Morpho - Bug #5058

Access wizard mishandles certain combinations of allow/deny elements

06/24/2010 05:40 PM - Jim Regetz

Status:	Resolved	**Start date:**	06/24/2010
Priority:	Immediate	**Due date:**	
Assignee:	ben leinfelder	**% Done:**	0%
Category:	morpho - general	**Estimated time:**	0.00 hour
Target version:	1.8.1	**Spent time:**	0.00 hour
Bugzilla-Id:	5058		

Description

Morpho substrings XPaths improperly in certain cases when extracting data for the Access wizard, ultimately causing it to unnecessarily punt to the tree editor.

Here are some reproducible cases:

- If there is an 'allow' element for principal 'public', along with 10 or more 'allow' elements for (non-public) principals, Morpho incorrectly parses the 10th one.
- If there is an 'allow' element for principal 'public', Morpho incorrectly parses any 'deny' elements for (non-public) principals.
- If there is a 'deny' element for principal 'public', along with at least one 'allow' element and at least one 'deny' element for (non-public) principals, Morpho incorrectly parses those 'deny' elements.

For a live example of the first case, try to open the Documentation->Access Information interface for this data package: http://knb.ecoinformatics.org/knb/metacat/krobinson.12.6/nceas

I believe the problem lies at least partly in lines 1304-1305 of AccessPage.java (as of rev 4925). For the cases above, this.xPathRoot.length() isn't actually the right value to pass to the substring statement. Depending on the case, the trimmed XPath ends up either having an extra character at the start, or having its first character truncated.

From stderr.log, here's an example of the first case above (note the extra ']' as the first character of the trimmed nextXPaths):

```
...
Access:  nextXPath = /access/allow[10]/principal[1]
nextVal   = user10
Access: TRIMMED nextXPath   = ]/principal[1]
Access:  nextXPath = /access/allow[10]/permission[1]
nextVal   = read
Access: TRIMMED nextXPath   = ]/permission[1]
...
```

And here's an example of the second case (note the missing first letter in the trimmed nextXPaths -- actually, maybe they're missing a leading '/' too?):

```
...
Access:  nextXPath = /access/deny[1]/principal[1]
nextVal   = user09
Access: TRIMMED nextXPath   = rincipal[1]
Access:  nextXPath = /access/deny[1]/permission[1]
nextVal   = read
Access: TRIMMED nextXPath   = ermission[1]
...
```

History

#1 - 06/25/2010 11:58 AM - Jim Regetz

I think there are a few separate issues, all in the Access setPageData method. Proposed patch attached, with explanations below.

1. The paths in accessAllowList and accessDenyList are each constructed with their own indices, like this:

/access/allow[1]
/access/deny[1]
/access/deny[2]

...but the while loops that iterate over these two lists use a common accessPredicate counter, as though they were like this:

/access/allow[1]
/access/deny[2]
/access/deny[3]

This is why 'deny' elements are not processed correctly if any 'allow' elements exist in the EML. I split it into two separate counters.

2. When the while loop encounters a public map, it still needs to increment the counter before continuing to the next loop iteration. This is changed in the patch. To be conservative, I left it so it still doesn't iterate when encountering an empty map, but I haven't totally thought it through.

3. This one doesn't seem to make a difference, but the path constructed and passed to nextStep.setPageData does not have a trailing slash in the allow case, but does in the deny case. For consistency, I changed the latter to be like the former, but perhaps the reverse is preferable?

#2 - 06/29/2010 10:37 AM - ben leinfelder

Jim - these changes are very good.
I tracked down where the empty maps were being added to the access rule lists (your point #2) and made sure that's not happening, at least for normal/valid eml.
You are correct with comment #3, it should not have a trailing slash as the AccessPage appends this when using the passed in xPath.

#3 - 06/29/2010 10:44 AM - ben leinfelder

now in the SVN trunk

#4 - 03/27/2013 02:29 PM - Redmine Admin

Original Bugzilla ID was 5058

Files

Access.java.patch	1.9 KB	06/25/2010	Jim Regetz