Item 172
git_comments:

git_commits:

1. **summary:** Add unit tests for isNumeric case (JENA-512)
   message: Add unit tests for isNumeric case (JENA-512) git-svn-id: https://svn.apache.org/repos/asf/jena/trunk@1514087 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** SSE Tags are used inconsistently
   description: In some of our more recent code we are trying to map functions into a messaging data structure by inspecting the function symbol. However we have found that there are some inconsistencies in the symbols. Namely that the capitalization and punctuation are not consistent, likely this cannot change as it would mean breaking many existing algebra examples/tests and systems like ours that use algebra strings internally. The more concerning inconsistencies are that some functions report symbols without using the tag constants and so some queries written report symbols that don't match their tag constants. BuilderExpr appears to get around this by doing case insensitive key lookup which seems very hacky There is also at least one function (isNumeric) which has no Tag constant and no SSE builder defined for it so queries containing this cannot be decoded from algebra. I plan to do two things: - Make expression classes return their Tags constant where they don't already and particularly in the cases where the two values aren't exact matches - Fix the isNumeric case (and any others I discover) where there is no registered builder for the symbol

jira_issues_comments:

1. Commit 1514086 from [~rvesse] in branch 'jena/trunk' [https://svn.apache.org/r1514086] Add Tags.tagIsNumeric and associated SSE builder (JENA-512)
2. Commit 1514087 from [~rvesse] in branch 'jena/trunk' [https://svn.apache.org/r1514087] Add unit tests for isNumeric case (JENA-512)
3. Commit 1514099 from [~rvesse] in branch 'jena/trunk' [https://svn.apache.org/r1514099] Ensure SPARQL built-in expressions use symbol constants from Tags wherever possible instead of hard coding values in each file (JENA-512)
4. SUCCESS: Integrated in Jena_Development_Test #838 (See [https://builds.apache.org/job/Jena_Development_Test/838/]) Add unit tests for isNumeric case (JENA-512) (rvesse: rev 1514087) * /jena/trunk/jena-arq/src/test/java/com/hp/hpl/jena/sparql/syntax/TestSSE_Builder.java Add Tags.tagIsNumeric and associated SSE builder (JENA-512) (rvesse: rev 1514086) * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/sse/Tags.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/sse/builders/BuilderExpr.java
5. Commit 1514101 from [~rvesse] in branch 'jena/trunk' [https://svn.apache.org/r1514101] Use correct Tags constant for E_UnaryMinus (JENA-512)
6. FAILURE: Integrated in Jena_Development_Test #839 (See [https://builds.apache.org/job/Jena_Development_Test/839/]) Use correct Tags constant for E_UnaryMinus (JENA-512) (rvesse: rev 1514101) * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_UnaryMinus.java Ensure SPARQL built-in expressions use symbol constants from Tags wherever possible instead of hard coding values in each file (JENA-512) (rvesse: rev 1514099) * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Add.java *

/jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_BNode.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Bound.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Call.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Coalesce.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Conditional.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Datatype.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Divide.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Equals.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Exists.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_GreaterThan.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_GreaterThanOrEqual.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IRI.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IsBlank.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IsIRI.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IsLiteral.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IsNumeric.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_IsURI.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Lang.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LangMatches.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LessThan.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LessThanOrEqual.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalAnd.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalNot.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalOr.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Multiply.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_NotEquals.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_NotExists.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_NotOneOf.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_OneOf.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Random.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Regex.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_SameTerm.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Str.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_StrConcat.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_StrDatatype.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_StrLang.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Subtract.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_URI.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_UnaryMinus.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_UnaryPlus.java

7. **body:** Rob - some of this predates Tags.* so thanks for cleaning it up. Which ones are inconsistent? It may not matter if they are changed. The trouble is they are sometimes used in two ways: e.g. printing SPARQL syntax and printing algebra expressions. For example, you've changed {{FILTER sameTerm(?x,?y)}} into {{FILTER sameterm(?x,?y)}}. Are there others? Why don't we systematically have a SPARQL syntax name and an algebra name? We need 3 kinds of name: * a print name for SPARQL syntax output * an algebra name or symbol * a name only for the algebra to use in a IRI name or if symbols are hard to read. The two forms for algebra are to allow function name only versions, no symbols. The case insensitive lookup in BuilderExpr is because these expressions may be written by people. "isIri" vs "isiri" vs "isIRI" shouldn't matter - it's a matter of style.
   label: code-design
8. Commit 1514373 from [~rvesse] in branch 'jena/trunk' [https://svn.apache.org/r1514373] Change capitalization on a couple of Tags constants to produce more readable query serializations (JENA-512)
9. SUCCESS: Integrated in Jena_Development_Test #843 (See [https://builds.apache.org/job/Jena_Development_Test/843/]) Change capitalization on a couple of Tags constants to produce more readable query serializations (JENA-512) (rvesse: rev 1514373) * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/sse/Tags.java
10. Yes I think there may be such as langmatches The advantage of using the Tags constants is now we can tweak them as necessary and they should automatically be reflected everywhere. I have tweaked sameTerm and langMatches appropriately but there may be others we want to tweak. Since both SSE and

SPARQL itself have case insensitive keywords the exact capitalization is a matter of taste. Please go ahead and tweak the constants in Tags as you feel appropriate

11. Commit 1514832 from [~andy.seaborne] in branch 'jena/trunk' [https://svn.apache.org/r1514832] JENA-512 Put algebra function names in Tags. Add equivalent builders for expessions by function name. Use constants in E_* classes. Rename printName as functionName. Some tests.
12. ExprFunction has three operations: getFunctionPrintName -- Used by the SPARQL formatter getFunctionName -- Used by the algebra formatter getOpName -- return the symbol name like '+' I've removed in-class constants and put all printNames (renamed functionName) into Tags. Given names to symbol-only things like "=" as "eq".
13. SUCCESS: Integrated in Jena_Development_Test #845 (See [https://builds.apache.org/job/Jena_Development_Test/845/]) JENA-512 Put algebra function names in Tags. Add equivalent builders for expessions by function name. Use constants in E_* classes. Rename printName as functionName. Some tests. (andy: rev 1514832) * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Add.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Conditional.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Divide.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Equals.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_GreaterThan.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_GreaterThanOrEqual.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LessThan.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LessThanOrEqual.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalAnd.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalNot.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_LogicalOr.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Multiply.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_NotEquals.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_NotOneOf.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_OneOf.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_SameTerm.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_Subtract.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_UnaryMinus.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/E_UnaryPlus.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/expr/ExprFunction.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/sse/Tags.java * /jena/trunk/jena-arq/src/main/java/com/hp/hpl/jena/sparql/sse/builders/BuilderExpr.java * /jena/trunk/jena-arq/src/test/java/com/hp/hpl/jena/sparql/syntax/TestSSE_Builder.java