

---

Subject: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 10 Aug 2009 11:39:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I have a bug that seems to have been introduced in a recent upgrade (it doesn't occur in my production environment running 1.48 but does occur in my dev environment running 1.50). I'm struggling to track where it is going wrong in debug - hoping you will be able to help me track it down...

I have a table of items and descriptions. I have a second table containing rdcaccount\_id and the item\_id from the first table, and amounts. So a user can specify their own amounts for the items in the master table.

Updating of the amounts works perfectly, and the framework handles the rdcaccount\_id automatically.

I also have two output2 tasks to print reports of the charges

- 1) Current amounts only (by setting curr/hist = 'C' in fixed selection)
- 2) All amounts (leave fixed selection blank)

For some reason I have to pass the rdcaccount\_id to these tasks manually using `_cm_getWhere()` to ensure I only get the relevant rows (is this a bug also?) - the framework is not providing it automatically. In 1.48 this approach works OK.

However, in 1.50 only the 'current amounts' report is working correctly. The 'all amounts' report is displaying amounts for all rdcaccount\_id (seems to be being filtered out somewhere, but I can't find where). The only difference between the tasks is the presence of the fixed selection criteria.

Can you help?

Thanks,

Graham

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 10 Aug 2009 14:49:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

How are you activating the OUTPUT2 task? Is it from a navigation button? If so, does the table in that parent task also contain a column called rdcaccount\_id?

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 10 Aug 2009 15:29:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Yes I am activating via a navigation button from a LIST2 transaction (there are two navigation buttons - one for each report).

The list2 transaction has the parent table that does NOT contain rdcaccount\_id, and the child table DOES contain rdcaccount\_id. Thus each user sees all the data from the parent table, but only their own rows from the child table. It works fine for the administration of the data, but not for the output2 tasks.

Graham

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 10 Aug 2009 15:33:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

It would appear that the getData\_serial() method (as used in OUTPUT tasks) did not contain the code to set the value of rdcaccount\_id. This is fixed in the attached file.

#### File Attachments

---

1) [std.table.class.zip](#), downloaded 1871 times

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 10 Aug 2009 15:57:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Thanks - that's confirmed working now.

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 10:57:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Tony, with further more detailed testing I've found that this isn't completely fixed under certain conditions.

If I run the OUTPUT2 task without making any specific selections (i.e. a complete report) the following SQL is generated:

```
SELECT cost_group_charge.*, cost_group.cost_group_desc, school.school_name
FROM cost_group_charge
LEFT JOIN cost_group ON (cost_group.cost_group_id=cost_group_charge.cost_group_id)
LEFT JOIN school ON (school.school_id=cost_group_charge.rdcaccount_id)
WHERE cost_group_charge.rdcaccount_id='7'
ORDER BY cost_group_charge.age_group_id, cost_group_seq_nbr,
cost_group_charge.banner_req =>Count=86
```

However, if I make a couple of selections before running the report the SQL is:

```
SELECT cost_group_charge.*, cost_group.cost_group_desc, school.school_name
FROM cost_group_charge
LEFT JOIN cost_group ON (cost_group.cost_group_id=cost_group_charge.cost_group_id)
LEFT JOIN school ON (school.school_id=cost_group_charge.rdcaccount_id)
WHERE ( cost_group_charge.cost_group_id='SDA2' ) OR (
cost_group_charge.cost_group_id='SDA3' ) AND cost_group_charge.rdcaccount_id IN ('1', '7')
ORDER BY cost_group_charge.age_group_id, cost_group_seq_nbr,
cost_group_charge.banner_req =>Count=6
```

This actually selects from other rdcaccount\_id rows also. It needs the selections that are grouped with the OR statement to be bracketed together.

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 12:49:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

If the 'cost\_group' table contains rdcaccount\_id, and rdcaccount\_id is part of the primary key, are you including rdcaccount\_id in the select list of the inner entity of the LIST2 task?

When you mark rows as being selected in a LIST task it will extract the primary keys of the selected rows and place them in the \$selection string, which will be used as the \$where string in any child task. If any part of the primary key is not available in \$this->fieldarray then it cannot be extracted, which means that the \$selection string will not correctly identify the records which you selected.

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 13:20:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Well you are correct in that this is the root cause of the problem because the table cost\_group does not contain rdcaccount\_id and so this is not specified in the \$selection string.

In the original post in this thread I pointed out that I have linked two tables, only ONE of which contains rdcaccount\_id.

The functionality I have tried to create is to have a master set of items (using table cost\_group not containing rdcaccount\_id). Each account owner can then create their own scale of charges for these items (using table cost\_group\_charge containing rdcaccount\_id) and related to table cost\_group using cost\_group\_id.

The framework is handling it perfectly apart from the reporting.

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 13:36:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

As a work around to this it would be OK for me to ignore any selections made by the user when producing the report (i.e. I would always produce a full report rather than a selective report). Is there any way I can bypass/ignore any selections the user has made? (e.g. set \$selection to null in an appropriate method?)

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 13:39:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

What do you have as the \$outer and \$inner entities of your LIST2 task? Does the OUTPUT2 task which is activated from a navigation button work with the \$inner entity?

If the \$inner entity contains rdcaccount\_id in its primary key, is this included in the select list so that it can be extracted and put into the \$selection string?

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 13:55:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I actually run the output2 task from a list1 transaction based on the cost\_group table.

The user can set their charges by selecting cost\_groups and entering a list2 task:

The outer table is cost\_group (key cost\_group\_id)

The inner table is cost\_group\_charge (key rdcaccount\_id and cost\_group\_id + a couple of others)

The intention was for users to be able to print a full scale of charges for all cost groups by running the output2 task from the list1 without making any selections, and this works fine.

However, I can't prevent a user making a selection in the list1 screen before clicking the output2 button, and this is when it goes wrong.

Because of the unusual scenario linking tables, one without rdcaccount\_id, maybe it isn't worth the effort of getting the framework to handle this automatically. However, I need to be able to code to prevent the wrong data being presented to the users, hence the suggestion in my latest post.

Further to this, the output2 task is based on the inner table, but the button is placed on the list1 screen of the outer table.

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 15:09:17 GMT

This is where you are confusing me. In an earlier post you said that you were activating the OUTPUT2 task from a LIST2 task (where the \$inner entity contains rdcaccount\_id), but now you are saying that this problem occurs when you are activating it from a LIST1 task (where the \$main entity does NOT contain rdcaccount\_id). This also means that you are using a navigation button in a parent task which works on a different table than the one in the child task, which in this case also means that the foreign key which is passed from the parent to the child is incomplete as it is missing rdcaccount\_id.

I have played around with my code and have managed to get it to produce a WHERE string in the format:

```
(( cost_group_id='SDA2' ) OR (cost_group_id='SDA3' )) AND rdcaccount_id IN ('1', '7')
```

Try the attached file and see if it solves your problem.

### File Attachments

---

1) [std.table.class.zip](#), downloaded 1875 times

---

---

**Subject: Re: Losing \$where when fixed selection is not present**

Posted by [gpatti](#) on Mon, 17 Aug 2009 15:37:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I'm really sorry for the confusion. I wasn't being careful when I said earlier it was being called from a LIST2 transaction.

And you are quite right about the foreign key. I wrote this code months ago and when I check I have taken rdcaccount\_id from \$\_SESSION and added rdcaccount\_id to \_cm\_filterWhere().

Your alterations to bracket the OR statements have indeed solved the problem - so thank you for that.

I'd still like to ask about trapping and modifying selections made by a user: in another situation I have a navigation button on a LIST2 transaction that calls another LIST2 transaction with exactly the same parent and child tables (this is simply so that I can split different functionality over different screens). I've noticed that if a user makes a selection before pressing the navigation button I get an error because the primary key is missing in the receiving task (presumably because the \$selection takes precedence over the \$where). However, in this case I don't want the \$selection to be used anyway - I want it to be ignored. If there was a way of doing this I would be able to prevent the missing key error (which confuses the user as the solution is to remove the selections they made). Hope this is explained clearly enough - let me know if not.

Graham

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 17:15:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

You like being awkward, don't you!

Attached is a new version of std.list2.inc in which I have cut out the overwriting of \$where with \$selection. Both of these values will be passed to initialise() and therefore \_cm\_initialise(). I have changed the logic inside initialise() as follows:

```
$where2 = $this->_cm_initialise($where, $selection, $search);
if ($where2 != $where) {
    // this was changed in _cm_initialise(), so use the new version
    $where = $where2;
} else {
    if (!empty($selection)) {
        // $selection takes precedence over $where
        $where = $selection;
        $selection = null;
    } // if
} // if
```

If \_cm\_initialise() changes the contents of \$where then \$selection will be ignored.  
If \$where is not changed and \$selection is not empty then \$where will be overwritten by \$selection.

Note that the argument list for \_cm\_initialise() has been changed to the following:

```
function _cm_initialise ($where, &$selection, $search)
// perform any initialisation for the current task.
// NOTE: $selection is passed by reference as it may be amended.
// NOTE: $search is only available for OUTPUT tasks.
{
    // customisable code goes here

    return $where;
} // _cm_initialise
```

\$selection is passed by reference so that you can change its contents. If you erase it then it cannot be used to overwrite the contents of \$where.

Try the attached file to see if it works OK.

### File Attachments

---

1) [std.list2.zip](#), downloaded 1908 times

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 17:59:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

That's me all over

This is going to take quite some testing which I will get on with. However, first thing I notice is that \$where and \$selection aren't as I expected when I look in `_cm_initialise()`. I need a bit of help understanding what's going on.

If I call the second list2 transaction without making any selections I see that \$where contains the primary key passed over, and \$selection is NULL. Perfect sense.

However, if I make a selection and then call the list2 transaction I now see that \$where contains the selections I made, and \$selection is NULL. Not what I was expecting!

[Will be out for a few hours now, but will do some more testing on my return]

Graham

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 19:15:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I am not seeing this at all in my testing. When the 2nd LIST2 is activated the \$where string contains the primary key of the \$outer entity from the 1st LIST2 and the \$selection string contains whatever has been selected from the \$inner entity (may be empty). These are the exact values which are made available to `_cm_initialise()` without being changed at all.

Are you sure you are using the new versions of both `std.table.class.inc` and `std.list2.inc` which I supplied you?

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Mon, 17 Aug 2009 21:26:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I've been scratching my head over this one and I've finally realised why I'm not seeing the same results as you...

The second task is not a LIST2 after all: it's a MULTI2 !!

Really sorry!

---

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 22:38:16 GMT

---

[View Forum Message](#) <> [Reply to Message](#)

---

That would explain it. Try the attached file.

### File Attachments

1) [std.multi2.zip](#), downloaded 1985 times

---

---

**Subject: Re: Losing \$where when fixed selection is not present**

Posted by [gpatti](#) on Tue, 18 Aug 2009 08:58:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

OK. \$where and \$selection now show expected results:

1. Without making selections \$where and \$selection both hold the primary key
2. With selections \$where holds the primary key and \$selection holds the selections made

However, if I make selections and then set \$selection = \$where in `_cm_initialise` (to match scenario 1) I still get the error message 'Primary key (primary\_key\_id) is not complete - check selection'.

Do I need to do something more complex than simply setting \$selection?

A solution I can use if the approach I was taking is not valid is to trap that selections have been made and display a custom error telling the user not to make selections. This works OK.

---

---

**Subject: Re: Losing \$where when fixed selection is not present**

Posted by [AJM](#) on Tue, 18 Aug 2009 09:17:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

If you don't want to use \$selection then why not simply set it to null?

---

---

**Subject: Re: Losing \$where when fixed selection is not present**

Posted by [gpatti](#) on Tue, 18 Aug 2009 09:32:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I did try that and still got the same result - the 'Primary key not complete' message is displayed. I was trying to prevent that.

---

---

**Subject: Re: Losing \$where when fixed selection is not present**

Posted by [AJM](#) on Tue, 18 Aug 2009 10:24:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

---



I do not understand. You said in message #2280 that \$where always contains the primary key whether a selection has been made or not, so there is no reason for the error message "Primary Key is not complete". I cannot reproduce this behaviour at all - it always behaves as expected.

Can you step through with your debugger to see where the primary key in \$where is being rejected?

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [gpatti](#) on Thu, 27 Aug 2009 15:24:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've now spent some time trying to debug what is happening and I think it is as follows:

I move from an outer table to an inner table using a LIST2 transaction and \$where at this point contains the primary key of the outer table row, such that all the inner table rows are selected as expected.

Now, I have two choices:

1. I press a navigation button which gives me a MULTI2 transaction based on the same inner table - having NOT made any selections. This works correctly, and \$where still contains the primary key from the outer table.
2. I press a navigation button which gives me a MULTI2 transaction based on the same inner table - having made a selection. Now I find that \$where is being replaced by \$selection at line 2467 of std.table.class.inc because \$selection is not empty. It now doesn't contain the primary key of the outer table resulting in the 'primary key not complete error'.

Maybe I need to do something to handle this?

Graham

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Thu, 27 Aug 2009 15:55:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

It is the default behaviour to replace \$where with \$selection when \$selection is not empty. In order to override this default you should place the following code in the outer/parent table of the MULTI2 task:

```
function _cm_initialise ($where, &$selection, $search)
// perform any initialisation for the current task.
// NOTE: $selection is passed by reference as it may be amended.
// NOTE: $search is only available for OUTPUT tasks.
{
```

```
$selection = null;  
  
return $where;  
  
} // _cm_initialise
```

This will then prevent \$where from being replaced with the contents of \$selection.

---

---

Subject: Re: Losing \$where when fixed selection is not present  
Posted by [gpatti](#) on Thu, 27 Aug 2009 16:23:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Perfect!

Problem solved. Thanks for the help.

---