BooleanQuerySyntax

Confusion

I found something kind of weird about the way Lucene interprets boolean expressions without parenthesis. when i run the query A AND B OR C, it returns only the documents that have A(in other words as if the query was just the term A). when I run the query A OR B AND C, it returns only the documents that have B AND C(as if the query was just B AND C). I set the default operator in my application to be AND. can anyone explain this behavior, thanks.

Explanation

Lucene indeed does some funny stuff with boolean operators. Output the toString of your resultant Query's to see the details, or have a look at the Bugzilla issue that Morus mentions below.

First some background: BooleanQuery clauses each have their own set of required/optional/prohibited attributes. Putting operators between them is awkward in the Lucene sense because the parser has to set each clause individually, not in relation to another one.

QueryParser takes the most recent operator and applies it to both the clause before and after, and there is no sense of operator precedence (such as in a math expression like 1 + 2 * 3). In the case of A AND B OR C, when the AND is encountered, it sets the required attribute for both A and B, but then when the OR is encountered it sets the optional attribute on B and C, stepping on the previous required flag for B. And similarly with A OR B AND C.

I agree that the current behavior is awkward. Is it worth breaking backwards compatibility to correct this with the patch applied?

As for the default operator, it is not coming into play in your expression examples because there all clauses have an explicit conjunction that sets the flag. The default operator comes into play for an expression like A B AND C and would be used to set the flag on the A clause.

Changing Your Mindset

When dealing with Lucene people are strongly encouraged to think in terms of MUST, MUST_NOT and SHOULD (which are represented in the query parser as the prefixes "+", "-" and the default) instead of in terms of AND, OR, and NOT ... Lucene's Boolean Queries (and thus Lucene's QueryParser) is not a strict Boolean Logic system, so it's best not to try and think of it like one.

Related Links

http://www.mail-archive.com/java-user@lucene.apache.org/msg00008.html

https://issues.apache.org/jira/browse/LUCENE-167