

All Online Learning

www.allonlinelearning.com

XML Validator (Xmllint) Example on Linux Terminal

1. ----- validation of xml file with xmllint -----

Run the xmllint command

```
[user@localhost ~]$ xmllint
```

Now we got following lines ,this shows that xmllint is present in your system

Usage : xmllint [options] XMLfiles ...

Parse the XML files and output the result of the parsing

--version : display the version of the XML library used

--debug : dump a debug tree of the in-memory document

--shell : run a navigating shell

--debugent : debug the entities defined in the document

--copy : used to test the internal copy implementation

--recover : output what was parsable on broken XML documents

--huge : remove any internal arbitrary parser limits

--noent : substitute entity references by their value

--noenc : ignore any encoding specified inside the document

--noout : don't output the result tree

--path 'paths': provide a set of paths for resources

--load-trace : print trace of all external entites loaded

--nonet : refuse to fetch DTDs or entities over network

--nocompact : do not generate compact text nodes

--htmlout : output results as HTML

--nowrap : do not put HTML doc wrapper

--valid : validate the document in addition to std well-formed check

--postvalid : do a posteriori validation, i.e after parsing

--dtdvalid URL : do a posteriori validation against a given DTD

--dtdvalidfpi FPI : same but name the DTD with a Public Identifier

--timing : print some timings

--output file or -o file: save to a given file

--repeat : repeat 100 times, for timing or profiling

--insert : ad-hoc test for valid insertions

--compress : turn on gzip compression of output



www.allonlinelearning.com

All Online Learning

www.allonlinelearning.com

--html : use the HTML parser
--xmlout : force to use the XML serializer when using --html
--nodeftdtd : do not default HTML doctype
--push : use the push mode of the parser
--pushsmall : use the push mode of the parser using tiny increments
--memory : parse from memory
--maxmem nbbytes : limits memory allocation to nbbytes bytes
--nowarning : do not emit warnings from parser/validator
--noblanks : drop (ignorable?) blanks spaces
--nocdata : replace cdata section with text nodes
--format : reformat/reindent the input
--encode encoding : output in the given encoding
--dropdtd : remove the DOCTYPE of the input docs
--pretty STYLE : pretty-print in a particular style
0 Do not pretty print
1 Format the XML content, as --format
2 Add whitespace inside tags, preserving content
--c14n : save in W3C canonical format v1.0 (with comments)
--c14n11 : save in W3C canonical format v1.1 (with comments)
--exc-c14n : save in W3C exclusive canonical format (with comments)
--nsclean : remove redundant namespace declarations
--testIO : test user I/O support
--catalogs : use SGML catalogs from \$SGML_CATALOG_FILES
otherwise XML Catalogs starting from
file:///etc/xml/catalog are activated by default
--nocatalogs: deactivate all catalogs
--auto : generate a small doc on the fly
--xinclude : do XInclude processing
--noxincludenode : same but do not generate XInclude nodes
--nofixup-base-uris : do not fixup xml:base uris
--loaddtd : fetch external DTD
--dtdattr : loaddtd + populate the tree with inherited attributes
--stream : use the streaming interface to process very large files
--walker : create a reader and walk though the resulting doc
--pattern pattern_value : test the pattern support



www.allonlinelearning.com

All Online Learning

www.allonlinelearning.com

--chkregister : verify the node registration code
--relaxng schema : do RelaxNG validation against the schema
--schema schema : do validation against the WXS schema
--schematron schema : do validation against a schematron
--sax1: use the old SAX1 interfaces for processing
--sax: do not build a tree but work just at the SAX level
--oldxml10: use XML-1.0 parsing rules before the 5th edition
--xpath expr: evaluate the XPath expression, imply --noout
Libxml project home page: <http://xmlsoft.org/>
To report bugs or get some help check: <http://xmlsoft.org/bugs.html>

2. Create .xml file:

```
[user@localhost ~]$ gedit a.xml
```

Create .dtd file:

```
[user@localhost ~]$ gedit dtda.dtd
```

3. Run following command to see the xml file:

```
[user@localhost ~]$ xmllint --format a.xml
```

Output:

```
<?xml version="1.0"?>  
<!DOCTYPE Newspaper SYSTEM "dtda.dtd">  
<bookstore>  
<book>  
<title>The Great Gatsby</title>  
<author>F. Scott Fitzgerald</author>  
<year>1925</year>  
<price>9.99</price>  
</book>  
<book>  
<title>Pride and Prejudice</title>  
<author>Jane Austen</author>  
<year>1813</year>
```



www.allonlinelearning.com

All Online Learning

www.allonlinelearning.com

```
<price>12.99</price>
```

```
</book>
```

```
</bookstore>
```

4. Validate xml file:

```
[user@localhost ~]$ xmlint --valid --noout --dtdvalid dtda.dtd a.xml
```

Output: error

```
a.xml:3: element bookstore: validity error : root and DTD name do not match  
'bookstore' and 'Newspaper'
```

```
<bookstore>
```

```
^
```

5. Correct the error and validate:

```
[user@localhost ~]$ gedit a.xml
```

```
[user@localhost ~]$ xmlint --valid --noout --dtdvalid dtda.dtd a.xml
```

Output:

No error ; xml file is now validated



www.allonlinelearning.com

All Online Learning

www.allonlinelearning.com

Program:

xml file----> a.xml

```
<?xml version="1.0"?>
<!DOCTYPE Newspaper SYSTEM "dtda.dtd">
<bookstore>
<book>
<title>The Great Gatsby</title>
<author>F. Scott Fitzgerald</author>
<year>1925</year>
<price>9.99</price>
</book>
<book>
<title>Pride and Prejudice</title>
<author>Jane Austen</author>
<year>1813</year>
<price>12.99</price>
</book>
</bookstore>
```

dtd file-----> dtda.dtd

```
!ELEMENT bookstore (book+)>
<!ELEMENT book (title, author, year, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

```
<!ATTLIST book isbn CDATA #IMPLIED>
```



www.allonlinelearning.com