



Corese/KGRAM

Corese: Conceptual Resource Search Engine
KGRAM: Knowledge Graph Abstract Machine
<http://wimmics.inria.fr/corese>

- RDF/S & SPARQL
- KGRAM: SPARQL 1.1 Interpreter
- Java 1.6
- Design by Wimmics team @ Inria/I3S

Goodies

- SPARQL 1.1
 - XPath and SQL
 - Path variable, length, enumeration
 - Approximate search
 - Pragma
 - RDF Graph as Query Graph
 - Query several graphs
 - Distributed Query
- SPARQL Templates for RDF Pretty Printing
- Inference Rule Engine

Corese API : Create a Graph

```
import fr.inria.edelweiss.kgraph.core.*;

Graph g = Graph.create();

// with RDFS entailments:
Graph g = Graph.create(true);
```

Corese API : Load RDF/S & Rule

```
Load ld = Load.create(g);

ld.load("/home/kb/schema.rdfs");
ld.load("/home/kb/rule.rul");
ld.load("/home/kb/data.rdf");
ld.load("/home/kb/graph.ttl");
```

Corese API : Query

```
QueryProcess exec = QueryProcess.create(g);

String query =
"prefix c: <http://www.inria.fr/schema#> " +
"select * where {?x c:name ?n}";

try {
  Mappings map = exec.query(query);
}
catch (EngineException e){}
```

Corese API : Debug Query

```
String query =  
"prefix c: <http://www.inria.fr/schema#>" +  
"select debug * where {?x c:name ?n}" ;
```

Corese API : Result

```
Mappings map = exec.query(q);  
  
for (Mapping m : map) {  
    IDatatype dt = (IDatatype) m.getValue("?x");  
    dt.stringValue();  
    dt.intValue();  
    dt.doubleValue();  
}
```

Corese API : Pretty Print Result

```
ResultFormat f = ResultFormat.create(map);  
  
System.out.println(f);  
  
// XML Result (select where)  
// RDF Result (construct where)
```

Corese API : Pretty Print Graph

```
TripleFormat f = TripleFormat.create(g);  
  
System.out.println(f);  
  
// Triple Format
```

Inference Rules

```
ld.load("data.rul");  
  
RuleEngine re = ld.getRuleEngine();  
  
re.process();  
  
g.addEngine(re);  
g.process();
```