Validate Tool Configuration File

The *Validate Tool* has a large number of command options, many of which can take multiple arguments. The configuration file option allows you to code a set of complex and their arguments into a single file for re-use or just convenient editing.

File Syntax

The syntax rules for the file are similar to syntax rules for shell scripts or batch files, with some additions:

- Blank lines and lines beginning with the character '#' are ignored.
- Option lines begin with the option keyword, followed by '=', followed by the argument list.
- The argument list can be continued on the next line by ending the previous line with '\'.
- The '\' character can be escaped (in a file specification, for example) by using two in succession ("\\")
- On Windows machines, the usual directory separator '\' can either be escaped, or replaced with '/'.

Option Keywords

All option keywords begin with "*validate.*" and must be followed by an equal sign ('=') and the value. Space around '=' is not significant. Keywords are listed in alphabetical here. For details on the referenced command line options, see the Using Validate Tool page, where the options are listed by functional group.

validate.basePath

This option keyword is equivalent to the **-base-path** command line option.

> **Note:** *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the* ***-checksum-manifest*** *command option will only supersede what is in the configuration file if the* ***-checksum-manifest*** *option comes* after *the* ***-config*** *option on the command line.*

validate.catalog

This option keyword is equivalent to the **-catalog** command line option, and thus largely useless. In general, using the *validate.schema* and *validate.schematron* or equivalent command line options will be simpler. The exception might be in the case where a catalog file is used to resolve a reference to a core ("pds") schema file that cannot be referenced in any other way because of various constraints or issues with other options. This has not been tested.

validate.checksum

This keyword is used to specify an MD5 checksum manifest file for integrity checking.

> **Note:** *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the* ***-checksum-manifest*** *command option will only supersede what is in the configuration file if the* ***-checksum-manifest*** *option comes* after *the* ***-config*** *option on the command line.*

validate.force

This keyword takes a boolean argument. Setting it to "true" causes the same behavior as specifying the **-force** command line option - directory recursion is suppressed. Values indicating "true" are: **True**, **true**, **Yes**, and **yes**. Values indicating "false" are: **False**, **false**, **No**, and **no**. Anything else is a syntax error. The default value for this option is "true".

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the **-force** command option will only supersede what is in the configuration file if the **-force** option comes* after *the **-config** option on the command line.*

validate.local

This keyword takes a boolean argument. Setting it to "true" causes the same behavior as specifying the **-local** command line option - directory recursion is suppressed. Values indicating "true" are: **True**, **true**, **Yes**, and **yes**. Values indicating "false" are: **False**, **false**, **No**, and **no**. Anything else is a syntax error. The default value for this option is "true".

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the **-local** command option will only supersede what is in the configuration file if the **-local** option comes* after *the **-config** option on the command line.*

validate.model

The option keyword takes the same values as the corresponding **-mode-version** command line option.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the **-model-version** command option will only supersede what is in the configuration file if the **-model-version** option comes* after *the **-config** option on the command line.*

validate.regexp

This keyword takes a list of file-globbing patterns, each enclosed in double quotes, to select file for validation. The input syntax and limitations are identical to those defined for the **-regexp** command option.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the **-regexp** command option will only supersede what is in the configuration file if the **-regexp** options comes* after *the **-config** option on the command line.*

validate.report

This keyword takes a filename argument and directs the standard report output to the named file.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the **-report-file** command option will only supersede what is in the configuration file if the **-report-file** option comes* after *the **-config** option on the command line.*

validate.reportStyle

This keyword takes one of the values **full**, **json**, or **xml**. The default is **full**.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the* ***-report-style*** *command option will only supersede what is in the configuration file if the* ***-report-style*** *option comes* after *the* ***-config*** *option on the command line.*

validate.rule

This keyword option takes the same values as the *-rule* command line option: **pds4.label**, **pds4.folder**, **pds4.collection**, **pds4.bundle**, and **pds3.volume**.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the* ***-rule*** *command option will only supersede what is in the configuration file if the* ***-rule*** *option comes* after *the* ***-config*** *option on the command line.*

validate.schema

This keyword is supposed to be the equivalent for the **-schema** option on the command line, but it is not. It can be used to provide references to schema files for discipline and local dictionaries, however. As with the command line **-schema** option, you should also use the *validate.schematron* option to indicate the corresponding Schematron file to get full schema validation.

Note:' This option cannot *be used to supply an alternate schema location for the core ("pds") namespace. Doing so will not produce an error, and the schema indicated will be listed as the "User Specified Schema", but it* ***will not*** *be applied. You can use the* ***-model-version*** *command-line option to override the default schema. Attempting to use the* ***-schema*** *command line option with the* ***-config*** *option will also cause the referenced core schema to be listed but ignored.*

validate.schematron

This keyword is the equivalent of the **-schematron** command line option.

Note: *Unlike other option keywords, this option keyword* ***is not*** *overridden by the equivalent command-line option irrespective of the actual command line sequence. The total cumulative list of all Schematron files named in options is applied to labels - once for each time the file is referenced. There is one exception - if a Schematron for the core namespace is included in the list (as determined by the file contents), the default Schematron file for the core namespace is not applied.*

validate.target

This keyword takes the same sort of value list as the **-target** option. This keyword is ignored if targets are supplied for either the *validate* command or the **-target** option.

validate.verbose

This keyword takes one of the values: **1**, **2**, or **3**, indicating the severity level of messages reported in the "Validation Details" list in the output report.

Note: *Contrary to what is stated in the* Validate Tool - Operation *document, the explicit use of the* ***-verbose*** *command option will only supersede what is in the configuration file if the* ***-verbose*** *option comes* after *the* ***-config*** *option on the command line.*

Usage Notes

- Unlike options on the command line, if a boolean option keyword is repeated in the configuration file it is the *first* occurrence in the configuration file that takes precedence, rather than the last.
- In all cases tested, command line options override configuration options *only* when the **-config** option *precedes* the overriding option on the command line.
- While it is an error to list multiple configuration files in a single **-config** option, it is not a syntax error to use the **-config** option multiple times in the same command line. This is probably a bad idea. The various configuration specifications seem to be combined in a way that is difficult to track or predict.