Version number: 2.0

Issue Date: 13 May 2020

# Introduction

## Purpose

The purpose of this procedure is to describe how to use the Easy VEO Creator tool to create Version 3 VEOs.

## Background

Public Record Office Victoria (PROV) has adopted Version 3 (V3) VEOs (VERS Encapsulated Objects) as the standard mechanism to transfer digital records to PROV. This requires agencies to create V3 VEOs. This tool is provides a simple mechanism to construct V3 VEOs if the digital records are represented as files in a computer file system.

The Easy VEO Creator GUI tool is in development, and the current version is primarily released to gain feedback.

## Audience

This procedure has been developed for:

* Agency staff to create V3 VEOs before moving digital objects to PROV
* PROV staff needing to create V3 VEOs

# Easy VEO Create tool

## What is the Easy VEO Create tool?

The Easy VEO Create is a tool developed by PROV that allow the easy creation of VERS V3 VEOs.

## How is the Easy VEO Create tool used?

The Easy VEO Create tool may be run as Java executables or called using an Application Programming Interface (API).

The tools must be run using Java 1.8 or later. They will not work with Java 1.7 or earlier versions.

On a computer operating a Windows operating system, the tools can be invoked using the ‘cmd.exe’ program.

## Legal

The toolset is licensed under the Creative Commons CC BY 4.0 license. This means that you have a license to do anything that you want with the toolset, provided that you:

* Acknowledge Public Record Office Victoria as the source of the toolset.
* Do not misrepresent the license or your relationship with Public Record Office Victoria.

Specifically, you may:

* Include the code from the toolset in your products, either “as is” or in a modified format.
* Use the code from the toolset as the basis of code in your toolset.

# Creating V3 VEOs using the GUI

## Preparation work

Before running the tool it is necessary to structure the record content in the file system and add files containing the metadata. The structure of the generated VEO reflects the structure of the record content in the file system.

### Step 1 – Create a directory for the content

The first step is to create a directory that will contain all of the content and metadata for all the VEOs that will be created. It does not matter where this directory is located.

### Step 2 – Create sub-directories for each VEO

Within the directory create a subdirectory for each VEO that you wish to create. The name of the subdirectory will be used as the name of the created VEO (e.g. a subdirectory named ‘Email 2020’ will cause the production of a VEO with the name ‘Email 2020.veo.zip’.

### Step 3 – Copy the record content for each VEO into the appropriate subdirectory

Copy the content that you wish to include in each VEO into the appropriate subdirectory (e.g. all the 2020 emails that you wish to include in the ‘Email 2020.veo.zip’ VEO will be placed in the ‘Email 2020’ subdirectory).

You can structure the content within the subdirectories. For example, the ‘Email 2020’ subdirectory could contain subdirectories representing your email folders, with the emails contained in each email folder represented in the appropriate subdirectory. You can use as many subdirectory levels as you wish. The subdirectory structure will be directly represented within the VEO (technically, each subdirectory will become an Information Object within the VEO).

Note that if the information content is already structured in the way you wish the VEOs to be divided and structured, it is not necessary to carry out Steps 1 to 3 – you can just use the existing folder structure.

### Step 4 – Create metadata files in each VEO subdirectory

It is necessary to include metadata in each VEO. At the very minimum, a standard set of AGLS or ANZS5478 metadata must be present at the root of each VEO, however, multiple sets of metadata may be present in each Information Object in a VEO.

When creating VEOs using the Easy VEO Create tool, this metadata is represented as additional text files that are added to the VEO subdirectory (and can optionally be added to the content subdirectories).

The absolutely minimum you need to do is:

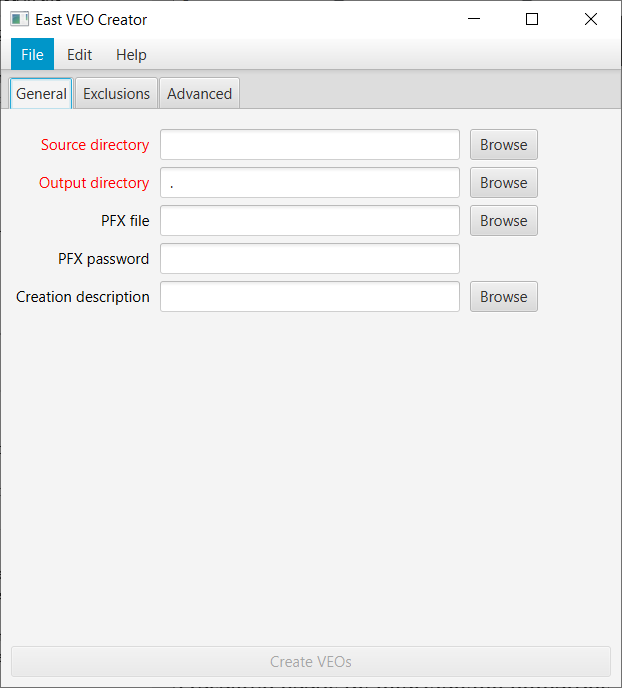
* Decide on how you wish to represent the metadata. You have four choices: to represent it as extremely simple name/value pairs that will be converted into AGLS metadata; as AGLS metadata expressed in RDF/XML; as AS5478 metadata expressed in RDF/XML, or as straight XML. The first is far simpler, the other three allow for more flexibility. The following instructions assume that you wish to use the first approach – simple name/value pairs.
* In each VEO subdirectory create a file named ‘VERS-MD-01.txt’.
* The contents of this text file is a list of metadata elements, one per line. Each line starts with an AGLS metadata element name which is followed by an ‘=’ and then by the metadata value. (e.g. ‘dcterms:title=This is a title’ – note do not include the quotes in the file.)
* Any AGLS metadata element can be used as the metadata element, and metadata elements can be repeated in multiple lines. To make it even simpler, the following metadata abbreviations can be used: identifier for dcterms:identifier; title for dcterms:title; creator for dcterms:creator; date for dcterms:date; created for dcterms:created; protectiveMarking for aglsterms:protectiveMarking; disposalDate versterms:disposal-ReviewDate; disposalAction for versterms:disposal-Action; disposalReference for versterms:disposal-Reference; description for dcterms: description; and dcterms:format for format.
* The metadata value is the characters between the ‘=’ and the end of the line. The special value $$currentDateTime$$ is replaced by the current date and time in the VERS format. Dates are expressed in the standard Web date format (e.g. ‘2021-04-15T13:00:01+10:30’ is one second past one p.m. on the 15 April 2021, 10 hours and 30 minutes in advance of UTC). The timezone (‘+10:30’) is optional, as is the time (e.g. ‘2021-04-15’ is valid and simply indicates a date without specifying a time or timezone.
* One metadata file must appear immediately underneath each VEO entry. Multiple metadata files can be included; the second has the name ‘VERS-MD-02.txt’ (and so on). Metadata files can also appear in any other content subdirectory.

### Step 5 – Run the Easy VEO Create tool over the directory of VEOs

The instructions for running the East VEO Create tool are given in the next section.

Before running the tool you will need to obtain a PFX file containing the private key you will use to sign the VEOs.

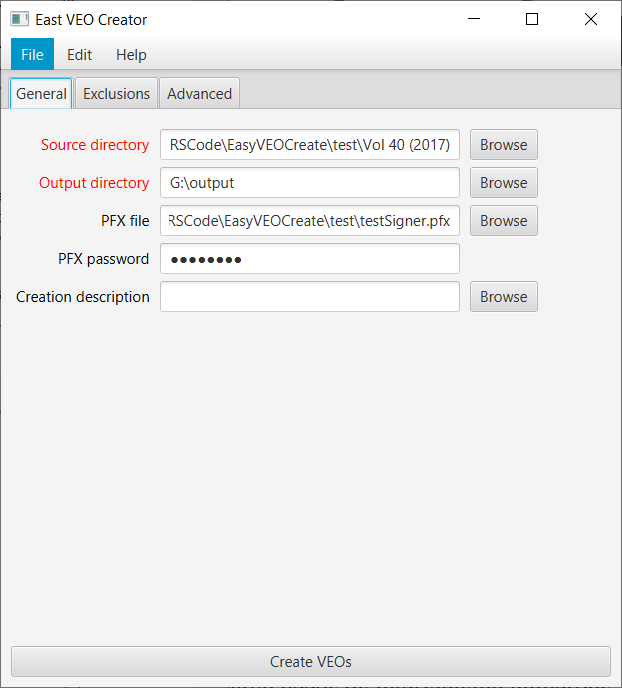
## Creating VEOs (basic)

The initial screen of the Easy VEO Create GUI sets up the VEOs to be created.

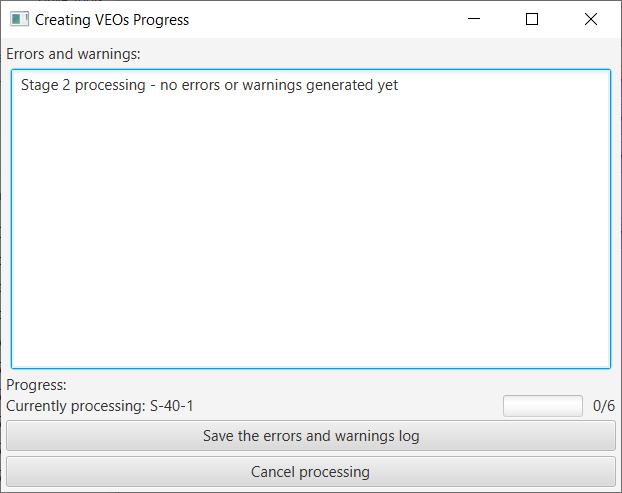
There are five fields that can be entered. The mandatory fields (Source directory and Output directory) must contain data, and have red field names.

The fields are:

* Source directory (Mandatory). The directory containing the VEO subdirectories to be created. The directory can be either directly typed into the text box, or the ‘Browse’ button can be used to popup a file selection window.
* Output directory (Mandatory). The directory in which the VEOs are to be created. The directory can be either directly typed into the text box, or the ‘Browse’ button can be used to popup a file selection window.
* PFX file (Optional). The PFX file that will be used to sign the VEOs. The VEOs will be created, but not signed, if no file is specified.
* PFX password (Conditional). The password that locks the PFX file. This field must be present if the PFX file is present. The file can be either directly typed into the text box, or the ‘Browse’ button can be used to popup a file selection window.
* Creation description (Optional). A text file that contains a description about the creation of this VEO. The file is intended a place where the records manager or archivist can describe the process of creating these records. This might include what the records are, how they were selected, how the content is arranged, what was discarded. If a file is specified, it is included in every VEO created.

When all of the mandatory fields have been entered, the ‘Create VEOs’ button at the bottom of the window will become active. Note that the text ‘Create VEOs’ on the button has changed from grey to black. This indicates that the button can be pressed and creation of the VEOs will commence.

To avoid having to re-enter the contents of the fields for multiple runs of VEO creation, it is possible to create and save ‘Jobs’. After entering some or all of the text fields, select the ‘File’ menu and ‘Save Job’ menu item. This allows the current state of the window to be saved as a JSON file. Subsequently selecting the ‘File’ menu and ‘Load Job’ allows the JSON file to be read and the values restored.

Pressing the ‘Create VEOs’ button starts the creation of the VEOs. This pops up the ‘Creating VEOs progress’ window:

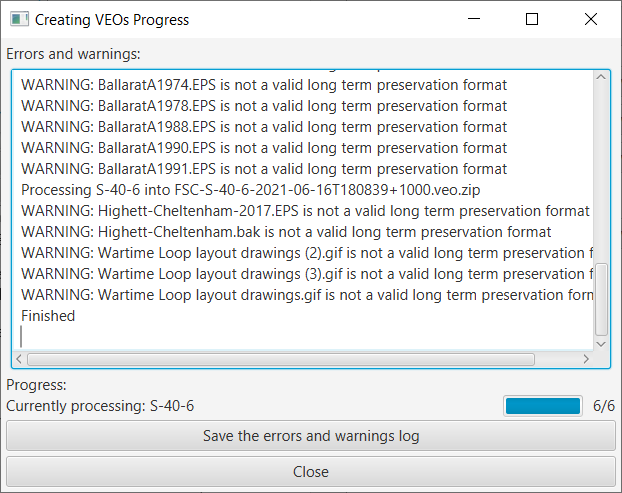
The text area in the centre of the window contains any status or error messages generated when creating the manifest. At the start of the processing a message is displayed that no errors or warnings have yet been generated.

Under the window is a status bar. At the left the VEO directory currently being processed is displayed. At the right is a progress bar. This shows the total number of VEOs to be created, and the number that have been completed.

Clicking the ‘Save the errors and warnings log’ button allows the status and error messages generated to be saved as a log file. The log file also includes a summary of the configuration options.

Clicking the ‘Cancel processing’ button stops the generation of the VEOs and returns you to the initial window. It is possible to recommence creating the VEOs if desired.

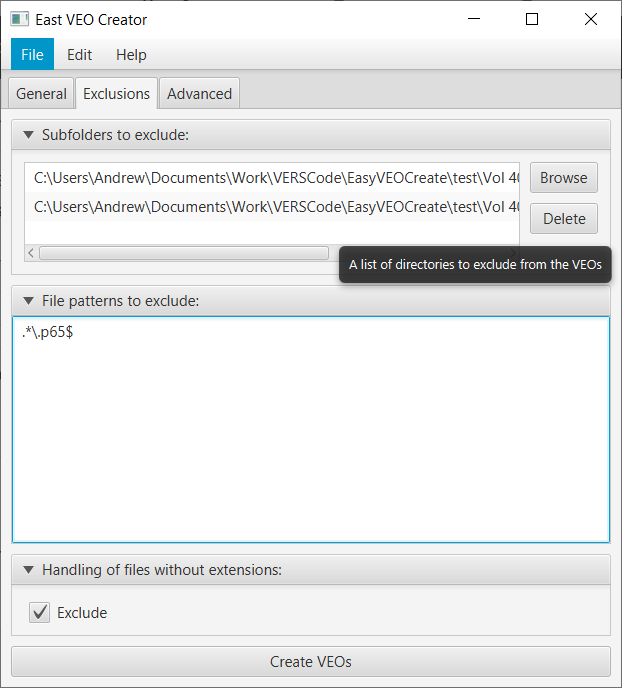
When creation of the VEOs has been competed the progress screen will display:

The text ‘Finished’ has been added to the status panel; the count shows that all VEOs (‘6/6’) have been processed, and the progress bar is at 100%.

The VEOs have been created and the window can be closed by pressing ‘X’ or ‘Close’.

## Excluding files or directories from the VEO

It is possible to specify that certain files or directories present in the VEO subdirectories are NOT to be included in the VEO. This might be useful if you are creating VEOs from an existing set of directories and you do not wish to manually clean out the directories.

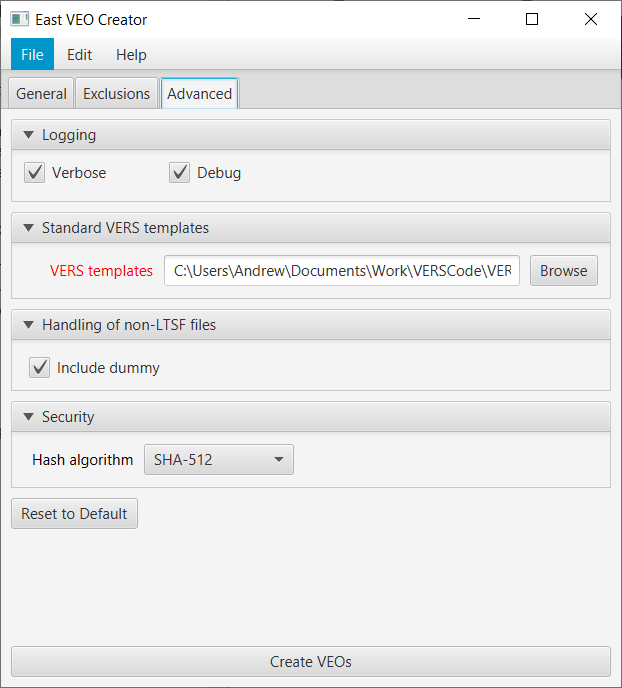
This functionality can be accessed from the ‘Exclusions’ tab of the initial window:

The top portion of the Exclusions pane allows specific directories (and all their contents) to be excluded from the VEO creation. One directory is listed per line. The ‘Browse’ button allows directories to be added. If directories are selected by clicking on that line, the ‘Delete’ button removes them.

The middle portion allows files (or directories) to be excluded from the VEO based on a pattern in the file name. In the example, we have instructed Easy VEO Create to exclude any filename ending with the ‘.p65’ extension (i.e. a Pagemaker file that is not a long term sustainable format). Patterns are listed one per line, and the area can be directly edited to add, modify, or delete patterns. Patterns are specified using the Java regular expression syntax, which is documented in https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html. A slightly simpler description can be found in <https://www.vogella.com/tutorials/JavaRegularExpressions/article.html>.

The final portion prevents file without file extensions from being included in a VEO. A file extension is the final portion of a file name after the ‘.’ (e.g. in the file name ‘Document.docx’, the file extension is ‘docx’). In the Windows and Linux operating system, file extensions are used to indicate the file type (e.g. a Word document). The Digital Archive uses the file extension to test whether files in the VEO are one of the long term sustainable formats, and the VEO validation will fail if the file does not have a file extension. Checking this box prevents such failures by preventing the inclusion of these files in the VEO.

## Advanced options

The final tab (Advanced) in the initial window allows the setting of a number of advanced options. It is unlikely that most users will need to change these options.

The top options control the level of logging: verbose indicates that contextual information should be included in the log as well as errors or warnings; and debug indicates that even more detailed information is to be included.

The middle options control where the standard VERS configuration information is to be found (in particular, the list of valid long term sustainable formats). This is normally set to a standard place in the system.

The third option controls handling of content files that are not valid long term sustainable format files. If the ‘Include dummy’ option is ticked, Easy VEO Create will include a dummy file in the VEO alongside the content file that is not sustainable. This will prevent the Digital Archive from rejecting the VEO because it contains content that is not a long term sustainable format.

The final option allows selection of the hash algorithm used in generating the VEO.

# Creating VEOs without the GUI

The functionality of Easy VEO Create can be used without the GUI, either from the command line or via an API.

## Command line arguments

The following command line arguments must be supplied:

* **-j <jobfile> a JSON file in which the parameters for the creation of the VEOs will be found. The format of this JSON file is the same as that used to load and store jobs; the easiest way to create the job file is to edit a saved job file created by the GUI.**
* <directory> the source directory that contains the VEO subdirectories.

A minimal example of usage is

easyVEOCreate -j jobfile.json sourceDir

## APIs

An API can be used to create VEOs.

The Javadoc may be consulted for more details about the API

End of procedure