

MetWare screenshot: propagating XML Schema data types

Egon Willighagen 

Published August 26, 2008

Citation

Willighagen, E. (2008). MetWare screenshot: propagating XML Schema data types. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/9se4x-ewm11>

Keywords

Metware, Xml, Skos

Abstract

Just a quick screenshot. Remember our use of SKOS in MetWare ? Steffen has been working on creating integrated JSF pages, while I am focusing on autogeneration of blobs. The below screenshot is such a blob, called a UI component in JSF, which allows easy embedding the the aggregations Steffen is working on.

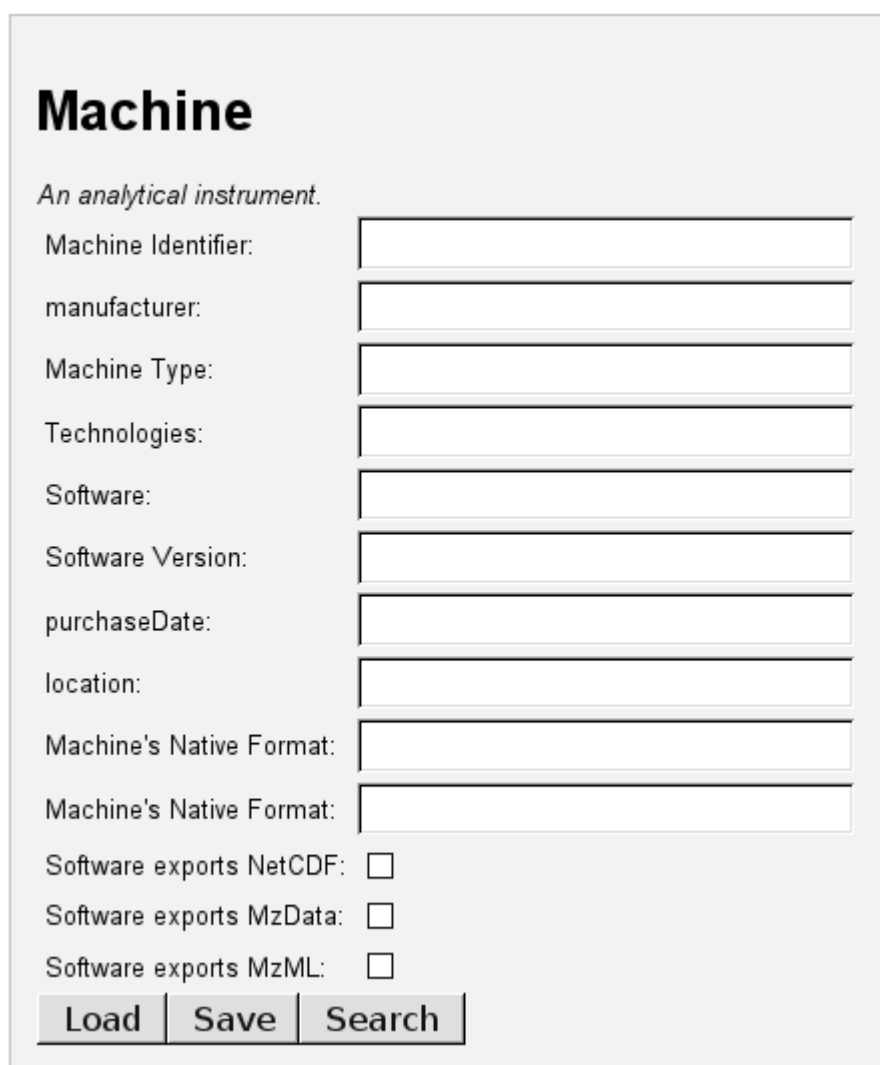
Copyright

Copyright © Egon Willighagen 2008. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

chem-bla-ics

Just a quick screenshot. Remember our use of [SKOS in MetWare](#) ? [Steffen](#) has been working on creating integrated [JSF pages](#), while I am focusing on autogeneration of blobs. The below screenshot is such a blob, called a UI component in JSF, which allows easy embedding the the aggregations Steffen is working on.

Autogeneration of web content benefits greatly from well defined input, including data types. [MetWare](#) uses [XML Schema Data Types](#) for this, as [mentioned ealier](#) when I briefly mentioned generation of search pages. That example showed the creation of range input on `xsd:integer` types. The below screenshot shows the different output for `xsd:string` (input text box) and `xsd:boolean`:



Machine

An analytical instrument.

Machine Identifier:

manufacturer:

Machine Type:

Technologies:

Software:

Software Version:

purchaseDate:

location:

Machine's Native Format:

Machine's Native Format:

Software exports NetCDF:

Software exports MzData:

Software exports MzML:

Now, this example is not really shocking, but MetWare defines additional types, for example an InChI data type:

```
<simpleType name="inchi">
  <restriction base="string">
    <pattern value="InChI=1/.*" />
  </restriction>
</simpleType>
```

chem-bla-ics

This allows me to tweak the HTML output created by the JSF pages to include [microformats](#) to support the [Sechemtic](#) userscript (see also doi:[10.1186/1471-2105-8-487](https://doi.org/10.1186/1471-2105-8-487)).

Or, to provide a drop down box, listing the allowed values:

```
<simpleType name="deviceVendor">
  <restriction base="string">
    <enumeration value="BioCrates"/>
    <enumeration value="Bruecker"/>
  </restriction>
</simpleType>
```