

gwybodaeth Documentation

Iestyn Pryce

August 3, 2009

Revision: a3bee5b

Contents

1	Usage	1
1.1	Command Line	1
1.2	CGI	2
2	Functionality	2
2.1	Input	2
2.1.1	CSV	2
2.1.2	GeoNames XML	2
2.2	Maps	2
2.2.1	References	4

1 Usage

gwybodaeth is a set of Perl classes for conversion of data into RDF. It can be either used from a command line tool or via a CGI web script.

1.1 Command Line

The current command line tool takes three arguments, the source and map locations and the input data type. The source and map locations can be a URL or path to a file:

```
./gwyboadaeth --source=data1.csv --map=dataMap1.N3 --input=csv  
  
./gwyboadaeth --source=http://foo.bar.com/data1.csv  
--map=http://foo.bar.com/dataMap1.N3 --input=csv
```

You can also combine sources from local files and the and the web, i.e:

```
./gwybodaeth --source=data.csv --map=http://foo.bar.com/dataMap.N3
--input=csv
```

1.2 CGI

In the URL `src` is used to assign the data source and `map` is used to assign the map source. The CGI script currently only works with data and maps which are available via HTTP.

```
http://127.0.0.1/~imp25/cgi-bin/gwybodaeth.cgi?src=http://rdf123.umbc.edu/csv/office1.csv&map=http://rdf123.umbc.edu/map/officeMap1.N3
```

With the use of the CGI script it is important to remember to escape characters such as `&` in the URLs to data/maps or the script will treat them as delimiters. For example use

```
http://ws.geonames.org/findNearbyPlaceName?lat=47.3%26lng=9%26style=full
```

instead of

```
http://ws.geonames.org/findNearbyPlaceName?lat=47.3&lng=9&style=full
```

2 Functionality

2.1 Input

The input formats currently supported are XML from GeoNames¹ and CSV.

2.1.1 CSV

For CSV the use of ‘start row, *num*’ within the CSV file is supported if you wish to start reading the data from a row other than the second row².

2.1.2 GeoNames XML

2.2 Maps

The mapping files should be based on N3³. The mapping syntax is generally the same for both CSV and XML input. The major difference is what the

¹<http://www.geonames.org/>

²The first row is always skipped as it's assumed that it contains column headings

³<http://www.w3.org/DesignIssues/Notation3.html>

`var` in `Ex:$var` refers to. For CSV `var` refers to the row's column number (starting from 1); for XML `var` refers to a XML tag within the block being converted.

An example of the currently supported grammar for CSV conversion is shown in Figure 1. This mapping makes each row a Person subject with the email taken from the third column; the name being a concatenation of the first and second columns; the nickname being taken from the first column; office number and phone being taken from the the fifth and fourth columns respectively. The object for the `workAddress` predicate is set to be a `rdf:resource` referring to the block described by the `<Ex:$5>` function which defines a `work:Office` subject.

```
@prefix work:    <http://www.foo.org/work/> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix :       <#> .

<Ex:$5>
  a          work:Office ;
  work:number "Ex:$5" ;
  foaf:phone "Ex:$4" .

[]          a          foaf:Person ;
  foaf:mbox "Ex:$3" ;
  foaf:name "Ex:$1+' '+$2" ;
  foaf:nick "Ex:$1" ;
  foaf:officeNumber "Ex:$5" ;
  foaf:officePhone "Ex:$4" ;
  foaf:workAddress <Ex:$5> .
```

Figure 1: RDF123's officeMap3.N3 for CSV

The map example shown in Figure 2 shows the use of the `[]` syntax to allow inline functions. This is useful for short functions, however for more advanced functions the grammar used in Figure 1 is more suitable.

An explanation of a number of the specific mapping grammar is in Table 1.

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix :        <#> .

[]      a      foaf:Person ;
foaf:mbox "Ex:$2" ;
foaf:name "Ex:$1" ;
foaf:officeNumber "Ex:$3" ;
foaf:supervisor
    [ a      foaf:Person ;
      foaf:name "Ex:$4"
    ]
```

Figure 2: RDF123’s officeMap2.N3

2.2.1 References

The ability to reference data within a document is one of the strengths of RDF. The main RDF syntax for this is `rdf:ID` and `rdf:about` to set an anchor and `rdf:resource` to reference the anchor. Within a document each `rdf:ID` must be a unique word (i.e. a string without white space), however `rdf:about` does not have to be unique and may contain strings.

Currently if the anchor is a word `gwybodaeth` will use the `rdf:ID` syntax, otherwise it will use the `rdf:about` syntax. In all cases `rdf:resource` will refer to the anchor. The syntax for setting and referring to references is set out in table 1.

Due to the requirement for unique IDs we must avoid clashes. In order to do this a system of precedence has been put in place in the mapping file. A function has precedence over all the others which follow it in the file — the lower in the file the function comes the lower its precedence. The result of this is that if any reference anchor is defined multiple times only the first one will appear in the output.

An example of where this precedence comes into action is with the data from figure 3 and with the map from figure 4. Here we wish to have a block of information about each member of an office, with the block having a reference anchor based on the person’s name. We want to refer to the person’s supervisor using a reference. If we ran the first data line through both mapping functions we would end up with an RDF block describing the Al Turing and a short block describing Don Knuth as the supervisor as seen in fig 5. Problems arise however when we parse the next line of data. You

	Input: All
"Ex:\$var"	The value of the column <i>num</i> is placed here as the object.
<Ex:\$var>	This is a function. On its own line it defines the following block as the contents of the function with <code>rdf:ID</code> or <code>rdf:about</code> . As an object it references the defined function with <code>rdf:resource</code> .
[a ...] .	Define an inline function.
+	Allows for the concatenation within the field.
	Input: CSV
@If(<i>condition,true,false</i>)	Evaluates <i>condition</i> and returns <i>true</i> if the condition is true, otherwise it returns <i>false</i> .

Table 1: Table of supported N3 grammar.

would expect a RDF block describing Don Knuth and a short block describing Grace Hopper as a supervisor. However as we are only allowing one instance of each ID we will only get the second block about Grace Hopper because the ID for Don Knuth has already been used for the short supervisor block. This means that we have lost three pieces of information about Don Knuth when we RDFized the data.

By having applying precedence to the mapping file the first function will be mapped to the whole file before the second function is mapped to the whole file. The first pass gives us a full Person block for each of Al Turing, Don Knuth and Marvin Minsky. The second pass gives us a short supervisor block for Grace Hopper as she has yet to be defined. Now we have all the information possible from the file according to the mapping used without any redefinitions of IDs as seen in fig 6.

NAME,EMAIL,OFFICE,SUPERVISOR Al Turing,amt@umbc.edu,ITE332,Don Knuth Don Knuth,dek@umbc.edu,ITE332,Grace Hopper Marvin Minsky,mlm@umbc.edu,ITE442,Grace Hopper

Figure 3: RDF123 example office2.csv

```
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix :      <#> .

<Ex:$1>
  a      foaf:Person ;
  foaf:mbox "Ex:$2" ;
  foaf:name "Ex:$1" ;
  foaf:officeNumber "Ex:$3" ;
  foaf:supervisor <Ex:$4> .

<Ex:$4>
  a      foaf:Person ;
  foaf:name "Ex:$4" .
```

Figure 4: RDF123 example officeMap3.N3

```
<foaf:Person rdf:about="#Al Turing">
  <foaf:mbox>amt@umbc.edu</foaf:mbox>
  <foaf:name>Al Turing</foaf:name>
  <foaf:officeNumber>ITE332</foaf:officeNumber>
  <foaf:supervisor rdf:resource="#Don Knuth"/>
</foaf:Person>
<foaf:Person rdf:about="#Don Knuth">
  <foaf:name>Don Knuth</foaf:name>
</foaf:Person>
```

Figure 5: RDF/XML output of the first line of the data from fig 3 with the map shown in fig 4 without precedence

```
<?xml version="1.0"?>
<rdf:RDF xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <foaf:Person rdf:about="#Al Turing">
    <foaf:mbox>amt@umbc.edu</foaf:mbox>
    <foaf:name>Al Turing</foaf:name>
    <foaf:officeNumber>ITE332</foaf:officeNumber>
    <foaf:supervisor rdf:resource="#Don Knuth"/>
  </foaf:Person>
  <foaf:Person rdf:about="#Don Knuth">
    <foaf:mbox>dek@umbc.edu</foaf:mbox>
    <foaf:name>Don Knuth</foaf:name>
    <foaf:officeNumber>ITE332</foaf:officeNumber>
    <foaf:supervisor rdf:resource="#Grace Hopper"/>
  </foaf:Person>
  <foaf:Person rdf:about="#Marvin Minsky">
    <foaf:mbox>mlm@umbc.edu</foaf:mbox>
    <foaf:name>Marvin Minsky</foaf:name>
    <foaf:officeNumber>ITE442</foaf:officeNumber>
    <foaf:supervisor rdf:resource="#Grace Hopper"/>
  </foaf:Person>
  <foaf:Person rdf:about="#Grace Hopper">
    <foaf:name>Grace Hopper</foaf:name>
  </foaf:Person>
</rdf:RDF>
```

Figure 6: RDF/XML output of the data from fig 3 and map from fig 4 using precedence