



---

**PERT**  
**XML VALIDATOR TOOL**  
**SETUP AND USAGE**

---

## Table of Contents

ABOUT THIS SIMPLE QUICK START GUIDE.....	2
DISCLAIMER.....	2
XML VALIDATOR FUNCTIONALITY .....	3
SETUP OF XML VALIDATOR TOOL .....	4
USING THE XML VALIDATOR TOOL.....	6
XML VALIDATOR TOOL OUTPUT.....	7
APPENDIX A .....	11

## Version History

Ver.	DESCRIPTION OF CHANGE
V1	Initial release of document

## **ABOUT THIS SIMPLE QUICK START GUIDE**

This guide is intended to help you setup the XML Validator tool on your local system and serve as an instruction set on running the XML Validator tool.

## **DISCLAIMER**

This software is provided “as is” by CalPERS without any implied support, enhancements, modifications, defect fixes and/or future releases. This software is not intended to be a CalPERS supported application now or in the future. This software is subject to change without any notice.

The software is to be used “as is”. You may use this software or not at your discretion, and CalPERS is not liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, loss of use, data, profits or business interruption)

## XML VALIDATOR FUNCTIONALITY

The XML Validator Tool is a simple Java based routine that takes a target XML file and applies a specified XML Schema (xsd) to simulate a portion of the my|CalPERS Level 1, file level, validation. For a list of validations performed by the Tool, please see the Appendix.

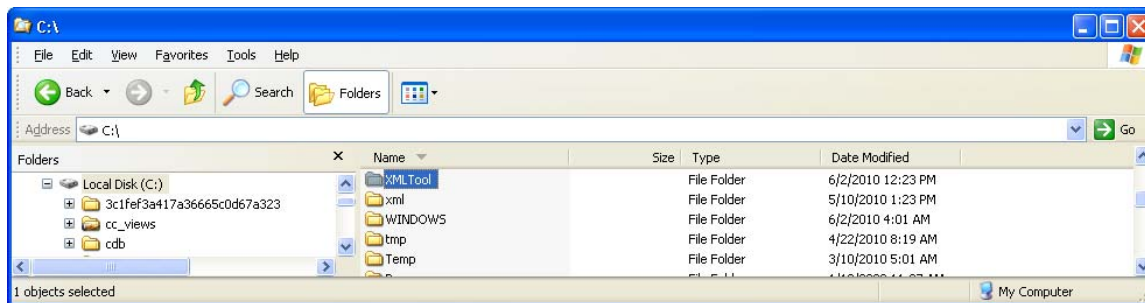
This tool may be used to assist in the development of XML files and to make sure that the XML is well-formed prior to submitting the file to my|CalPERS for processing.

The XML Validator Tool will not apply Level 2 validations. my|CalPERS Level 2 validations check the file for compliance with my|CalPERS business rules. Validating that the records in the file conform to my|CalPERS business rules takes place when the file has been accepted in to the my|CalPERS system for processing by passing Level 1 validation.

The XML Validator Tool will not correct errors and will not produce a report of all errors in the file. It is a simple command line java program that will identify some errors in the file structure. Finding the errors and fixing them will require analysis of the created XML file.

## SETUP OF XML VALIDATOR TOOL

- The XML Validator Tool is located in the Technical Toolkit found on the PERT website.
  - For employers, the Technical Toolkit is found at:  
<http://www.calpers.ca.gov/index.jsp?bc=/employer/pert/resource-library/tech-toolkit/home.xml>
  - For Direct Authorization Vendors (DAVs), the Technical Toolkit is found at:  
<https://www.calpers.ca.gov/index.jsp?bc=/business/mycalpers-readiness/direct-authorization-vendors/tech-resources/tech-toolkit/home.xml>
- Create a folder named "XMLTool" directly of your local C drive where the XML Validator Tool will be unzipped. You will be navigating to this directory from a command prompt in a later step. See below for an example of a folder named "XMLTool" created under "c:\":

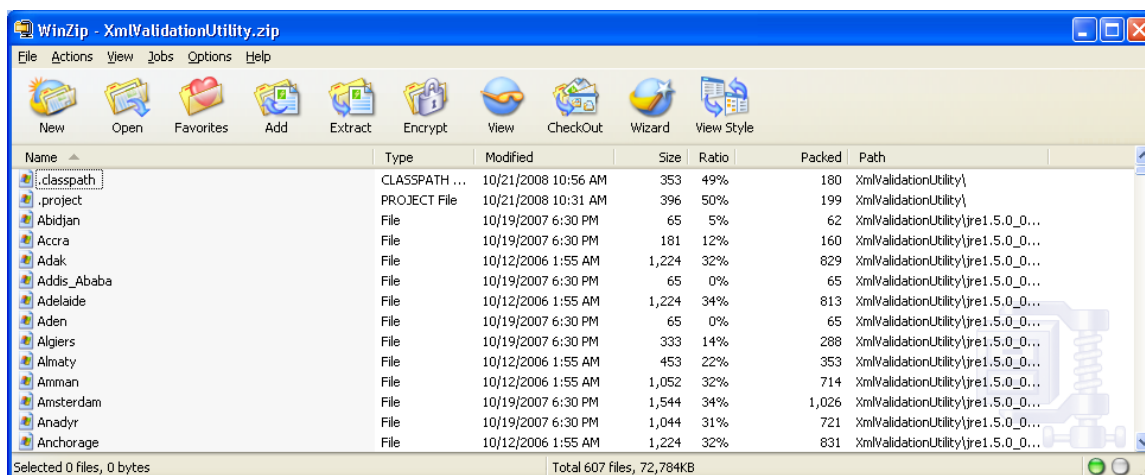


- Within the Technical Toolkit, locate the XML Validator Tool and double click on the icon. The file is named as follows:

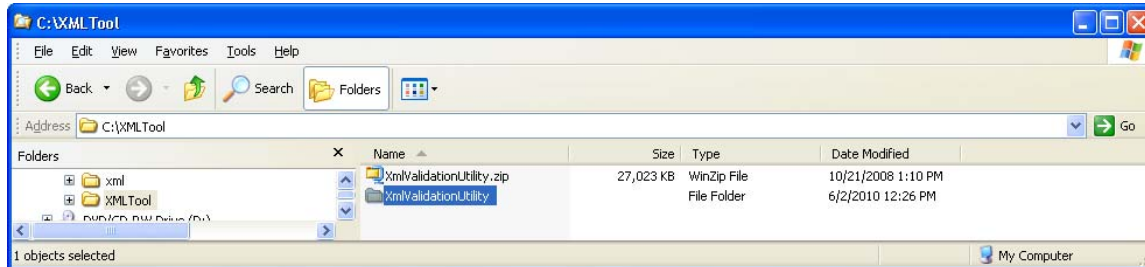


XmlValidationUtility.zip

- Double click on the file and a dialog box similar to the following will be opened. Extract all files to the folder you created in Step 2 above (C:\XMLTool):



5. In the folder you created in Step 2 (C:\XMLTool) you will now see a new folder titled “XmlValidationUtility”:

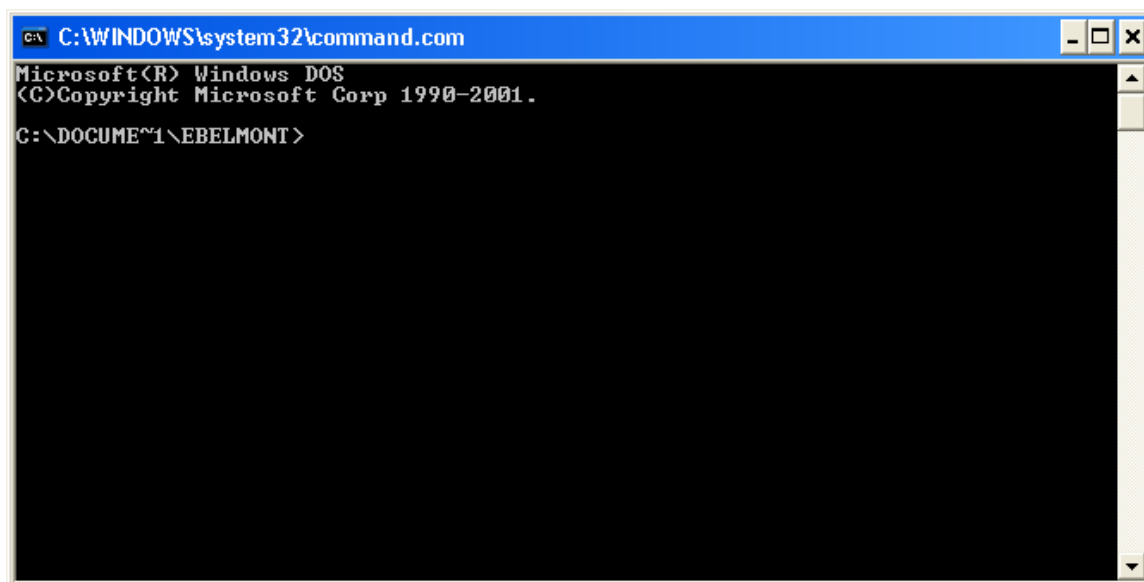


After the extract process is done the setup is now complete and you may begin to use the XML Validator Tool to validate XML Files.

## USING THE XML VALIDATOR TOOL

The XML Validator Tool is a command line based tool that runs a Java routine to validate any input XML file (i.e. Payroll File, Retirement Enrollment File, Health Enrollment File, etc.) using the specified xsd from the command line. The tool uses only two parameters, the first being the XML file that is to be validated and the second is the xsd Schema that is to be applied to the XML file. The following steps will walk you through the usage of the XML Validator Tool.

1. Open a command line on your pc by selecting “Start”, “Run” and typing “command” in the dialog box. Upon typing “command” and clicking OK you will see a windows similar to the one below:

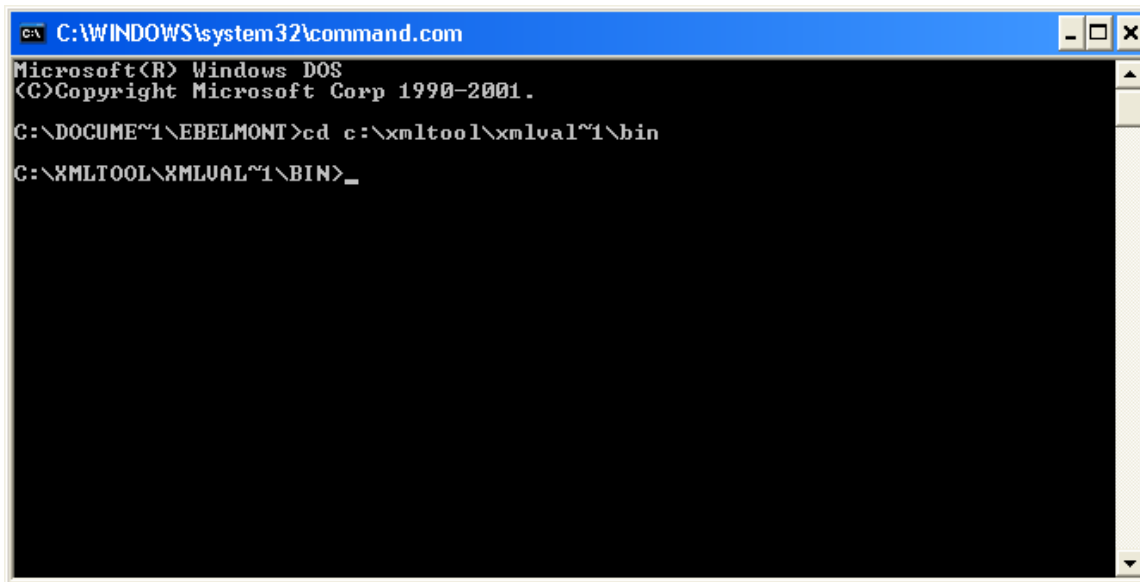


2. Using the command line navigate to the location of the xmlvalidator.cmd file. Using the example here you would type:
  - cd c:\xmltool\xmlval~1\bin



## XML VALIDATOR TOOL OUTPUT

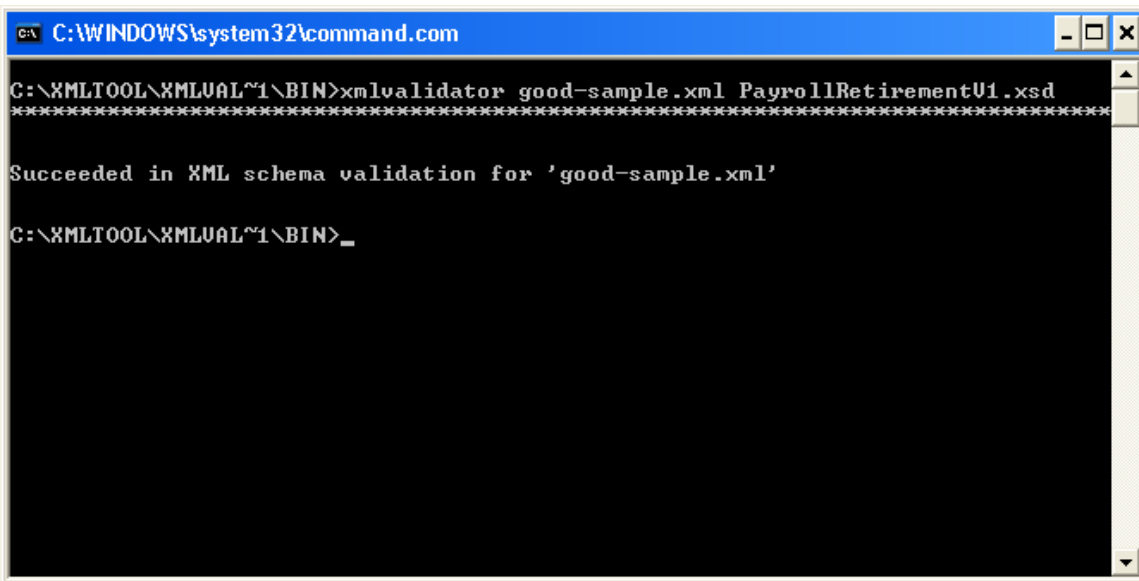
After typing the above command you will see the following:



```
C:\WINDOWS\system32\command.com
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.
C:\DOCUMENT~1\EBELMONT>cd c:\xmltool\xmlval~1\bin
C:\XMLTOOL\XMLVAL~1\BIN>_
```

3. You are now ready to run the XML Validation Tool to validate your XML File. Copy the XSD Schema you are testing against (e.g. PayrollRetirementV1.xsd) and the XML file you wish to validate to the following directory:
  - C:\XMLTool\XmlValidationUtility\bin
4. The following shows the usage and an example of how to run the XML Validator Tool after you have navigated to the appropriate directory (as shown in Step 2 above). There are two sample payroll files already located in the “bin” directory for your use. One has no errors in it and is named good-sample.xml while the other is a poorly constructed xml file (bad-sample.xml) that will throw some errors.
  - Usage:  
xmlvalidator <Xml Data File> <Xsd File>
  - Example:  
xmlvalidator good-sample.xml PayrollRetirementV1.xsd

A valid XML file will result in the following output in the command line window showing “Succeeded in XML schema validation for” the file being validated. In this case “good-sample.xml”.

A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\command.com". The command prompt shows the command "C:\XMLTOOL\XMLUAL~1\BIN>xmlvalidator good-sample.xml PayrollRetirementU1.xsd" followed by a separator line of asterisks. Below the separator, the output "Succeeded in XML schema validation for 'good-sample.xml'" is displayed. The prompt then shows "C:\XMLTOOL\XMLUAL~1\BIN>\_" indicating the command has completed.

```
C:\WINDOWS\system32\command.com
C:\XMLTOOL\XMLUAL~1\BIN>xmlvalidator good-sample.xml PayrollRetirementU1.xsd
*****
Succeeded in XML schema validation for 'good-sample.xml'
C:\XMLTOOL\XMLUAL~1\BIN>_
```

A badly constructed XML file will result in a Java dump being displayed with the resulting error. The validator tool will display one error at a time and the error would need to be corrected and the file validated again in order to check for additional errors. Subsequent errors would result in a similar output to the one below.

In the example below the bad-sample.xml file was validated and resulted in an error stating that the CalPERS ID is not valid because it does not conform to the CalPRESIdType. The CalPERSIdType is specified as 10 digits and the length of the value submitted in the file is not 10 digits.

```

C:\WINDOWS\system32\command.com

C:\XMLTOOL\XMLVAL~1\BIN>xmlvalidator bad-sample.xml PayrollRetirementV01.xsd
*****

XML file 'bad-sample.xml' failed in schema validation
psr.xml.utility.SimpleXmlValidator$XmlValidationException: org.xml.sax.SAXParseException: cvc-pattern-valid: Value '111222' is not facet-valid with respect to pattern '[0-9]{10,10}' for type 'CalPERSIdType'.
    at psr.xml.utility.SimpleXmlValidator$ExceptionHandler.error(SimpleXmlValidator.java:173)
    at org.apache.xerces.util.ErrorHandlerProxy.error(Unknown Source)
    at com.sun.org.apache.xerces.internal.jaxp.validation.ErrorHandlerAdaptor.error(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator$XSIREporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.reportSchemaError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.elementLocallyValidType(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.processElementContent(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.handleEndElement(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.endElement(Unknown Source)
    at com.sun.org.apache.xerces.internal.jaxp.validation.ValidatorHandlerImpl.endElement(Unknown Source)
    at org.apache.xerces.jaxp.JAXPValidatorComponent$XNI2SAX.endElement(Unknown Source)
    at org.apache.xerces.jaxp.JAXPValidatorComponent.endElement(Unknown Source)
    at org.apache.xerces.impl.XMLNSDocumentScannerImpl.scanEndElement(Unknown Source)
    at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContentDispatcher.dispatch(Unknown Source)
    at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(Unknown Source)
    at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
    at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
    at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
    at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
    at org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown Source)
    at org.dom4j.io.SAXReader.read(SAXReader.java:465)
    at psr.xml.utility.SimpleXmlValidator.read(SimpleXmlValidator.java:84)
    at org.dom4j.io.SAXReader.read(SAXReader.java:264)
    at psr.xml.utility.SimpleXmlValidator.main(SimpleXmlValidator.java:151)
Caused by: org.xml.sax.SAXParseException: cvc-pattern-valid: Value '111222' is not facet-valid with respect to pattern '[0-9]{10,10}' for type 'CalPERSIdType'.
    at com.sun.org.apache.xerces.internal.jaxp.validation.Util.toSAXParseException(Unknown Source)
    ... 24 more

C:\XMLTOOL\XMLVAL~1\BIN>
  
```

Display showing the error that was encountered and the value resulting in the value that caused the error to be thrown

The bad-sample.xml file contains another error and when we run the XML Validator Tool again after fixing the first error on the file, we see that the data element ReportPeriodbegindate should have a capital "B" and "D" for ReportPeriodBeginDate:

```
C:\WINDOWS\system32\cmd.exe
C:\XMLTool\XMLVAL~1\bin>xmlvalidator bad-sample-bad-date.xml PayrollRetirementU1.xsd
*****
XML file 'bad-sample-bad-date.xml' failed in schema validation
psr.xml.utility.SimpleXmlValidator$XmlValidationException: org.xml.sax.SAXParseException: cvc-complex-type.2.4.a: Invalid content was found starting with element 'n1:ReportPeriodbegindate'. One of '<'http://calpers.ca.gov/PSR/PayrollRetirementU1':ReportPeriodBeginDate' is expected.
    at psr.xml.utility.SimpleXmlValidator$ExceptionHandler.error(SimpleXmlValidator.java:173)
    at org.apache.xerces.util.ErrorHandlerProxy.error(Unknown Source)
    at com.sun.org.apache.xerces.internal.jaxp.validation.ErrorHandlerAdaptor.error(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator$XSIErrorReporter.reportError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.reportSchemaError(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.handleStartElement(Unknown Source)
    at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.startElement(Unknown Source)
    at com.sun.org.apache.xerces.internal.jaxp.validation.ValidatorHandlerImpl.startElement(Unknown Source)
    at org.apache.xerces.jaxp.JAXPValidatorComponent$XNI2SAX.startElement(Unknown Source)
    at org.apache.xerces.jaxp.JAXPValidatorComponent.startElement(Unknown Source)
    at org.apache.xerces.impl.XMLNSDocumentScannerImpl.scanStartElement(Unknown Source)
    at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContentDispatcher.dispatch(Unknown Source)
    at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(Unknown Source)
    at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
    at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
    at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
    at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
    at org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown Source)
    at org.dom4j.io.SAXReader.read(SAXReader.java:465)
    at psr.xml.utility.SimpleXmlValidator.read(SimpleXmlValidator.java:84)
    at org.dom4j.io.SAXReader.read(SAXReader.java:264)
    at psr.xml.utility.SimpleXmlValidator.main(SimpleXmlValidator.java:151)
Caused by: org.xml.sax.SAXParseException: cvc-complex-type.2.4.a: Invalid content was found starting with element 'n1:ReportPeriodbegindate'. One of '<'http://calpers.ca.gov/PSR/PayrollRetirementU1':ReportPeriodBeginDate' is expected.
    at com.sun.org.apache.xerces.internal.jaxp.validation.Util.toSAXParseException(Unknown Source)
    ... 22 more
C:\XMLTool\XMLVAL~1\bin>
```

Display showing the error that was encountered and the value resulting in the value that caused the error to be thrown

## APPENDIX A

The following errors will be detected by use of the XML Validator Tool

Validation	Level	Error Code	Error Severity	Error Message
The file failed schema validation. Please validate the file prior to submitting.	Level 1	INW00002	Error	The file or SOAP message is not XML well-formed. Validate it against the provided schema from CalPERS and submit again.
The File is empty	Level 1	COI00073	Error	Submit a file that is not empty.

Please consult the appropriate validation document for the XML schema you are validating, for a complete list of potential Level 1 and Level 2 errors that may be encountered during the formal File Readiness testing period and Production. All validation documents can be found in the Technical Toolkit