ODFValidator

ODFValidator is a tool that validates OpenDocument files and checks certain conformance criteria.

ODFValidator is available as a online service [1] and as a command line tool. This page primarily describes the command line tool. Please visit the OpenOffice.org ODF Validator [2] web page for details regarding the online tool.

How to start

To use the ODFValidator, the following steps are required:

- Check out the **odfvalidator** NetBeans project.
- Add the jar files listed under Requirements to the project's libraries.
- Get the ODF 1.0, 1,1 and 1.2 schemas, and add the directories where your local copies of the schemas are stored to the **schema.user.properties** file.
- Get the MathML 1.01 DTD and the MathML 2.0 schemas, and add the directories where you local copies of the schemas are stored to the **schema.user.properties** file.
- Build the project.
- Choose an arbitray OpenDocument file **<odffile>**.
- If you are using JRE 1.5, call the ODFValidator with

```
 java -jar "<path>/odfvalidator.jar" <odffile>
```

- If you are using JRE 1.6, call the ODFValidator with

```
java
-Djavax.xml.validation.SchemaFactory:http://relaxng.org/ns/structure/1.0=org.iso_rela
-Dorg.iso_relax.verifier.VerifierFactoryLoader=com.sun.msv.verifier.jarv.FactoryLoade
-jar "<path>/odfvalidator.jar" <odffile>
```

Usage

Usage Summary

In an Java 1.5 environment, ODFValidator is invoked by the following command:

```
 java -jar "<path>/odfvalidator.jar"
```

In an Java 1.6 environment, the following command line has to be used:

```
java
-Djavax.xml.validation.SchemaFactory:http://relaxng.org/ns/structure/1.0=org.iso_rela
-Dorg.iso_relax.verifier.VerifierFactoryLoader=com.sun.msv.verifier.jarv.FactoryLoade
-jar "<path>/odfvalidator.jar"
```

These command lines are abbreviated *odfvalidator* from now on.

The ODFValidator may be called with the following options:

```
odfvalidator [-r] [-c|-s] [-d] [-v|-w] [-f <filterfile>] [-x
<expclude pattern>] [-o outputfile] [-1.0|-1.1|-1.2]
```

```
<odffiles>
odfvalidator [-r] [-c|-s] [-d] [-v|-w] [-f <filterfile>] [-x
<expclude pattern>] [-o outputfile] -S <schemafile>
<odffiles>
odfvalidator [-c|-s] [-v|-w] [-d] [-o outputfile] -C <configfile>
 odfvalidator -g <odffiles>
 odfvalidator -h
 odfvalidator -V
```

If no parameters are specified, the files **<odffiles>** are validated using the schema that belongs to the ODF version of the file. For ODF 1.0 files the ODF 1.0 schema is used, for ODF 1.1 files the ODF 1.1 schema is used, and so on. The version of an ODF file is detected for each file separately.

The options have the following meaning:

-c: Apply ODF conformance rules before validation: unknown markup is ignored during validation.

-d: Use MathML 1.01 DTD rather than MathML 2.0 schema for content.xml of formula documents.

-f: Ignore error messages specified in **<filterfile>** (see Specifying a filter file).

-g: Show the generator information of the specified **<odffiles>** without validation.

-h: Print a short help.

-o: Print output into specified file rather than standard output.

-r: Process directories recursively.

-c: Use the strict schema for validation.

-v: Verbose: print information like the generator or the documents that are processed.

-w: Print warnings.

-x: Exclude files that match the specified regular expression [3] from validation.

-C: Validate using configuration file **<configfile>**.

-S: Use the specified **<schemafile>** for validation.

-V: Print version information.

Validation using default schemas

The ODF 1.0, ODF 1.1, ODF 1.2 schemas as well as the MathML 1.01 DTD and the MathML 2.0 schema are included in the **ODFValidator.jar** file. If the ODFValidator is called without the **-S** or **-C** options, these schemas are used for validation.

```
odfvalidator [-r] [-c|-s] [-d] [-v|-w] [-f <filterfile>] [-x
<expclude pattern>] [-o outputfile] [-1.0|-1.1|-1.2]
<odffiles>
```

<odffiles> is the list of files and directories that should be validated. If a directory is specified, all files that have an OpenDocument extension (like **odt**, **ods** or **ott**) are validated. If **-r** is specified additionally, all directories are processed recursively.

By default, the ODFValidator detects the version of the files that shall be validated and chooses the corresponding schema. The command line options **-1.0**, **-1.1** and **-1.2** can be used to specify that the schemas of a particular ODF version should be used for all files, regardless of the version they specify themselves.

The command line option **-d** specifies that the MathML 1.01 DTD should be used for the validation of the **content.xml** of formula documents. Default is to use the MathML 2.0 schema.

If the **-c** command line option is specified, unknown markup is ignored as specified in the conformance rules for ODF 1.0/1.1. If the **-s** command line option is specified, the strict schema is used for validation. If neither **-c** nor **-s** are specified, the regular ODF schemas are used, but errors are reported for unknown markup unless it appears in styles or metadata.

The optional **-x** switch allows to exclude certain files or directories from the validation. The files that shall be excluded are specified by a regular expression [4]. Please note that the full absolute path names of directories and files are matched against this pattern. This means that the regular expression either must include the absolute path of the files and directories that shall be excluded, or must start with **.***. The **-x** option can be specified only once. If several paths shall be excluded, these paths have to be specified in a single regular expression using the **|** operator.

If **-w** is specified additionally, not only validation errors are reported, but also warnings.

If **-v** is specified additionally, not only validation errors and warnings are reported, but also the generator stored in the manifest, MIME types, the files that are processed, etc.

If the **-o** option is present, all messages go into the specified file. Otherwise, they are printed to standard out. It is possible to exclude known validation errors from the output by specifying a filter file using the **-f** option.

Validation using a non-default schema

Tho specify the schema that is used for **meta-xml**, **content.xml**, **styles.xml** and **settings.xml** on the command line, ODFValidator has to be called with the following parameters:

```
odfvalidator [-r] [-c|-s] [-d] [-v|-w] [-f <filterfile>] [-x
<expclude pattern>] [-o outputfile] -S <schemafile>
<odffiles>
```

<schemafile> is the schema that shall be used.

All other command line option are as described in Validation using default schemas.

Validation using a configuration file

The schemas and the files that should be validated can be specified in a configuration file. A configuration file is a Java XML properties file as described in the Java 2 API documentation [5]. The following properties are supported:

- **strict-schema**: Specifies the strict schema to be used.
- **manifest-schema**: Specifies the manifest schema to be used.
- **mathml-schema**: Specifies the MathML 1.01 schema to be used.
- **mathml2-schema**: Specifies the MathML 2 schema to be used.
- **path***: All properties whose names start with "path" are considered to be files or directories that shall be validated.

- **recursive**: This boolean property specifies whether directories are scanned recursively. It takes the values **true** and **false**.
- **exclude**: Specifies files and directories that shall be excluded as a regular expression [6]. See description of **-x** option.
- **filter**: Specifies filter file.

A sample configuration file looks like this:

```
 <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.
dtd">
 <properties>
<entry
key="strict-schema">/home/odf11-cd2/msv/OpenDocument-strict-schema-v1.1-cd2.rng

<entry
key="manifest-schema">/home/odf11-cd2/msv/OpenDocument-manifest-schema-v1.1-cd2.rng

  <entry key="mathml-schema">/home/odf11-cd2/msv/mathml2.xsd

  <entry key="path1">/home/testdocs

  <entry key="path2">/home/temp
```

Specifying a filter file

It is possible to omit the output of known validation errors by adding their error message to a so called filter file. A filter file is an XML file, which matches the following Relax-NG Compact [7] schema:

```
start = filter
filter = element filter { filter-attlist, filter-entry+ }
filter-attlist &= empty
filter-entry = element filter-entry { filter-entry-attlist, text }
filter-entry-attlist &=
  attribute task-id {
    xsd:string { pattern = '[a-z]?[0-9]+"' }
  }
filter-entry-attlist &= attribute resolved-in { xsd:integer }?
```

A sample filter file is

```
 <filter>
<filter-entry task-id="i38753" resolved-in="8864">attribute
"presentation:display-header" has a bad value. Possible values are:
false,true
```

Each **<filter-entry>** element described one message that shall be excluded. The message itself is the element content. The mandatory **task-id** attribute includes the task which resolves the validation error, or that has been submitted for the validation error if it has not been resolved already.

The optional **resolved-in** attribute contains the build-id of the master where the validation error has been resolved. Error messages are only omited if the **resolved-in** attribute is provided, or if the build-id is lower than the one which is stored in the **<meta:generator>** element of the file that is validated. This means that a validation error is not omitted if it occurs in a file which has been saved with an OOo version where the bug shall be resolved already.

If a error message is omited, a warning is printed that contains the task-id specified in the filter file. Please note that warning are only printed if either the **-w** or the **-v** switch is present.

A filter file is specified using the **-f** command line option or the **filter** property in the configuration file.

Where do I get the schemas?

The OpenDocument schemas are those provided on the `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office` OASIS OpenDocument technical Committee page.

Note: Due to an issue in MSV, errors are reported that do not exist. To avoid this, the schemas build into ODFValidator have been modified.

The MathML DTD that is included in the ODFValidator us the one that is included in the **/share/dtd/math/1_01/** folder of each OOo/StarOffice installation.

Requirements

ODFValidator requires J2RE 1.5 or J2RE 1.6. For J2RE 1.6, the following options have to be included into the **java** command line:

```
-Djavax.xml.validation.SchemaFactory:http://relaxng.org/ns/structure/1.0=org.iso_rela
-Dorg.iso_relax.verifier.VerifierFactoryLoader=com.sun.msv.verifier.jarv.FactoryLoade
```

ODFValidator further requires the following packages:

- MSV [8]. **msv.jar**, **isorelax.jar**, **relaxngDatatype.jar** and **xsdlib.jar** included in the MSV distribution must be added to the project's libraries and must be in the classpath or a folder called **lib** that is next to the **odfvalidator.jar** at runtime.
- ISORELAX JARV -> JAXP 1.3 Xml Validation Engine Adaptor [9]. **isorelax-jaxp-bridge-1.0.jar** must be added to the project's libraries and must be in the classpath or a folder called **lib** that is next to the **odfvalidator.jar** at runtime.
- ODFDOM [10] **odfdom.jar** from ODFDOM [11] must be added to the project's libraries and must be in the classpath or a folder called **lib** that is next to the **odfvalidator.jar** at runtime.

Warning: Make sure you use an up-to-date release of MSV. Older versions (before msv-20061103) report errors for the OpenDocument schema itself.

NetBeans Project

The **ODFValidator** source code [12] is located in the *tools* folder of the OpenOffice.org ODF Toolkit Project [13].

The **ODFValidator** ia a NebBeans project. To work with the project, just open the project folder.

External links

[1] http://tools.services.openoffice.org/odfvalidator

[2] http://tools.services.openoffice.org/odfvalidator

[3] http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#sum

[4] http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#sum

[5] http://java.sun.com/j2se/1.5.0/docs/api/java/util/Properties.html

[6] http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#sum

[7] http://www.relaxng.org/compact-tutorial-20030326.html

[8] https://msv.dev.java.net/

[9] https://isorelax-jaxp-bridge.dev.java.net/

[10] http://odftoolkit.openoffice.org/servlets/ProjectDocumentList?folderID=759&expandFolder=759&folderID=0

[11] http://odftoolkit.openoffice.org/servlets/ProjectDocumentList?folderID=759&expandFolder=759&folderID=0

[12] http://odftoolkit.openoffice.org/source/browse/odftoolkit/tools/odfvalidator/

[13] http://odftoolkit.openoffice.org

Source: http://wiki.services.openoffice.org/w/index.php?title=ODFValidator

Principle Authors: Mib