
Name

jing — Validator for RELAX NG in Java

Synopsis

```
jing [-i] [-c] [-t] [-f] [-e encoding] {RNGFile} {XMLFile...}
```

Options

The following options are available:

`-c`

The schema uses RELAX NG Compact Syntax.

`-e encoding`

Uses the *encoding* to read the schema.

`-f`

Checks that the document is *feasibly valid*. A document is feasibly valid if it could be transformed into a valid document by inserting any number of attributes and child elements anywhere in the tree. This is equivalent to transforming the schema by wrapping every `data`, `list`, `element` and `attribute` element in an `optional` element and then validating against the transformed schema. This option may be useful while a document is still under construction. This option also disables checking that for every IDREF there is a corresponding ID.

`-i`

Disables checking of ID/IDREF/IDREFS. By default, Jing enforces the constraints imposed by RELAX NG DTD Compatibility with respect to ID/IDREF/IDREFS.

`-t`

Prints the time used by Jing for loading the schema and for validation.

Jing uses a vendor-independent pluggable datatypes API [<http://www.thaiopensource.com/relaxng/pluggable-datatypes.html>] that allows datatype libraries to be added at runtime and be interoperable with Java-based RELAX NG implementation that supports the API.

Jing also includes an implementation of a datatype library for the W3C XML Schema Part 2 datatypes. There is a separate document about Jing's datatype [<http://www.thaiopensource.com/relaxng/jing-datatypes.html>] describing this implementation and how to use it.

Other Schema Languages

In addition to RELAX NG (both XML and compact syntax), Jing has support for some other schema languages. This support is less mature than the RELAX NG support. These schema languages all use XML and Jing can autodetect using the namespace URI of the document element. However, if you use the `-c` option, no autodetection will be performed and the schema will be parsed as RELAX NG compact syntax.

Jing has support for the following schema languages:

Schematron 1.5 [<http://xml.ascc.net/schematron>]

Jing's implementation is not based on the reference Schematron 1.5 implementation. It is implemented partly in XSLT and partly in Java. This implementation requires that the Schematron elements be properly namespaced using the namespace URI <http://www.ascc.net/xml/schematron>. Jing can report correct line numbers both for errors in the instance and errors in the schema (including XPath errors). It can use either Saxon or Xalan as its XSLT engine. Jing is distributed with Saxon, because JDK 1.4 includes an old version of Xalan, which does not work properly for this application, and it is tricky to prevent the JRE using this version. A command-line option of `-d` enables diagnostics. A command-line option of `-p phase` specifies the phase to use. Both reports and failed assertions are considered errors. The Schematron schema

is subject to rather more rigorous checking (using a RELAX NG schema) than with the reference Schematron implementation.

W3C XML Schema [<http://www.w3.org/TR/xmlschema-1/>]

Jing provides support for W3C XML Schema using a wrapper around Xerces2-J [<http://xerces.apache.org/xerces2-j/>]. Any `xsi:schemaLocation` and `xsi:noNamespaceSchemaLocation` hints in the instance are ignored.

Namespace Routing Language (NRL) [<http://www.thaiopensource.com/relaxng/nrl.html>]

The XML Namespaces Recommendation allows an XML document to be composed of elements and attributes from multiple independent namespaces. Each of these namespaces may have its own schema; the schemas for different namespaces may be in different schema languages. NRL is an experimental language for specifying how the schemas for the different namespaces are to be composed in order to allow validation of the complete document. The implementation in Jing fully supports the language described in the specification. Subschemas can be in any of the languages supported by Jing (including, recursively, NRL). Jing does not yet support the schema option for W3C XML Schema subschemas, which would allow more than one schema URI to be specified.

Modular Namespaces (MNS) [<http://www.thaiopensource.com/relaxng/mns.html>]

MNS is the predecessor to NRL, and should be considered obsolescent. MNS support will disappear in a future release.

See Also

<http://code.google.com/p/jing-trang/>

Project homepage and source code

<http://www.thaiopensource.com/relaxng/jing-manual.html>

This manual in HTML format