# Instructions for running VPA, CreateBulkV2, and CreateBulkV3

## CreateBulkV2

CreateBulkV2 created a large number of semi-random V2 VEOs to assist in testing the VPA module. Note that VEO names ending in ‘-F’ are File VEOs; all other VEOs are Record VEOs.

For each run of CreateBulkV2, the following features may be changed:

* The number of VEOs to be created
* The number of Record VEOs within each File VEO is randomly set.
* The maximum number of Documents that a VEO can contain (the actual number of Documents in each VEO will vary up to this limit)
* The File Metadata, Record Metadata, and Document Metadata. All VEOs created in a run have the same basic metadata, but some provision is made to vary the metadata systematically in a run. This ensures that Documents vary systematically.
* The Encodings within each Document. Note all Encodings within all Documents created in one run are identical (however the file names associated with the encodings vary systematically)
* The number and identify of the signer
* The hash algorithm used.

The following command line arguments are available:

* -r <number> The number of VEOs to create
* -m <number> The maximum size of the VEOs (in Documents). The actual size of each VEO will vary randomly between 1 document and this number.
* -t <directory> The directory containing the templates used to create the VEOs. This directory contains two subdirectories, and one file:
* content (directory). The files within this directory are the different encodings of the document(s) within the VEOs.
* templates-99-007 (directory). This subdirectory contains the directory ‘encodingTemplates’ (which should not be changed), and three files: document.txt, file.txt, and record.txt. The three text files are the templates used to create the Document, File, and Record metadata in each VEO. These text files can be altered at will to change the VEOs produced in a run.
* signer.pfx The PFX file used to sign the VEOs.
* -s <PFXfile> <password>. This is the PFX file used to sign the VEOs (and its password)
* -o <directory>. A directory in which to put the created VEOs
* -ha <hash alg>. The hash algorithm to be used in signing each VEO. By default ‘SHA-512’.
* -v Produce verbose output.

For File Metadata, the following information is available to be substituted:

* Column 1: a string representing the VA (agency) number. Currently set to ‘123’.
* Column 2: a string representing the VPRS (series) number. Currently set to ‘421’.
* Column 3: a string representing the file identifier of the file (automatically generated and will vary among the VEOs)
* Column 4: a string representing the record title. This is identical for each VEO.
* Date: a string representing the current data and time.

For Record Metadata, the following information is available to be substituted:

* Column 1: a string representing the VA (agency) number. Currently set to ‘123’.
* Column 2: a string representing the VPRS (series) number. Currently set to ‘421’.
* Column 3: a string representing the file identifier of the containing file (automatically generated and will vary among the VEOs)
* Column 4: a string representing the name of this VEO (used as a record identifier) (automatically generated and will change in each VEO)
* Column 5: a string representing the record title. This is identical for each VEO.
* Date: a string representing the current data and time.

For Document Metadata, the following information is available to be substituted:

* Column 1: a string representing the VA (agency) number. Currently set to ‘VA123’.
* Column 2: a string representing the document title (automatically generated and will change in each Document – it is currently the VEO name and a count)
* Date: a string representing the current data and time.

## CreateBulkV3

CreateBulkV3 created a large number of semi-random V3 VEOs to assist in testing the VPA module.

For each run of CreateBulkV3, the following features may be changed:

* The number of VEOs to be created
* The maximum number of Information Objects that a VEO can contain (the actual number of Information Objects in each VEO will vary up to this limit)
* The metadata package. All VEOs created in a run have the same basic metadata, but some provision is made to vary the metadata systematically in a run. This ensures that Documents vary systematically.
* The Encodings within each Document. Note all Encodings within all Documents created in one run are identical (however the file names associated with the encodings vary systematically)
* The number and identify of the signer
* The hash algorithm used.

The following command line arguments are available:

* -r <number> The number of VEOs to create
* -m <number> The maximum size of the VEOs (in Information Objects). The actual size of each VEO will vary randomly between 1 Information Object and this number.
* -p <probability> The probability that the depth of the next Information Object will be increased or decreased by one. If specified the probability must be greater than 0 and less than or equal to 0.4
* -t <directory> The directory containing the templates used to create the VEOs. This directory contains two subdirectories, and one file:
* content (directory). The files within this directory are the different content files of the Information Object(s) within the VEOs.
* templates-15-03 (directory). This subdirectory contains the directory ‘neoVEOSchemas’ (which should not be changed), and two files: agls.txt and VEOReadme.txt (which should not be changed). These agls.txt file can be altered at will to change the VEOs produced in a run.
* signer.pfx The PFX file used to sign the VEOs.
* -s <PFXfile> <password>. This is the PFX file used to sign the VEOs (and its password)
* -o <directory>. A directory in which to put the created VEOs
* -ha <hash alg>. The hash algorithm to be used in signing each VEO. By default ‘SHA-512’.
* -v Produce verbose output.

For metadata, the following information is available to be substituted:

* Column 3: a string representing the RDF identifier of the Information Object (automatically generated and will vary among the VEOs)
* Column 4: a string representing the Information Object title. This is the VEO Name.
* Column 5: a string representing the Information Object identifier. This is the VEO Name.
* Date: a string representing the current data and time.

## VPA

This program invokes the VPA module on a set of VEOs. V2 and V3 VEOs can be mixed in the set.

The following command line arguments are available:

* -v Verbose output
* -s <directory> A directory containing the necessary V2 and V3 schemas for validation
* -o <directory> A directory where the packages generated from the VEO are to be generated
* -h If present, use the real handle service. Otherwise fake handles
* A set of directories containing VEOs (or the VEO files themselves). At least one directory must be present.