R2RML Mapping Documentation

Namespace prefix bindings

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix xmlns: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix se: <http://www.semanticweb.org/kde/ontologies/sport-events#> .
@prefix base: <http://www.semanticweb.org/kde/ontologies/sport-events#> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
```

Using Dataset Teams.csv

```
<#TriplesMapTeams>
rr:logicalTable [
  rr:tableName "Teams";
];
```

Subject of the Triple Map. Defining Class-Team

```
 rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{TEAM}";
    rr:class se:Team;
 ];
```

Predicates and Objects of the class Team

```
rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [ rr:column "TEAM" ] ;
  ] ;
rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:points;
  ];

   rr:objectMap [
    rr:column "POINTS";
   ];
];
.
```

Using Dataset Fixture.csv

Note: This Triple Map is created to get the value for property isOpponentof in TriplesMapAwayVsHOME

```
 <#TriplesMapTeams2>
 rr:logicalTable [
  rr:tableName "Fixture";
 ];

## Subject of the Triple Map. Class-Team
  rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{HOMETEAM}";
    rr:class se:Team;
  ];

## Predicates and Objects of the class Team
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [
        rr:column "HOMETEAM"
        ] ;
   ] ;
 .

# Using Dataset Fixture.csv
### Note:This Triple Map is created to get the value for property isOpponentof in TriplesMapHOMEvsAWAY

<#TriplesMapTeams3>
 rr:logicalTable [
  rr:tableName "Fixture";
 ];

## Subject of the Triple Map. Class-Team
  rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{AWAYTEAM}";
    rr:class se:Team;
  ];

## Predicates and Objects of the class Team
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [
       rr:column "AWAYTEAM"
       ] ;
   ] ;
 .

# Using Dataset Players_Score.csv
```

```
 <#TriplesMapClub>
 rr:logicalTable [
  rr:tableName "Players_Score";
 ];

## Subject of the Triple Map. Defining Class-Club
  rr:subjectMap [
     rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{CLUB}";
     rr:class dbo:SoccerClub;
  ];

## Predicates and Objects of the class Club
 rr:predicateObjectMap [
     rr:predicate rdfs:label ;
     rr:objectMap [
         rr:column "CLUB"
         ] ;
   ] ;

 rr:predicateObjectMap [
    rr:predicateMap [
     rr:constant dbo:SoccerClub;
    ];

     rr:objectMap [
      rr:column "CLUB";
     ];
 ];
 .

# Using Dataset Players_Stats.csv
 <#TriplesMapRole>
 rr:logicalTable [
   rr:tableName "Players_Stats";
 ];

  rr:subjectMap [
     rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{ROLE}";
     rr:class se:Role;
  ];

## Subject of the Triple Map. Defining Class-Role
 rr:predicateObjectMap [
     rr:predicate rdfs:label ;
```

```
    rr:objectMap [
        rr:column "ROLE"
        ] ;
    rr:objectMap [
        rr:column "ROLE" ;
        rr:language "en" ] ;
  ] ;

## Predicates and Objects of the class Role
 rr:predicateObjectMap [
  rr:predicateMap [
   rr:constant se:Role;
  ];

   rr:objectMap [
    rr:column "ROLE";
   ];
 ];
 .

# Using Dataset Teams.csv
 <#TriplesMapCountry>
 rr:logicalTable [
  rr:tableName "Teams";
 ];

## Subject of the Triple Map. Defining Class-Country
  rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{TEAM}";
    rr:class se:Country;
  ];

## Predicates and Objects of the class Country
 rr:predicateObjectMap [
    rr:predicate rdfs:label;
    rr:objectMap [
        rr:column "TEAM"
        ] ;
    rr:objectMap [
        rr:column "TEAM" ;
        rr:language "en" ];
  ] ;

 rr:predicateObjectMap [
```

```
  rr:predicateMap [
    rr:constant se:Country;
  ];

   rr:objectMap [
    rr:column "TEAM";
   ];
 ];
 .
```

#Using Dataset Fixture.csv

```
<#TriplesMapCity>
 rr:logicalTable [
  rr:tableName "Fixture";
 ];

## Subject of the Triple Map. Defining Class-City
  rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{CITY}";
    rr:class dbo:City;
  ];

## Predicates and Objects of the class City
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [
        rr:column "CITY"
        ] ;
    rr:objectMap [
       rr:column "CITY" ;
       rr:language "en" ];
  ] ;

### Relating Team City to Country using predicate isIn
 rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:isIn;
  ];
   rr:objectMap [
    rr:parentTriplesMap <#TriplesMapCountry>;
    rr:joinCondition [
     rr:child "COUNTRY";
     rr:parent "TEAM";
```

```
    ];
   ];
  ];
  .

#Using Dataset Fixture.csv
  <#TriplesMapStadium>
  rr:logicalTable [
    rr:tableName "Fixture";
  ];

## Subject of the Triple Map. Defining Class-Stadium
   rr:subjectMap [
      rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{LOCATION}";
      rr:class dbo:Stadium;
   ];

## Predicates and Objects of the class Stadium
  rr:predicateObjectMap [
      rr:predicate rdfs:label ;
      rr:objectMap [
          rr:column "LOCATION"
          ] ;
    ] ;
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant dbo:Stadium;
    ];

     rr:objectMap [
      rr:column "LOCATION";
     ];
  ];

 ### Relating Stadium to City using predicate isIn
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:isIn;
    ];

     rr:objectMap [
      rr:parentTriplesMap <#TriplesMapCity>;
      rr:joinCondition [
       rr:child "CITY";
```

```
      rr:parent "CITY";
    ];
   ];
 ];
 .
```

#Using Dataset Players_Score.csv

```
 <#TriplesMapPlayer1>
 rr:logicalTable [
  rr:tableName "Players_Score";
 ];
```

Subject of the Triple Map. Defining Class-SoccerPlayer

```
  rr:subjectMap [
    rr:template "http://dbpedia.org/ontology/{PLAYER}";
    rr:class dbo:SoccerPlayer;
  ];
```

Predicates and Objects of the class SoccerPlayer

```
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [
       rr:column "PLAYER"
       ] ;
  ] ;
```

Relating Soccer Player to Club using predicate playsIn

```
 rr:predicateObjectMap [
   rr:predicateMap [
     rr:constant se:playsIn;
   ];

    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapClub>;
     rr:joinCondition [
      rr:child "CLUB";
      rr:parent "CLUB";
    ];
   ];
 ];
```

Other Predicates and Objects of the class SoccerPlayer

```
 rr:predicateObjectMap [
   rr:predicateMap [
     rr:constant dbo:age;
```

```
  ];

   rr:objectMap [
    rr:column "AGE";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:aerialAttacksWon;
    rr:termtype rr:Literal;
    rr:datatype xsd:double;
  ];

   rr:objectMap [
    rr:column "AERIALSWON";
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:assists;
  ];

   rr:objectMap [
    rr:column "ASSISTS";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:averageShotsPerGame;
  ];

   rr:objectMap [
    rr:column "SPG";
    rr:termtype rr:Literal;
    rr:datatype xsd:double;
   ];
];
```

```
rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:goals;
  ];

   rr:objectMap [
    rr:column "GOALS";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:manOfTheMatch;
  ];

   rr:objectMap [
    rr:column "MOTM";
    rr:termtype rr:Literal;
    rr:datatype xsd:int;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:passSucccessPercentage;
  ];

   rr:objectMap [
    rr:column "PS";
    rr:termtype rr:Literal;
    rr:datatype xsd:double;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:rating;
  ];

   rr:objectMap [
    rr:column "RATING";
```

```
    rr:termtype rr:Literal;
    rr:datatype xsd:double;
   ];
 ];

 rr:predicateObjectMap [
  rr:predicateMap [
   rr:constant se:redCards;
  ];

   rr:objectMap [
    rr:column "RED";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
 ];

 rr:predicateObjectMap [
  rr:predicateMap [
   rr:constant se:yellowCards;
  ];

   rr:objectMap [
    rr:column "YEL";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
 ];
 .

#Using Dataset Players.csv
 <#TriplesMapPlayer2>
 rr:logicalTable [
  rr:tableName "Players";
 ];

## Subject of the Triple Map. Defining Class-SoccerPlayer
### Note: SoccerPlayer class is defined again to get the Soccer Player data from different dataset i.e. Players.csv
  rr:subjectMap [
    rr:template "http://dbpedia.org/ontology/{PLAYER}";
    rr:class dbo:SoccerPlayer;
  ];

## Predicates and Objects for the class SoccerPlayer
```

```
  rr:predicateObjectMap [
     rr:predicate rdfs:label ;
     rr:objectMap [
        rr:column "PLAYER"
        ] ;
    ] ;
```

Relating Soccer Player to Country using predicate hasNationality

```
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:hasNationality;
    ];

     rr:objectMap [
      rr:parentTriplesMap <#TriplesMapCountry>;
      rr:joinCondition [
       rr:child "NATIONALITY";
       rr:parent "TEAM";
     ];
    ];
  ];
```

Relating Soccer Player to Team using predicate playsIn

```
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:playsIn;
    ];

     rr:objectMap [
      rr:parentTriplesMap <#TriplesMapTeams>;
      rr:joinCondition [
       rr:child "NATIONALITY";
       rr:parent "TEAM";
     ];
    ];
  ];
  .
```

```
#Using Dataset Players_Stats.csv
  <#TriplesMapPlayer3>
  rr:logicalTable [
    rr:tableName "Players_Stats";
  ];
```

```
## Subject of the Triple Map. Defining Class-SoccerPlayer
### Note: SoccerPlayer class is defined again to get the Soccer Player data from different dataset i.e. Players_Stats.csv
  rr:subjectMap [
    rr:template "http://dbpedia.org/ontology/{PLAYER}";
    rr:class dbo:SoccerPlayer;
  ];

## Predicates and Objects for the class SoccerPlayer
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [
        rr:column "PLAYER"
        ] ;
  ] ;

### Relating Soccer Player to Role using predicate hasRole
 rr:predicateObjectMap [
   rr:predicateMap [
    rr:constant se:hasRole;
  ];

### Other Predicates and Objects for the class SoccerPlayer
    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapRole>;
     rr:joinCondition [
      rr:child "ROLE";
      rr:parent "ROLE";
    ];
   ];
 ];

 rr:predicateObjectMap [
   rr:predicateMap [
    rr:constant se:performanceAttack;
  ];

   rr:objectMap [
    rr:column "PERFATTACK";
    rr:termtype rr:Literal;
    rr:datatype xsd:int;
   ];
 ];

 rr:predicateObjectMap [
```

```
  rr:predicateMap [
    rr:constant se:performanceDefense;
  ];

   rr:objectMap [
    rr:column "PERFDEF";
    rr:termtype rr:Literal;
    rr:datatype xsd:int;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:performancePossesion;
  ];

   rr:objectMap [
    rr:column "PERFPOSS";
    rr:termtype rr:Literal;
    rr:datatype xsd:int;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:performanceTotal;
  ];

   rr:objectMap [
    rr:column "TOTAL";
    rr:termtype rr:Literal;
    rr:datatype xsd:int;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:playedGames;
  ];

   rr:objectMap [
    rr:column "PLAYEDGAMES";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
```

```
    ];
  ];

  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:playedMinutes;
    ];

     rr:objectMap [
      rr:column "PLAYEDMINS";
      rr:termtype rr:Literal;
      rr:datatype xsd:nonNegativeInteger;
     ];
  ];

  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:rank;
    ];

     rr:objectMap [
      rr:column "RANK";
      rr:termtype rr:Literal;
      rr:datatype xsd:nonNegativeInteger;
     ];
  ];
  .
```

Using Dataset Fixture.csv

```
  <#TriplesMapMatch>
  rr:logicalTable [
    rr:tableName "Fixture";
  ];
```

Subject of the Triple Map. Defining Class-Match

```
   rr:subjectMap [
      rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{DATE}{HOMETEAM}vs{AWAYTEAM}";
      rr:class dbo:FootballMatch;
   ];
```

Predicates and Objects for the class Match

Relating Match to Team using predicate hasAwayTeam

```
  rr:predicateObjectMap [
    rr:predicateMap [
```

```
    rr:constant se:hasAwayTeam;
   ];
    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapTeams>;
     rr:joinCondition [
      rr:child "AWAYTEAM";
      rr:parent "TEAM";
    ];
   ];
 ];

### Relating Match to Team using predicate hasHomeTeam
  rr:predicateObjectMap [
   rr:predicateMap [
    rr:constant se:hasHomeTeam;
   ];

    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapTeams>;
     rr:joinCondition [
      rr:child "HOMETEAM";
      rr:parent "TEAM";
    ];
   ];
 ];

### Relating Match to Team using predicate hasHomeTeam
  rr:predicateObjectMap [
   rr:predicateMap [
    rr:constant se:tookPlaceIn;
   ];

    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapStadium>;
     rr:joinCondition [
      rr:child "LOCATION";
      rr:parent "LOCATION";
    ];
   ];
 ];

### Other Predicates and Objects for the class SoccerPlayer
  rr:predicateObjectMap [
   rr:predicateMap [
```

```
    rr:constant se:awayTeamScore;
  ];

   rr:objectMap [
    rr:column "AWAYTEAMSCORE";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:homeTeamScore;
  ];

   rr:objectMap [
    rr:column "HOMETEAMSCORE";
    rr:termtype rr:Literal;
    rr:datatype xsd:nonNegativeInteger;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:group;
  ];

   rr:objectMap [
    rr:column "GROUP";
    rr:termtype rr:Literal;
    rr:datatype rdfs:Literal;
   ];
];

rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:result;
  ];

   rr:objectMap [
    rr:column "RESULT";
    rr:termtype rr:Literal;
    rr:datatype rdfs:Literal;
   ];
```

```
  ];

  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:round;
    ];

     rr:objectMap [
      rr:column "ROUNDNUMBER";
      rr:termtype rr:Literal;
      rr:datatype rdfs:Literal;
     ];
  ];

  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:scheduledAt;
    ];

     rr:objectMap [
      rr:column "DATE";
      rr:termtype rr:Literal;
      rr:datatype xsd:dateTime;
     ];
  ];
.
```

Using Dataset Players.csv

Note:This Triple Map is created to get the values for property hasPlayers

```
  <#TriplesMapTeamPlayer>
  rr:logicalTable [
    rr:tableName "Players";
  ];
```

Subject of the Triple Map. Defining Class-Team

```
   rr:subjectMap [
      rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{NATIONALITY}";
      rr:class se:Team;
   ];
```

Predicates and Objects for the class Team

Relating Team to Player using predicate hasPlayer

```
  rr:predicateObjectMap [
      rr:predicate rdfs:label ;
      rr:objectMap [ rr:column "NATIONALITY" ] ;
```

```
  ] ;
 rr:predicateObjectMap [
  rr:predicateMap [
   rr:constant se:hasPlayer;
  ];

   rr:objectMap [
    rr:parentTriplesMap <#TriplesMapPlayer2>;
    rr:joinCondition [
     rr:child "PLAYER";
     rr:parent "PLAYER";
   ];
  ];
 ];
 .

# Using Dataset Players_Score.csv
### Note:This Triple Map is created to get the values for property hasPlayers

 <#TriplesMapClubPlayer>
 rr:logicalTable [
  rr:tableName "Players_Score";
 ];

## Subject of the Triple Map. Defining Class-Club
  rr:subjectMap [
    rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{CLUB}";
    rr:class se:Club;
  ];

## Predicates and Objects for the class Club
### Relating Club to Player using predicate hasPlayer
 rr:predicateObjectMap [
    rr:predicate rdfs:label ;
    rr:objectMap [ rr:column "CLUB" ] ;
  ] ;
 rr:predicateObjectMap [
  rr:predicateMap [
    rr:constant se:hasPlayer;
  ];

   rr:objectMap [
    rr:parentTriplesMap <#TriplesMapPlayer1>;
    rr:joinCondition [
```

```
      rr:child "PLAYER";
      rr:parent "PLAYER";
    ];
   ];
 ];
 .
```

Using Dataset Fixture.csv

Note:This Triple Map is created to get the values for property isOpponentOf

```
 <#TriplesMapHomeVsAway>
 rr:logicalTable [
  rr:tableName "Fixture";
 ];
```

Subject of the Triple Map. Defining Class-Team

```
  rr:subjectMap [
     rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{HOMETEAM}";
     rr:class se:Team;
  ];
```

Predicates and Objects for the class Team

Getting opponent Team of Home Team using predicate isOpponentOf

```
 rr:predicateObjectMap [
   rr:predicateMap [
     rr:constant se:isOpponentOf;
   ];

    rr:objectMap [
     rr:parentTriplesMap <#TriplesMapTeams3>;
     rr:joinCondition [
      rr:child "HOMETEAM";
      rr:parent "HOMETEAM";
    ];
   ];
 ];
.
 <#TriplesMapAwayVsHOME>
 rr:logicalTable [
  rr:tableName "Fixture";
 ];
```

Subject of the Triple Map. Defining Class-Team

```
  rr:subjectMap [
     rr:template "http://www.semanticweb.org/kde/ontologies/sport-events/{AWAYTEAM}";
     rr:class se:Team;
  ];
```

```
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant se:isOpponentOf;
    ];
## Predicates and Objects for the class Team
### Getting opponent of AwayTeam using predicate isOpponentOf
     rr:objectMap [
      rr:parentTriplesMap <#TriplesMapTeams2>;
      rr:joinCondition [
       rr:child "AWAYTEAM";
       rr:parent "AWAYTEAM";
     ];
    ];
  ];
.
```