FORTH-ICS
Institute of Computer Science

Records XML Format

The xml file starts with the following line: *<?xml version="1.0" encoding="UTF-8" standalone="no" ?>*. The root node of the xml is described by the tag <*Records*>.

Each record is described by a “<*Record*>” node which is included into the root node. The child nodes <*id*> and <*rank*> must be included in each Record node. The <*id*> contains the identifier of the *Record* while the <*rank*> is the ranking of the *Record*. In case that we have not any information about the ranking we put the value *0.0* (is float). Then we have to add the information about the terms of the facets that this record is described. We create a child node for each facet where it has as tag name the name of the facet’s taxonomy and as value the name of the term.

Below we can see a xml file which contains a record with *id* = 1, *rank* = 0.0, and it is described by the term *Spring* of the facet *Season*, *100* by the facet *Resolution* and *Greece* by the facet *Location*.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Records>
        <Record>
                <id>1</id>
                <rank>0.0</rank>
                <Season>Spring</Season>
                <Resolution>100</Resolution>
                <Location>Greece</Location>
        </Record>
        ………….
        ………….
</Records>
```

Nikos Armenatzoglou
armenan@ics.forth.gr

FORTH-ICS
Institute of Computer Science

Facets TXT Format

This txt file contains information about the facets, and the taxonomies (hierarchical organized or not). When a line starts with the symbol @ then it contains information about the taxonomy of a specific facet. The first parameter is the name of the taxonomy. The second defines the Java type of the terms that are included to the taxonomy (e.g. String, Integer). The third defines if the taxonomy is hierarchical organized or not, in case that it is put *H*; *NH* otherwise. The fourth parameter defines if you want to check the relationships when you set a new one (redundancy, cycles). In case we prefer to check the relationships we write *CHECK*; *UNCHECK* otherwise. The fifth parameter defines if the terms of the taxonomy are comparable or not, in case that it is we write *COMP*; *NOCOMP* otherwise. The sixth parameter defines if a record can be described by more than one term from the same facet. In case yes, we write *COMPLEX*; *SIMPLE* otherwise. Finally, the last parameter defines the name of the facet.

When a line starts with the symbol - these means that this line describes the terms of the last defined taxonomy. In case that the taxonomy is hierarchical organized we have to define the terms and their relationships. Each relationship is defined by a parent to son relationship. The first term is the parent while the second is the son. For the head of the taxonomy, we add as first term the "*topElement"* and the second is the name of the head.

When a line starts with the # symbol then it is for comments.

Below we can see a txt file which contains information about two facets. The facet Season is not hierarchical and a record can belong to only one season. On the other hand, a record can be described by more than one terms of the facet Location which is hierarchical organized with hierarchy:

```
World >
        Europe >
                Greece >
                        Crete
                Italy >
                        Rome
```

```
@ LCTs, String, H, CHECK, NOCOMP, COMPLEX, Location
- topElement, World
- World, Europe
- Europe, Greece
- Europe, Italy
- Greece, Crete
- Italy, Rome
@ SNs, String, NH, NOCHECK, NOCOMP, SIMPLE, Season
- Summer, Autumn, Winter, Spring
```

Nikos Armenatzoglou
armenan@ics.forth.gr