

Freeware XML editors

- [The editors](#)
 - [XAmple](#)
 - [Rinzo](#)
 - [XMLmind XML Editor Personal Edition](#)
- [The example XML document and schema](#)
 - [The schema file eeprom.xsd:](#)
 - [The document eeprom_example.xml:](#)

The editors

I tested each editor using the schema below, reading in the example XML document and trying to change it in invalid ways. I restricted myself to editors that are free for non-commercial use, which work under Linux and MacOS X and which don't require the installation of a log of new libraries.

XAmple

- Home page: <http://www.felixgolubov.com/XMLEditor>
- Needs only Java 1.5 or 1.6
- Uses XML-handling Swing widgets designed by Golubov.
 - No source code for the widgets, only a JAR file xmleditor.jar.
 - Swing itself is part of the Java standard library.
- The editor UI built around the widgets is called XAmple.
 - We do get source code for XAmple.
- Too-small font problem common to many Swing apps. Probably fixable.
- Allows saving of malformed documents without asking for confirmation.
- Understands xsi::noNamespaceSchemaLocation.
- Generates an editing form from the schema.
- Fast. Validation results appear without noticeable delay.

Rinzo

- Home page: <http://editorxml.sourceforge.net/>
- It's a plugin for Eclipse 3.3 or later.
- SourceForge open source project.
- No editing form. Documents are represented by syntax-colored text in one pane together with a structure tree in another.
- Doesn't understand xsi::noNamespaceSchemaLocation.
 - I had to use xsi::schemaLocation and give a bogus namespace URI along with the name of the schema file.
- Slow. Noticeable delay between typing and the results of validation.
- XML files must be part of an Eclipse project.
 - If not then saves produce error pop-ups about invalid thread states.
- Allows saving of malformed documents without asking for confirmation.
- When inserting new elements it offers a menu of possibilities that often includes invalid choices, allowing me to insert elements in the wrong order. After insertion however the editor catches the error.

XMLmind XML Editor Personal Edition

- Home page: <http://www.xmlmind.com/xmleditor>
- Freeware Java 1.6, no source code.
 - \$330 Professional Edition comes with full source.
- Schema glitch: It didn't accept the use of maxExclusive to restrict the range of xs:unsignedInt.
 - It claimed that maxExclusive and maxInclusive were both being used.
 - I'll bet maxInclusive was used when deriving xs:unsignedInt from xs:integer.
 - The two conditions should conflict only when used in the same restriction element.
 - Fixed the problem by using xs:integer as the base and adding minInclusive="0".
- Doesn't validate as you type. Validates upon load, save and upon request.
- Fast.
- Offers the correct choices when inserting new elements.
- Understands xsi::noNamespaceSchemaLocation.

The example XML document and schema

The schema file eeprom.xsd:

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="cmb1">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="boardSubtype" />
        <xs:element ref="configSchema" />
        <xs:element ref="fulcrumPort" />
        <xs:element ref="mac" />
        <xs:element ref="gatewayIp" />
        <xs:element ref="netMask" />
        <xs:element ref="endpoints" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="endpoints">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="internal" minOccurs="0" maxOccurs="12" />
        <xs:element ref="external" minOccurs="0" maxOccurs="26" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="boardSubtype" type="xs:unsignedShort" />

  <xs:element name="configSchema">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <!-- 14-bit unsigned int -->
        <xs:maxExclusive value="16384" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="fulcrumPort" type="xs:unsignedLong" />
  <xs:element name="mac" type="macAddress" />
  <xs:element name="gatewayIp" type="dottedIp" />
  <xs:element name="netMask" type="dottedIp" />
  <xs:element name="type" type="xs:unsignedByte" />
  <xs:element name="attr" type="xs:unsignedByte" />

  <xs:element name="internal" type="endpoint" />
  <xs:element name="external" type="endpoint" />

  <xs:complexType name="endpoint">
    <xs:sequence>
      <xs:element ref="type" />
      <xs:element ref="attr" />
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="macAddress">
    <!-- MAC addresses are to be only and exactly 6 bytes long. -->
    <xs:restriction base="xs:hexBinary">
      <xs:length value="6" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="dottedIp">
    <xs:restriction base="xs:string">
      <xs:pattern value="(1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])" />
    </xs:restriction>
  </xs:simpleType>

</xs:schema>

```

The document `eeeprom_example.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- The root element is one of cmb1, cmb2, ftm or rtm. -->
<cmb1
  xsi:noNamespaceSchemaLocation="eeeprom.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <!-- Version info -->
  <boardSubtype>1</boardSubtype>
  <configSchema>1</configSchema>

  <!-- Routing and crossbar -->
  <fulcrumPort>0</fulcrumPort>
  <mac>ee0000aaaa00</mac>
  <gatewayIp>192.168.1.200</gatewayIp>
  <netMask>255.255.255.0</netMask>

  <endpoints>
    <!-- Internal endpoints first, then external. Slots are
         are assigned in the appropriate LUT in order of appearance.
    -->
    <internal>
      <type>1</type>
      <attr>15</attr>
    </internal>

    <internal>
      <type>3</type>
      <attr>42</attr>
    </internal>

    <internal>
      <type>0</type>
      <attr>7</attr>
    </internal>

    <external>
      <type>5</type>
      <attr>17</attr>
    </external>

    <external>
      <type>3</type>
      <attr>6</attr>
    </external>

  </endpoints>
</cmb1>
```