, success,

Git it, thank you. It appears my root problem in trying to figure it out was that I was to many versions back in Radicore. Once I solved that, things started to fall into place.

One last question, When I select one record from the inner, the update works fine, but when I select multiple records, only the first one gets updated.

I checked the sql, and it only has the first records unique_sequence value in the 'where'. Should the update4, not pass all selections into the POST?

Dan

Subject: Re: Update 4
Posted by danpoleary on Wed, 11 Jul 2007 16:48:33 GMT
View Forum Message <> Reply to Message

Hmmm,

May have found it myself.

I used _cm_pre_updateRecord to set the values as requested. Should I have used _cm_pre_updateMultiple? I thought that updateRecord is called for each selected record.

Dan

Subject: Re: Update 4
Posted by AJM on Wed, 11 Jul 2007 17:34:26 GMT
View Forum Message <> Reply to Message

The _cm_pre_updateMultiple() method is called just once so that you can manipulate ALL the occurrences in $fieldarray. Each occurrence is then passed to the updateRecord() method, so the _cm_pre_updateRecord() method is called separately for each occurrence.

It is up to you to decide which is the most appropriate place to put your code.

Are you sure that the selection string being passed from the MULTI4 to the UPDATE4 contains references to all the records which you selected? It should be in the format (selection1='..') OR (selection2='..') OR (selection3='..') If the string coming out of the MULTI4 is wrong then there is something in the MULTI4 which needs looking at. If the string coming out of the MULTI4 is correct then there is something in the UPDATE4 which is wrong.

Subject: Re: Update 4
Posted by danpoleary on Wed, 11 Jul 2007 17:35:13 GMT
View Forum Message <> Reply to Message

Strange,

Only the first record is being sent to update4. What would prevent more than one record from

being passed in?

Dan

Subject: Re: Update 4
Posted by AJM on Wed, 11 Jul 2007 17:46:02 GMT
View Forum Message <> Reply to Message

I have just tried multiple selections with my MULTI4, and the selection string which is created contains those multiple selections. You must have something wrong in your code.

Step through with your debugger, particulary at lines 180 and 181 in the childform() function in file 'include.session.inc' as this is where the contents of $fieldarray is turned into a string based on which rows were selected.

Subject: Re: Update 4
Posted by danpoleary on Wed, 11 Jul 2007 18:48:42 GMT
View Forum Message <> Reply to Message

Yes, I had confirmed that before, and just now, and all records selected exist in the debugger at that location, but when it enters update4, only the first record exists. I checked inside the _cm_pre_getData of the class used in update4.

dan

Subject: Re: Update 4
Posted by AJM on Wed, 11 Jul 2007 19:23:49 GMT
View Forum Message <> Reply to Message

You need to step through every part of the UPDATE4 to see where the $where string is being modified as it passes through several methods before it gets to _cm_pre_getData().

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 12:44:47 GMT
View Forum Message <> Reply to Message

Good morning Tony,

I think I found it, but a I am not sure why it is this way.

when the call is made to my _cm_pre_getData, the $where contains all the selections, but the $where_array only contains the first record. In your sample above, you stripped the

sel_publish_key from the $where_array. Should this have been stripped from the $where instead? or, does a problem exists in whic the $where_array is not populated correctly?

Thanks,

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 12 Jul 2007 13:28:45 GMT
View Forum Message <> Reply to Message

That is by design, actually. The $where_array in _cm_pre_getData() only shows values from the first selected row so that it is easy to determine which field names have been specified, and there is no need to duplicate the same field name for every row as they are all the same (just with different values).

In your case it would be better to extract the value for 'sel_publish_key' in the _cm_initialise() method as it will be filtered out of the $where string in the code which follows (provided that 'sel_publish_key' has not been manually inserted into the $fieldspec array). You can do it with code such as:

```
function _cm_initialise ($where)
{
    $where_array = where2array($where);
    $this->sel_publish_key = $where_array['sel_publish_key'];

    return $where;

} // _cm_initialise
```

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 14:13:25 GMT
View Forum Message <> Reply to Message

Hmmm,

I changed the code, but now something strange happens. When the call is made to _cm_initialise, the where does does contain everything, but the call to where2array, only returns the first occurrence, also, the where still contains the sel_publish_key in subsequent methods.

Dan

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 14:50:06 GMT
View Forum Message <> Reply to Message

Just as an update, I validated the fieldspec array inside the update4 methods, and it does not contain the sel_publish_key field.

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 12 Jul 2007 14:51:00 GMT
View Forum Message <> Reply to Message

Surely you only need the first occurrence as the value for 'sel_publish_key' is the same for every occurrence? There is only a single value available on the previous MULTI4 function.

The reason that 'sel_publish_key' is not being filtered out of the $where string is that it is contained within the $fieldspec array, presumably because of code within the _cm_changeConfig() method. This addition is necessary in the MULTI4 function, but not in the UPDATE4 function.

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 14:59:32 GMT
View Forum Message <> Reply to Message

As per my note above, I have validated that the update4 use of the filelist class does not modify the fieldspec array, by adding a check against getSelf() to ensure that the addition of sel_publish_key is only applied if the call current script is "filelist(multi4).php'.

That is why I am confused a bit, since the where still contains this field.

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 12 Jul 2007 15:06:48 GMT
View Forum Message <> Reply to Message

The $where string will not be filtered to remove 'sel_publish_key' until AFTER the call to _cm_initialise(). Take a look inside the initalise() method at the filterWhere() function.

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 15:34:17 GMT

View Forum Message <> Reply to Message

The following condition passes even though it should not when $fieldname contains sel_publish_key:

if (ereg('\b', $fieldname) OR array_key_exists($fieldname, $fieldlist) AND !isset($fieldlist[$fieldname]['nondb'])) {

fieldlist does not contain sel_publish_key.

Why would it pass this condition?

Subject: Re: Update 4
Posted by danpoleary on Thu, 12 Jul 2007 15:53:23 GMT
View Forum Message <> Reply to Message

I think I found the problem, the '\b' against ereg, looks for beginning or end of word, and will always evaluate to true. Was this your intent? or were you looking for digits instead?

Dan

Subject: Re: Update 4
Posted by AJM on Thu, 12 Jul 2007 18:59:16 GMT
View Forum Message <> Reply to Message

If you look at the comment before that line you will see that I am looking for multiple words, and '\b' is supposed to test for a word boundary. It returns false with a fieldname of 'foobar' but true if it contains '_' (underscore) as in 'sel_publish_key'. If you change ereg('\b', $fieldname) to preg_match('/\w+ \w+/', $fieldname) you should find that it solves the problem.

Subject: Re: Update 4
Posted by danpoleary on Fri, 13 Jul 2007 15:37:11 GMT
View Forum Message <> Reply to Message

Thanks,

Everything works great now. I can select records and it applies the update to them. One thing I have noticed however, is that if I do not select anything in the inner entity on the multi4 screen, but still select the auto_publish button (update4), it will update all the records in the inner entity by setting the autopublish boolean to true, but ignore the outer selected sel_publish_key.

Why would it update records if none have been selected? and if this is normal, how can I validate that the records have been selected inside the method auto_publish or its extended class?

I am stepping through everything, but any pointers on where to aim my search would be greatly appreciated.

Thanks for all your help
dan

Subject: Re: Update 4
Posted by AJM on Fri, 13 Jul 2007 16:11:55 GMT
View Forum Message <> Reply to Message

When you press a navigation button without selecting any entries the framework will create a selection string using the $where cause which went into the database object. In this case it will be the inner entity of the MULTI4. Because this selection string is not empty it will not be rejected by the UPDATE4 controller, so you will need code in your database object to inspect the $where string for validity before being allowed to continue.

The best place to do this would be the _cm_initialise method. If the $where string is invalid simply put an error message into $this->errors[] and the UPDATE4 controller will terminate and return to the MULTI4.

Subject: Re: Update 4
Posted by danpoleary on Fri, 13 Jul 2007 17:47:22 GMT
View Forum Message <> Reply to Message

Thanks,

Everything work excellent.

Thanks for all your help,

Dan
