

## Bugs - Bug #1576

### Moving ACL items up and down faulty in admin applet

10/05/2012 11:20 AM - Ovidiu Maxiniuc

<b>Status:</b>	WIP	<b>Start date:</b>	10/05/2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Ovidiu Maxiniuc	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No	<b>version:</b>	
<b>vendor_id:</b>	GCD		
<b>Description</b>			

#### History

##### #1 - 10/05/2012 11:21 AM - Ovidiu Maxiniuc

Moving ACL item up and down will overlap other items.

Consider the following steps:

1. Have 5 ACL items within the same resource with is: 100, 200, 300, 400 and 500. Only the last one (actual 500 is moved).
2. Move Up, it will be recomputed as id 250 (shouldn't it be 350, up one position only ?)
3. Continue moving up until it get Id 1.
4. Move Down, it will get Id 101 (again, shouldn't it be 150, just between 1st and 2nd (now 3rd) ?)
5. Move Down, it will get Id 201
6. Move it Up, and you'll observe that the old 100 is gone and you only have 4 ACL items !
7. Move it Down until it gets back to id 500 and the item will overwrite each of the other items.

If you look into the directory.xml, you can notice that when the overwriting occurs there are in fact 2 items with the same ids.

##### #2 - 10/05/2012 11:31 AM - Ovidiu Maxiniuc

One possible fix for this issue is modifying the `com.goldencode.p2j.admin.client.AdminACL.nextId(boolean up, boolean current)` method that is called for both up and down actions and computes the new id of the ACL item based on the other items and more specifically to the immediate neighbor. However, in boundary case, one of: `sid / 2` (step 2) or `sid + 100` (other steps) is used.

I believe that the boundary flag might be wrong computed to true, so instead of moving between two other items, the id will overlap.

##### #3 - 10/05/2012 11:54 AM - Ovidiu Maxiniuc

- Status changed from New to WIP

- Assignee set to Ovidiu Maxiniuc

##### #4 - 10/15/2012 09:50 AM - Ovidiu Maxiniuc

The `nextId()` computes the next item in the direction of movement taking into consideration the boundaries (if the selected item is first/last). Then it checks the found item (immediate neighbor) has the same resource type and instance as the selected item. If one is different it sets boundaries to true and act accordingly.

This is not correct. A search should be implemented for the nearest neighbor in the desired direction which has the same instance and resource; if none is found, then the new id should be computed in boundaries case.

Here is an example:

Instance	Restype	Id	...
-----	-----	---	
*shared*	system	300	...
*shared*	admin	400	...
*shared*	system	600	... <<<<<

If the current item is 3rd and you move it up, it will wrongly compute the new index as 300 as the next up neighbor has admin resource type which is different from the current system.

The same algorithm appears twice in the same method.