VPA

Manual

(version 0.1)

Public Record Office Victoria

Copyright 2021, Public Record Office Victoria

Further copies of this document can be obtained from the PROV Web site  
<http://www.prov.vic.gov.au/>

The State of Victoria gives no warranty that the information in this version is correct or complete, error free or contains no omissions. The State of Victoria shall not be liable for any loss howsoever caused whether due to negligence or otherwise arising from the use of this Specification.

|  |  |  |
| --- | --- | --- |
| Version | Version Date | Details |
| 1.0 | 22 May 2006 | Released |

Contents

[V2Check 1](#_Toc35431771)

[Manual 1](#_Toc35431772)

[(version 3.0) 1](#_Toc35431773)

[Public Record Office Victoria 1](#_Toc35431774)

[1. Purpose of Document 4](#_Toc35431775)

[2. Overview of running VEO Check II 4](#_Toc35431776)

[3. Detailed invocation instructions 4](#_Toc35431777)

[3.1. Requirements 4](#_Toc35431778)

[3.2. Installation of V2Check 4](#_Toc35431779)

[3.2.1. Installing V2Check 4](#_Toc35431780)

[3.2.2. Installing Java 4](#_Toc35431781)

[3.2.3. Configuring V2Check 4](#_Toc35431782)

[3.3. Running VEOCheck3 4](#_Toc35431783)

[3.3.1. Placement of VEOs 4](#_Toc35431784)

[3.3.2. Test options 5](#_Toc35431785)

[3.3.3. Running V2Check from the command line 5](#_Toc35431786)

[3.3.4. Running V2Check from the V2Check.bat file 5](#_Toc35431787)

[3.3.5. Run the test script 6](#_Toc35431788)

[4. Interpreting the output 6](#_Toc35431789)

[4.1. Preamble 6](#_Toc35431790)

[4.2. Test results 6](#_Toc35431791)

[4.3. No errors 6](#_Toc35431792)

[4.4. XML Errors 6](#_Toc35431793)

[4.5. Value Errors 7](#_Toc35431794)

[4.5.1. Empty Elements 7](#_Toc35431795)

[4.5.2. VAL1 Errors - Element Errors 8](#_Toc35431796)

[4.5.3. VAL2 Errors - Attribute Errors 8](#_Toc35431797)

[4.5.4. VAL3 Errors - Element Value Errors 8](#_Toc35431798)

[4.5.5. VAL4 Errors - Mandatory Element in a Version 2 VEO missing 9](#_Toc35431799)

[4.5.6. VAL5 Errors - Mandatory Element in a Version 1 VEO missing 9](#_Toc35431800)

[4.5.7. VAL6 Errors - Mandatory Element missing 9](#_Toc35431801)

[4.5.8. VAL7 Errors - Version 2 feature in a Version 1 VEO 10](#_Toc35431802)

[4.5.9. VAL9 Errors - Missing mandatory attribute in a Version 2 VEO 10](#_Toc35431803)

[4.5.10. VAL10 Error – No long term preservation format 10](#_Toc35431804)

[4.6. Signature Errors 11](#_Toc35431805)

[4.6.1. SIG8 Errors -- Signature or Lock Signature verification failures 11](#_Toc35431806)

[4.6.2. SIG9 -- Certificate verification failures 12](#_Toc35431807)

[4.7. Encoding Content 12](#_Toc35431808)

[4.8. Full Value and Attribute Dumps 12](#_Toc35431809)

[4.8.1. List of Attributes 13](#_Toc35431810)

[4.8.2. List of Elements 14](#_Toc35431811)

# Purpose of Document

This document describes how to run the tool that will test V2 & V3 VEOs for compliance with the standards. This tool uses V2Check and VEOAnalysis.

Using the VPA tool has the advantage that it is not necessary to learn how to use the two other tools. It has the further advantage of being the tool that the new (2021) DA uses.

# Overview of running VPA

To test VEOs, the user will

* Install the VPA tool (this will normally have already been done)
* Place all the VEOs in a single directory
* Copy the VPA.bat file into the directory
* Edit the VPA.bat file to change the command line arguments (if required)
* Start a MS-DOS command window
* Run the VPA.bat file in command window

# Detailed invocation instructions

## Requirements

VPA requires

* Java 1.8 (later version of Java have not been tested)

## Installation of VPA

### Installing V2Check

Download VERS-V2-Package.zip from the VERS support page at prov.vic.gov.au

Extract the java source code to a suitable place on your hard drive.

### Installing Java

VPA is a Java program and you need to have a Java 1.8 runtime environment or software development kit installed on your computer.

### Configuring VPA

VPA can be run from the command line in an MS-DOS window, but an alternative is to run it using a MS-DOS bat file (VPA.bat) included in the Zip file. The default contents of the bat file are shown below:

@echo off

if exist "J:/PROV/TECHNOLOGY MANAGEMENT/Application Development/VERS/VERSCode" (

set code="J:/PROV/TECHNOLOGY MANAGEMENT/Application Development/VERS/VERSCode"

) else (

set code="C:/Users/Andrew/Documents/Work/VERSCode"

)

java -classpath %code%/VPA/dist/\* VPA.DAIngest -s %code%/VPA/support %\*

Before running this script, the value of the versclasspath variable must be changed to the directory in which VPA was installed.

The value of the command line flags can be changed to customise the tests performed.

## Running VPA

### Placement of VEOs

All the VEOs to be tested need to be placed in a single directory. The VEOs can be either V2 or V3 VEOs (or a mixture).

### Test options

The VPA test tool has a number of options that control the tests that are performed on the VEOs. These options are:

* *-v* (Optional) Produce more details about the results of the tests.
* -rs (Optional) Produce a summary of the results generated by testing the VEOs. Each error or warning generated will be listed in the summary, together with the files that generated the error.
* -o <directory> (Optional) Specify the output directory. This is optional, and if not set the output will be generated in the current working directory (the directory in which the VPA tool is started).
* -s <directory> Specify the directory in which the support materials are located – for example the file containing the valid long term sustainable formats, the V2 DTD, and the V3 XML schema. This option must be present, but it is set in the VPA.bat script and would only need to be changed if the VPA tool is being run outside the PROV/BCS environment.
* -lite (Optional) If present, the VPA will only validate the VEO; it will not attempt to extract the content and make the packages that would normally be passed to the AMS, SAMS, and DAS. Normally, this option would be set if you are using the VPA to validate VEOs unless you wish to examine (open) the files contained within the VEOs.
* -sample <p> (Optional) If present, this will sample the available VEOs rather than testing each one. This is useful if you wish to do an initial investigation into problems with a large set, rather than an exhaustive analysis which may take a considerable time. The argument <p> is a number between 0.0 and 1.0 that controls the probability that an individual VEO will be selected for testing – for example ‘-sample 0.01’ means that roughly one in every 10 VEOs will be tested.
* -ignoreAbove <size> (Optional) This is another option to limit the time taken to test a collection of VEOs. If present, this option instructs the VPA to ignore VEOs above a certain size; large VEOs are slower to check. Size is expressed as an integer in kilobytes (KB) – the same size that Windows Explorer uses to express file sizes.
* -m (Optional) Migration mode. If present, the VPA will not test the values in a V2 VEO. This mode was added for migration of the V2 VEOs from the old DA to the new and reflects a gradual tightening of requirements. It should not normally be used.

Note that V2Check (V2) and VEOAnalysis (V3) have additional test options that are not available when using the VPA tool. These test options may be useful for tracking down the causes of errors. The VPA does a full test for compliance, however.

### Running VPA from the command line

VPA can be run from a MS-DOS (or Unix/Linux) command line. This requires the Java classpath to be configured correctly.

### Running VPA from the VPA.bat file

Copy the VPA.bat file from the VPA directory into the directory that contains the VEOs.

Edit the VPA.bat file to change the test options. The ‘-s’ option in the bat file should not be changed unless it is necessary to run the VPA outside the PROV/BCS environment.

### Run the test script

Using a command window run the VPA.bat file.

You can test

* one VEO at a time (e.g. ‘VPA test.veo’)
* multiple VEOs by listing individual VEOs (e.g. ‘VPA test1.veo test2.veo’)
* multiple VEOs by using the wildcard ‘\*’ or ‘?’ expansions (e.g. ‘VPA \*.veo’)
* multiple VEOs by specifying one or more containing directories (e.g. ‘VPA transfer1’ which would test all VEOs in the directory ‘transfer1’, including any subdirectories.

The test results are printed to standard out. By default they will appear in the command window. If you a more permanent copy of the output, redirect the standard output to a text file. For example: ‘VPA \*.veo > results.txt’

# Interpreting the output

The VPA tool processes VEOs by calling the V2Check and VEOAnalysis tools. The output is consequently an amalgam of the output of those two tools. No attempt was made to make the output of the two tools consistent.

## Preamble

The standard preamble to the report produced by the VPA has the following appearance:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* V E O T E S T I N G T O O L \*

\* \*

\* Version 1.0 \*

\* Copyright 2021 Public Record Office Victoria \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Test run:

2021-03-04 16:11:51+10:00

INFO: Verbose/Debug mode is selected

INFO: Producing a summary of results?: 'yes'

INFO: Using real handle service: 'false'

INFO: RDF Identifier prefix is 'null'

INFO: Source directory is '.'

INFO: Output directory is 'f:\output'

INFO: User id to be logged: 'Andrew'

INFO: Light mode

INFO: \*\*\*Processing directory 'f:\test'

INFO: \*\*\*Processing directory 'f:\test\V3'

The header identifies the VERS testing tool being used and the version. It is followed by the date and time the test is run and (optionally) the command line options set. The command line options are only reported if the ‘-v’ command line flag is set.

## Test results

The preamble is followed by a set of test results. One set is produced for each VEO tested.

Two typical result are as follows:

INFO: Processing 'f:\test\V3\v3-004-minMD-as5478.veo.zip'

2021-03-04T17:11:51.799 Processing: 'f:\test\V3\v3-004-minMD-as5478.veo.zip' (2638 mS) 7435176

INFO: Processing 'f:\test\V3\1-1-XNotVEO.veo.zip'

2021-03-04T17:11:55.009 Processing: 'f:\test\V3\1-1-XNotVEO.veo.zip' (199 mS) 4503960

FAILED (V3 VEO 2021-03-04T06:11:55.198Z to 2021-03-04T06:11:55.213Z)

ZIP format error in opening Zip file: error in opening zip file (RepnVEO.unzip())

In this case each result commences with one or two lines. The first line commences ‘INFO:’ (this line will only appear if the ‘-v’ command line option is selected). The second line will always be present and commences with a date time stamp. Both lines state the VEO file being processed. The second line gives the time taken to process the VEO (in milliseconds).

These header lines are followed by any diagnostics produced in processing the VEO. *If there were no issues identified, no diagnostic lines will be generated.*

So, in the first example (v3-004-minMD-as5478.veo.zip) the VEO was processed without any errors, while in the second an error was reported.

## XML Errors

Before any other tests are performed, the VEO is always parsed. This test will fail if

* The VEO is not valid XML
* The VEO does not conform to the VERS DTD.

V2Check will not detect if the VEO is not correctly encoded in UTF-8.

Typical parsing errors are:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '0-DTDExtension-7a.veo'

Test FAILED: SAXException: Fatal Error: URI=http://www.prov.vic.gov.au/vers/standard/ Line=3:Document root element is missing.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-DTDValidation-13a.veo'

Test FAILED: SAXException: Error: URI=http://www.prov.vic.gov.au/vers/standard/ Line=193:Element "vers:RecordMetadata" does not allow "naa:Language" here.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Errors picked up by the parser are:

* Mis-ordered elements in the VEO (XML document)
* Missing mandatory elements or attributes in the VEO (XML document)
* Additional (non-standard) elements or attributes in the VEO (XML document)

The line numbers reported by V2Check in the parser errors are not useful unless the ‘-parseVEO’ option is used.

When the ‘-parseVEO’ option is selected, V2Check will parse the original VEO. If this option is not selected (the normal behaviour), V2Check will make a copy of the original VEO that does not contain the document data, and then parse the copy. This is 10 to 100 times faster than parsing the original VEO, but means that the line numbers reported in parse errors do not relate to the original VEO. We recommend that you normally run V2Check without the ‘-parseVEO’ option for speed. If a parsing error is reported, rerun V2Check on the failed VEO with the ‘-parseVEO’ option set.

Unfortunately, even with the ‘-parseVEO’ option set, the line numbers may not be accurate as the parser does not always generate accurate line numbers.

## Value Errors

V2Check will check the values contained in certain elements for consistency if the ‘-values’ or ‘-all’ options are selected.

If the ‘-oneLevel’ option is selected, only the outermost (current) layer of a ModifiedVEO or onion VEO is tested. Inner (older) layers are not tested. The default is to test all layers.

If the ‘-verbose’ option is set the values of the elements and attributes will be printed out. This is described in section 4.8.

Note that the Agency Identifier, Series Identifier, and File Identifiers in the VEO cannot be checked for correctness. If these identifiers are incorrect, the VEOs cannot be ingested into the Digital Archive even though V2Check will pass the VEOs.

A typical value error is:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-ElementContent-43ab.veo'

Test FAILED: EMPTY VALUES: The VEO contains the following empty elements:

<vers:DocumentDate>

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL3: Error in value of element <vers:DocumentDate> (M123). Value is <empty>

------------------------------------------------------^ Year must match 'yyyy'

Note that in this example, the same VEO problem is being reported twice; once as an empty element and then because the date does not match the required format.

### Empty Elements

The V2Check will detect elements that have a blank value:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-ElementContent-43aa.veo'

Test FAILED: EMPTY VALUES: The VEO contains the following empty elements:

<vers:DocumentTitle>

A blank value is defined as an

* empty element (e.g. ‘<vers:DocumentTitle/>’)
* element that does not contain any content (e.g. ‘<vers:DocumentTitle></vers:DocumentTitle>’)
* element that only contains white space (spaces, tabs, carriage returns or line feeds) (e.g. ‘<vers:DocumentTitle> </vers:DocumentTitle>’)

### VAL1 Errors - Element Errors

Errors marked as ‘VAL1’ are concerned with element errors. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-VersionAttribute-25b.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL1: Error in element <vers:SignedObject> (M4)

A version 2.0 <vers:SignedObject> (M4) must contain a vers:VEOVersion attribute

In this case, the error is that a vers:SignedObject element in a version 2 VEO must contain a vers:VEOVersion attribute.

The description will contain information about the error that has occurred; in this case that a vers:SignedObject element in a Version 2 VEO does not contain a vers:VEOVersion attribute. The ‘M number’ is the reference to the element definition in VERS Specification 2.

### VAL2 Errors - Attribute Errors

Errors marked as ‘VAL2’ concern an error with an attribute value. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-VersionAttribute-25a.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL2: Error in attribute vers:VEOVersion (value='1.2') in element <vers:SignedObject> (M4)

which must be '2.0' to match <vers:Version> (M3) element

The first portion of the error message indicates the attribute that caused the error, the value of the attribute, and the element in which the attribute is located. The ‘M number’ is the reference to the element definition in VERS Specification 2.

This is followed by a detailed description of the problem with the value.

### VAL3 Errors - Element Value Errors

Errors marked as ‘VAL3’ concern an error with an element value. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '0-VEOType-30j.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL3: Error in value of element <vers:ObjectType> (M6). Value is 'File'

The value of the <vers:ObjectType> (M6) element in a <vers:RevisedVEO> (M158) element must match the value of the vers:OriginalVEOType attribute in the <vers:ModifiedVEO> (M156) element (which was Record)

The first portion of the error message indicates the element in which the error is located, and the value of the element. Once again the M number is the reference to the element definition in the VERS Specification 2.

This is followed by a detailed description of the problem. In this case it is a problem with consistency between the value of vers:ObjectType and the vers:OriginalVEOType attribute.

### VAL4 Errors - Mandatory Element in a Version 2 VEO missing

Errors marked as ‘VAL4’ are concerned with elements that are mandatory in a Version 2 VEO, but which are not present in the VEO. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-ElementContent-43az.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL4: Element that is mandatory in a version 2 VEO is missing

A version 2.0 VEO must contain at least one <vers:LockSignatureBlock> (M152) element

The VERS DTD covers both version 1 and version 2 VEOs. Consequently, features added in version 2 must be marked as ‘optional’, even if they are actually mandatory in a version 2 VEO (otherwise version 1 VEOs could not conform to the DTD). The description of the error lists the missing element and includes a reference to the definition of the element in the VERS specification.

The version of the VEO is normally detected automatically using the vers:Version element that is present at the start of every VEO. If necessary, the ‘-v1.2’ or ‘-v2’ options can be used to force V2Check to validate against a particular version.

### VAL5 Errors - Mandatory Element in a Version 1 VEO missing

Errors marked as ‘VAL5’ are concerned with elements that are mandatory in a Version 1 VEO, but which are not present in the VEO. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-V1Encoding-29a.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL270: Document without Long Term Preservation Format

A <Document> (M114) element must contain an <Encoding> (M126) element with a valid long term preservation format for acceptance into the digital archive. The Document has no <vers:RenderingKeywords> (M132) elements and so no long term preservation formats can be identified

VAL5: Element that is mandatory in a version 1 VEO is missing

In a version 1 VEO, a <vers:Document> (M114) element must contain at least one <vers:Encoding> element

The VERS DTD covers both version 1 and version 2 VEOs. Consequently, features added in version 1 must be marked as ‘optional’, even if they are actually mandatory in a version 1 VEO (otherwise version 2 VEOs could not conform to the DTD). The description of the error lists the missing element and includes a reference to the definition of the element in the VERS specification.

The version of the VEO is normally detected automatically using the vers:Version element that is present at the start of every VEO. If necessary, the ‘-v1.2’ or ‘-v2’ options can be used to force V2Check to validate against a particular version.

### VAL6 Errors - Mandatory Element missing

Errors marked as VAL6 mark elements that are optional according to the DTD, but are required by the VERS standard when the VEOs are submitted to PROV. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-XMLDeclaration-3b.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL6: Missing mandatory element

A <vers:VEOIdentifier> (M99) element must contain a <vers:AgencyIdentifier> (M100) element when submitted to PROV

VAL6: Missing mandatory element

A <vers:VEOIdentifier> (M99) element must contain a <vers:SeriesIdentifier> (M101) element when submitted to PROV

In this case the vers:AgencyIdentifier and vers:SeriesIdentifier are required when the VEOs are submitted to PROV.

The description includes the missing element and includes the reference to the element definition in the VERS specification 2.

### VAL7 Errors - Version 2 feature in a Version 1 VEO

Errors marked as VAL7 indicate version 2 features that have been found in a version 1 VEO. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-DTDValidation-48a.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL7: Version 2 feature in a version 1 VEO

A version 1 <vers:SignatureBlock> (M134) element cannot contain a vers:id attribute

VAL7: Version 2 feature in a version 1 VEO

A version 1 VEO cannot contain a <vers:LockSignatureBlock> (M152) element

The version of the VEO is normally detected automatically using the vers:Version element that is present at the start of every VEO. If necessary, the ‘-v1.2’ or ‘-v2’ options can be used to force V2Check to validate against a particular version.

### VAL9 Errors - Missing mandatory attribute in a Version 2 VEO

Errors marked as VAL9 indicate a missing mandatory attribute in a version 2. VEO. For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '1-V2LockSignatureBlock-27b.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL9: Missing mandatory attribute in a version 2 VEO

A <vers:LockSignatureBlock> (M152) element must contain a vers:signsSignatureBlock attribute

The version of the VEO is normally detected automatically using the vers:Version element that is present at the start of every VEO. If necessary, the ‘-v1.2’ or ‘-v2’ options can be used to force V2Check to validate against a particular version.

### VAL10 Error – No long term preservation format

Errors marked as VAL10 indicate that one or more documents in the VEO lack a valid long term preservation format (currently a PDF, TIFF, JPEG, or text encoding of the document). For example:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-V1PreservationFormat-21b.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL10: Document without Long Term Preservation Format

A <Document> (M114) element must contain an <Encoding> (M126) element with a valid long term preservation format for acceptance into the digital archive. Formats found in this Document are: '.b64; .doc'

The PROV digital archive will only accept VEOs which contain a valid long term preservation format for each document in the VEO. That is, each document in the VEO must contain a PDF, TIFF, JPEG, or text encoding (it may contain other encodings as well).

The formats are extracted from the vers:RenderingKeywords (M131) element. If this is not present in the encodings, V2Check will generate a long term preservation format error, even if a valid format exists. This can easily be determined by the error message:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '2-V1RenderingKeywords-31a.veo'

Test FAILED: INVALID VALUES: The VEO contains the following invalid elements:

VAL10: Document without Long Term Preservation Format

A <Document> (M114) element must contain an <Encoding> (M126) element with a valid long term preservation format for acceptance into the digital archive. The Document has no <vers:RenderingKeywords> (M132) elements and so no long term preservation formats can be identified

VAL6: Missing mandatory element

A <vers:RenderingKeywords> (M132) element must be present in each <vers:Encoding> (M126) element to allow automated extraction

## Signature Errors

V2Check will check the signatures and lock signatures associated with a VEO if the ‘-signatures’ or ‘-all’ options are set. If the ‘-oneLevel’ option is selected, only the signatures on the outermost (current) layer of a ModifiedVEO or onion VEO are tested. The signatures on inner (older) layers are not tested.

### SIG8 Errors -- Signature or Lock Signature verification failures

Errors marked as SIG8 indicate that verification of a digital signature or a lock signature failed. A typical error report for a signature or lock signature verification failure is:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '1-ValidateSignature-42a.veo'

Test FAILED: SIGNATURE: Testing the following signature failed:

SIG8: Signature verification failed for Signature (vers:id="Revision-1-Signature-1")

Signature failed verification

Signature (base64): eUS7r/o0Y7Ji9jEm79VnJHetONO/Ch/siSSRQx+JGgjgxjHXqvcbP+QnPSBzMfxlbqAxriW6QKidquNDLV5tDsWM8QTsC2dcE3338jIoRzo083o019eX5eKMHm0ICH7j1zXV3NoXTjwA0f1zns2u7RJBDmi9NNdkLuAr3TvG4nQ=<

Signature (hex): 7944BBAFFA3463B262F63126EFD5672477AD38D3BF0A1FEC892491431F891A08E0C631D7AAF71B3FE4273D207331FC656EA031AE25BA40A89DAAE3432D5E6D0EC58CF104EC0B675C137DF7F23228473A34F37A34D7D797E5E28C1E6D08087EE3D735D5DCDA174E3C00D1FD739ECDAEED12410E68BD34D7642EE02BDD3BC6E274

Hash of signed object: CD9481794517411B526A069E0290A0C63146DD9C

Certificate: Subject: CN=Tester, O=PROV, C=AU issued by: O=PROV, C=AU

[

[

Version: V3

Subject: CN=Tester, O=PROV, C=AU

Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

Key: SunJSSE RSA public key:

public exponent:

010001

modulus:

d26f4ca7 d360ee54 31835f49 5ef0dea5 474f358c d4193811 2247ff08 f401f0ee

1a9e2fb6 72f467bf 6aafbf1d d38b5ca4 00033846 1eb19349 4273af56 5988ec9d

76cc71f9 1ce00d7d 10bbfdb3 1f0e3fd4 4e136312 bae9c657 7ea05a2f f96314f6

ba4b7dbb 422280b4 e73100a1 3b7ba02f 9e28f1ad 815213be 99663896 ead78287

Validity: [From: Mon Dec 06 12:36:47 EST 2004,

To: Mon Dec 06 15:23:27 EST 2004]

Issuer: O=PROV, C=AU

SerialNumber: [ 11]

]

Algorithm: [SHA1withRSA]

Signature:

0000: 4A 6D DD A1 CF CD 4E 7E 5C E0 F3 C2 78 A3 51 B4 Jm....N.\...x.Q.

0010: 5E 68 86 8D 98 79 B3 3E F8 75 D3 6A 13 3A FF C5 ^h...y.>.u.j.:..

0020: 6A 8B 64 F1 77 96 A8 E9 2C 22 23 37 56 57 31 35 j.d.w...,"#7VW15

0030: F8 EE 31 D2 99 E2 9C D7 D2 86 80 F2 C4 E7 53 2A ..1...........S\*

0040: 58 5E 48 C5 8B 96 A3 C5 A0 E4 1F E5 86 CF A1 4E X^H............N

0050: 43 17 94 C0 98 D7 EC 7B D0 91 3F E7 D0 0F CE E5 C.........?.....

0060: 8C 99 F7 BA B7 21 FD 27 BA C4 25 50 48 C8 49 3E .....!.'..%PH.I>

0070: 9A A5 F9 BA BF 06 52 32 37 69 20 F2 14 27 00 64 ......R27i ..'.d

]

When a digital signature fails verification, there is no way of determining whether:

* The signed object has been modified
* The signature itself is wrong (e.g. has been calculated incorrectly or been corrupted)
* The public key (from the certificate) is invalid

For this reason, the report of a digital signature includes the following information:

* The vers:id attribute of the failed signature. This can be used to identify which signature failed
* The signature value encoded in Base64. This can be used to confirm which signature failed verification (particularly in a version 1 VEO which does not contain vers:id attributes).
* The signature value in hexadecimal.
* The hash value in hexadecimal. This is the result of applying the nominated hash algorithm to the signed object (this hash value is encrypted using the private key to give the digital signature). If the original hash value, calculated when signing the VEO, is known it can be compared with this value. If the two values are different it suggests that the cause of the verification failure is the use of the wrong hash algorithm, or, more likely, that the wrong portion of the VEO has been signed. If the two values are the same, it suggests that either the wrong digital signature algorithm was used, or, more likely, the wrong public key (certificate) was used, or the VEO has been corrupted.
* The certificate used to obtain the public key.

### SIG9 -- Certificate verification failures

Errors marked as SIG9 indicate that verification of a certificate failed. A typical error report for a certificate verification failure is:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing '1-DigitalCertificates-17d.veo'

Test FAILED: SIGNATURE: Testing the following signature failed:

SIG9: Signature verification failed for Signature (vers:id="Revision-1-Signature-1")

Certificate 0 failed verification

Subject of certificate is: CN=Tester, O=PROV, C=AU

Issuer of certificate is: O=PROV, C=AU

Each certificate error contains three basic pieces of information:

* The identity of the certificate that failed (0 is the first certificate in the vers:CertificateBlock)
* The subject of the certificate (i.e. the person or organisation that owns the public key in the certificate)
* This issuer of the certificate (i.e. the organisation that signed the certificate).

If the ‘-verbose’ option is set the contents of the failed certificate is displayed in the error message.

## Encoding Content

If the ‘-extract’ option is set, V2Check will extract and decode the contents of the document data elements.

The contents will be placed into files. The base name of the file will be the contents of the vers:id attribute, with the last file extension taken from the vers:RenderingKeywords element. For example: Revision-1-Document-1-Encoding-1-DocumentData.pdf. Note that if multiple VEOs are tested in one run, the test of a VEO may overwrite the contents of earlier VEOs. Consequently this option is only of use if a single VEO is being tested.