Metacat - Bug #5443

pathquery does not handle 'matches-exactly' or 'equals' searchmode values correctly

07/19/2011 04:47 PM - Duane Costa

Status:	Resolved	**Start date:**	07/19/2011
Priority:	Normal	**Due date:**	
Assignee:	ben leinfelder	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.x.y	**Spent time:**	0.00 hour
Bugzilla-Id:	5443		

Description

Metacat PathQuery used to support 'matches-exactly' as one of its possible 'searchmode' values. It appears that this is no longer the case. The string 'matches-exactly' appears in the JavaDoc comments for the two QueryTerm constructors:

- @param searchmode
- determines what kind of substring match is performed (one of
- starts-with|ends-with|contains|matches-exactly)

However, it no longer appears in the source code itself. When setting 'matches-exactly' as the searchmode value, the search behaves as if the searchmode value was instead set to 'contains'. For example, the following pathquery:

```
<pathquery version="1.2">
<querytitle>Advanced Search</querytitle>
<returnfield>dataset/title</returnfield>
<returnfield>originator/individualName/surName</returnfield>
<returnfield>dataset/creator/individualName/surName</returnfield>
<returnfield>originator/organizationName</returnfield>
<returnfield>creator/organizationName</returnfield>
<returnfield>keyword</returnfield>
<querygroup operator="UNION">
<queryterm searchmode="matches-exactly" casesensitive="false">
<value>ax</value>
<pathexpr>abstract/para</pathexpr>
</queryterm>
<queryterm searchmode="matches-exactly" casesensitive="false">
<value>ax</value>
<pathexpr>abstract/section/para</pathexpr>
</queryterm>
<queryterm searchmode="matches-exactly" casesensitive="false">
<value>ax</value>
<pathexpr>dataset/title</pathexpr>
</queryterm>
<queryterm searchmode="matches-exactly" casesensitive="false">
<value>ax</value>
<pathexpr>keyword</pathexpr>
</queryterm>
<queryterm searchmode="matches-exactly" casesensitive="false">
<value>ax</value>
<pathexpr>surName</pathexpr>
</queryterm>
</querygroup>
</pathquery>
```

generates the following SELECT statement:

```
(SELECT DISTINCT docid FROM xml_path_index WHERE  (UPPER LIKE '%AX%' AND path IN
('abstract/para','abstract/section/para','dataset/title','keyword','surName')))
```

I've also tried using a 'searchmode' value of 'equals' and the results are the same. The following pathquery:

```
<pathquery version="1.2">
<querytitle>Advanced Search</querytitle>
```

```
<returnfield>dataset/title</returnfield>
<returnfield>originator/individualName/surName</returnfield>
<returnfield>dataset/creator/individualName/surName</returnfield>
<returnfield>originator/organizationName</returnfield>
<returnfield>creator/organizationName</returnfield>
<returnfield>keyword</returnfield>
<querygroup operator="UNION">
<queryterm searchmode="equals" casesensitive="false">
<value>ab</value>
<pathexpr>abstract/para</pathexpr>
</queryterm>
<queryterm searchmode="equals" casesensitive="false">
<value>ab</value>
<pathexpr>abstract/section/para</pathexpr>
</queryterm>
<queryterm searchmode="equals" casesensitive="false">
<value>ab</value>
<pathexpr>dataset/title</pathexpr>
</queryterm>
<queryterm searchmode="equals" casesensitive="false">
<value>ab</value>
<pathexpr>keyword</pathexpr>
</queryterm>
<queryterm searchmode="equals" casesensitive="false">
<value>ab</value>
<pathexpr>surName</pathexpr>
</queryterm>
</querygroup>
</pathquery>
```

generates the following SELECT statement:

```
(SELECT DISTINCT docid FROM xml_path_index WHERE  (UPPER LIKE '%AB%' AND path IN
('abstract/para','abstract/section/para','dataset/title','keyword','surName')))
```

In both cases, the SELECT statement is constructed as if for a 'contains' searchmode.

I'm not sure whether support for 'matches-exactly' was withdrawn from Metacat intentionally or by accident, but I think it would be valuable to restore it. For example, LTER is now using a controlled vocabulary to improve searching its data catalog. Some of the search terms in that controlled vocabulary are short chemical formulas such as 'C' or 'CO'. Whenever a search term is three characters or fewer in length, we use 'matches-exactly' as the searchmode, otherwise we use 'contains'. Since 'matches-exactly' is not supported, all short search terms such as 'C' or 'CO' are matching virtually every EML document in the catalog. Consequently, the search results in these cases are not useful to the end user.

History

#1 - 07/19/2011 05:25 PM - ben leinfelder

This looks like a problem with some of the performance enhancements that have been added over the years. For query terms on different paths that use the same value, we are generating more compact SQL - but ignoring the search mode.
I think we should only consider the query terms as "the same" if they also share the same search mode.
This should be fairly straightforward to adjust.

#2 - 07/20/2011 02:52 PM - ben leinfelder

I've changed this in trunk (targeted for 1.10 release) - it should honor the searchmode used for the query term.

#3 - 07/21/2011 04:03 PM - ben leinfelder

I've added this to the 1.9.5 Metacat branch

#4 - 03/27/2013 02:30 PM - Redmine Admin

Original Bugzilla ID was 5443