# Rdfe tutorial – “Hello, painter!”

Vsn: 2019-02-11

=====================================

=== This document is under construction. ===

=====================================

This tutorial introduces you to the use of RDFe, using a few very simple examples.

## Prerequisites

May can read this tutorial as a text, or use it as a hands-on tutorial. In the latter case you must first install the BaseX XQuery processor. Please download from here:

<http://basex.org/download/>

and follow the instructions.

## Resources

### Data

We use a few tiny XML documents:

|  |  |
| --- | --- |
| **Data file name** | **Description of content** |
| painters.xml | a description of painters |
| painters-zh.xml | Chin  ese version of painters.xml |
| painters-with-wikidata-ids.xml | like painters.xml, with additional @wikidataID attribute |
| paintings.xml | a description of paintings;  semantically equivalent to painters.xml |
| paintings-catalog.xml | detailed description of paintings |

These documents are found in the data folder.

### Semantic maps

We use a few tiny semantic maps:

|  |  |
| --- | --- |
| **Semantic map file name** | **Description** |
| painters-minimal.rdfe.xml | Targets a document about painters; limited to painter resources and properties with literal (non-resource) values |
| painters-unconnected.rdfe.xml | Targets a document about painters; consideres painter and painting resources, but limited to properties with literal (non-resource) values |
| painters.rdfe.xml | Targets a document about painters; consideres painter and painting resources, including their connection |
| painters-zh.rdfe.xml | Targets a document about painters, Chinese version; consideres painter and painting resources, including their connection |
| paintings.rdfe.xml | Targets a document about paintings, which is semantically equivalent to a document about painters; consideres painter and painting resources, including their connection |
| paintings-catalog.rdfe.xml | Targets a catalog of paintings (paintings-catalog.xml) |
| paintings-augmented.rdfe.xml | Targets a document about painters, adding information from paintings-catalog.xml |
| paintings-augmented-if-wikidata.rdfe.xml | Targets a document about painters, adding information from paintings-catalog.xml, if wikidata link is present |

These documents are found in the rdfe folder.

## Overview

### Step 1 – getting started

We implement a mapping of painters.xml to RDF triples, limiting ourselves to properties with simple typed properties. In other words, we do not yet include properties whose values are resources.

### Step 2 – adding a second resource type

We extend the mapping to include descriptions of a second resource type, which are paintings. Again, we omit properties whose values are resources, so that painters and paintings are still unrelated.

### Step 3 – connecting resource types

We add resource-valued properties: painters get a resource whose values are paintings, and paintings get a resource whose values are painters. In other words, painters and paintings are connected.

### Step 3a – getting same RDF for semantically equivalent XML (1)

We map paintings.xml to the same semantic content as the content obtained for painters.xml. This demonstrates how different XML “shapes” may be viewed and described to have the same semantic content.

### Step 3b – getting same RDF for semantically equivalent XML (2)

We map a Chinese version of the document about painters to the same semantic content as the content obtained for painters.xml. This is another example showing how different XML “shapes” may be viewed and described to have the same semantic content.

### Step 4 – pulling in additional resources

We extend the mapping of painters.xml by a more elaborate description of paintings. The additional details are retrieved from an additional XML document.

### Step 5 – using property modedls with conditional settings

We modify a property model so as to take advantage of an additional data source if the required link information is presented.

## Preparation

The RDFe processor is invoked by command-line invocations. The examples use the variable $HOME\_SHAX, which should be set to the path of the bin folder of your shax installation.

Example for PowerShell:

$HOME\_SHAX=/tt/shax/bin

Example for Windows command-line shell:

set HOME\_SHAX=/tt/shax/bin

NOTE. When using the Windows command-line shell, the string $HOME\_SHAX must be replaced by %HOME\_SHAX%.

NOTE. The commands have linefeeds inserted for readibility. When copy-pasting the commands into the command-line interface, make sure to remove the linefeeds.

NOTE. The commands use file paths which presuppose that the command is entered in the main folder of this tutorial (parent folder of folders data and rdfe).

## Step 1 – getting started

=== Explanations will be added. ===

We implement a mapping of painters.xml to RDF triples, limiting ourselves to properties with simple typed properties. In other words, we do not yet include properties whose values are resources.

Command:

basex -b "request=rdfe?dox=data/painters.xml,semap=rdfe/painters-minimal.rdfe.xml"

-o ttl/painters-minimal.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 2 – adding a second resource type

=== Explanations will be added. ===

Command:

basex -b "request=rdfe?dox=data/painters.xml,semap=rdfe/painters-unconnected.rdfe.xml"

-o ttl/painters-unconnected.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 3 – connecting resource types

=== Explanations will be added. ===

We add resource-valued properties: painters get a resource whose values are paintings, and paintings get a resource whose values are painters. In other words, painters and paintings are connected.

Command:

basex -b "request=rdfe?dox=data/painters.xml,semap=rdfe/painters.rdfe.xml"

-o ttl/painters.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 3a – getting same RDF for semantically equivalent XML (1)

=== Explanations will be added. ===

We map paintings.xml to the same semantic content as the content obtained for painters.xml. This demonstrates how different XML “shapes” may be viewed and described to have the same semantic content.

Command:

basex -b "request=rdfe?dox=data/paintings.xml,semap=rdfe/paintings.rdfe.xml"

-o ttl/paintings.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 3b – getting same RDF for semantically equivalent XML (2)

=== Explanations will be added. ===

We map a Chinese version of the document about painters to the same semantic content as the content obtained for painters.xml. This is another example showing how different XML “shapes” may be viewed and described to have the same semantic content.

Command:

basex -b "request=rdfe?dox=data/painters-zh.xml,semap=rdfe/painters-zh.rdfe.xml"

-o ttl/painters-zh.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 4 – pulling in additional resources

=== Explanations will be added. ===

We extend the mapping of painters.xml by a more elaborate description of paintings. The additional details are retrieved from an additional XML document.

Command:

basex -b "request=rdfe?dox=data/painters.xml,semap=rdfe/painters-augmented.rdfe.xml"

-o ttl/painters-augmented.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*

## Step 5 – using property modedls with conditional settings

=== Explanations will be added. ===

We modify a property model so as to take advantage of an additional data source if the required link information is presented.

Command:

basex -b "request=rdfe?dox=data/painters-with-wikidata.xml,

semap=rdfe/painters-augmented-if-wikidata.rdfe.xml"

-o ttl/painters-augmented-if-wikidata.ttl $HOME\_SHAX/shax.xq

*(After copy-pasting – remove linefeed!)*