Manual for Processing and Ingesting Archival Email

MANCHESTER 1824

The University of Manchester

Manual for processing and ingesting archival email

July 2014

Fran Baker, Phil Butler, Ben Green
The University of Manchester Library

Table of Contents

Introduction

This manual provides instructions for curators and technical staff on the transfer, processing, packaging and ingest of email acquired as part of archives.

The instructions are based on work done as part of the Carcanet Press Email Preservation Project,[1] and all examples are drawn from that project. The guidance is also based specifically on email which has been created using a Microsoft Outlook client and captured in the form of Microsoft PST files. However, many of these workflows could potentially be adapted to deal with email in other formats.

The guidance given here will be updated periodically to reflect changes in hardware, software and any updates to existing workflows.

Currently many of the processes outlined are also manual in nature and involve using the command line and raw XML to execute; it is hoped that in future a greater degree of automation can be developed.

Chapter 1 of this manual focuses on processing and preserving at 'email sequence' level (based on Microsoft Outlook PST files).

Chapter 2 focuses on preservation at individual email level.

The guidance outlines the workflows involved in creating five different types of digital object for ingest into Manchester eScholar, illustrated in the following four workflow diagrams (Figures 1-4). These are:

- Collection object: the highest-level object. In PREMIS terminology[2], this is an 'Intellectual Entity', or conceptual object: a digital or hybrid archive which contains one or more accessions, multiple email sequence objects, each of which might contain many individual email objects.

- Accession object: also an 'Intellectual Entity', representing a specific accrual of archive material with the same provenance, which might contain one or more Email Sequence objects. The Carcanet Press Archive has always been arranged by accession, but this type of digital object may not be used in all digital archives, and future guidance will take this into account.

- Email Sequence object: for ease, the content of each PST file is defined as an 'email sequence', representing an archived PST or snapshot of a mailbox taken at a particular point in time. Each email sequence object is an 'Intellectual Entity' (a sequence of email correspondence); a 'Representation' (a particular way of rendering that sequence of email correspondence); and a 'File' (i.e. an actual PST file).

- Email folder and email message object: an individual email, with all its attachments where relevant: the email is an 'Intellectual Entity', containing several different 'Representations' (the email is preserved in several different formats, each providing a different way of rendering the message), and consists of several 'Files'.

[1] Reports are available for Phase 1 and Phases 2-3 of this project.

[2] Preservation Metadata: Implementation Strategies, http://www.loc.gov/standards/premis/.

Figure 1: Email Sequence

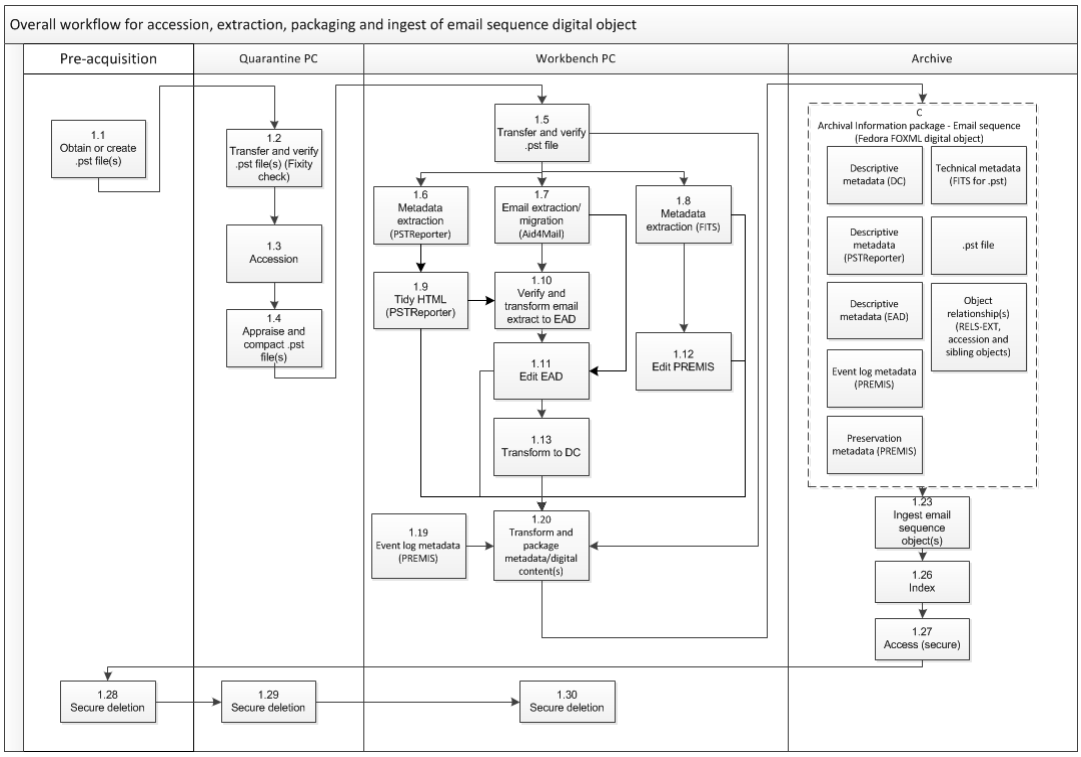


Figure 2: Accession

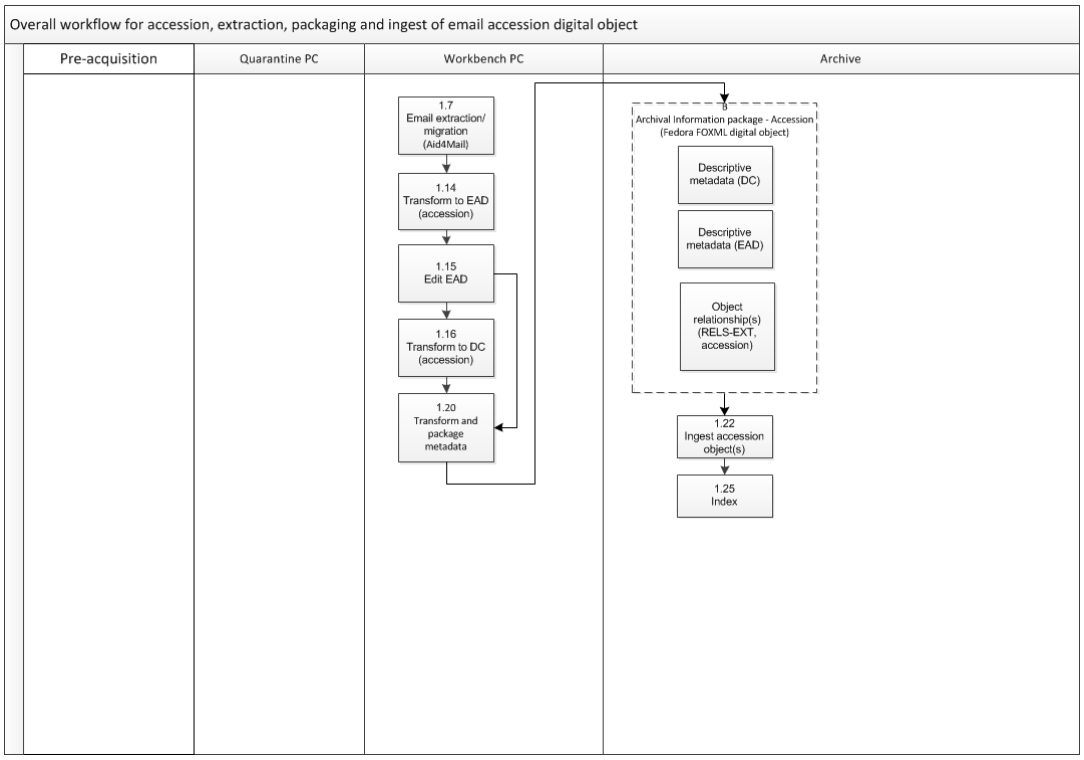


Figure 3: Collection

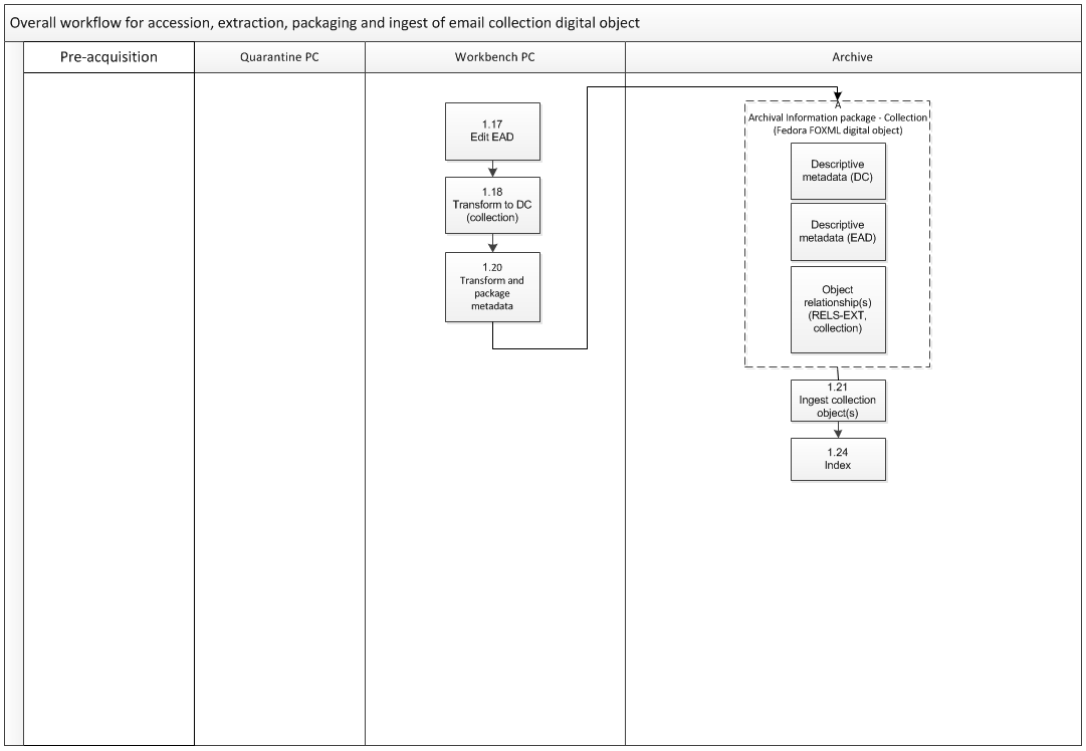
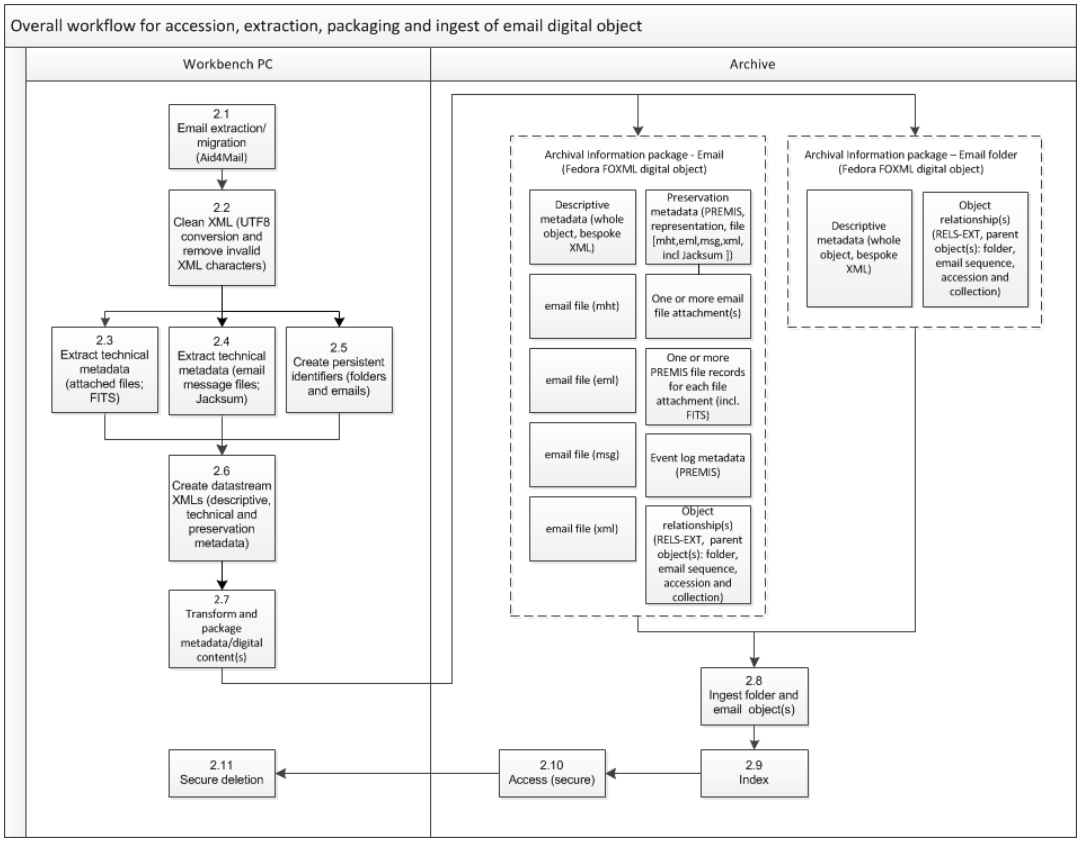


Figure 4: Email folder and email message



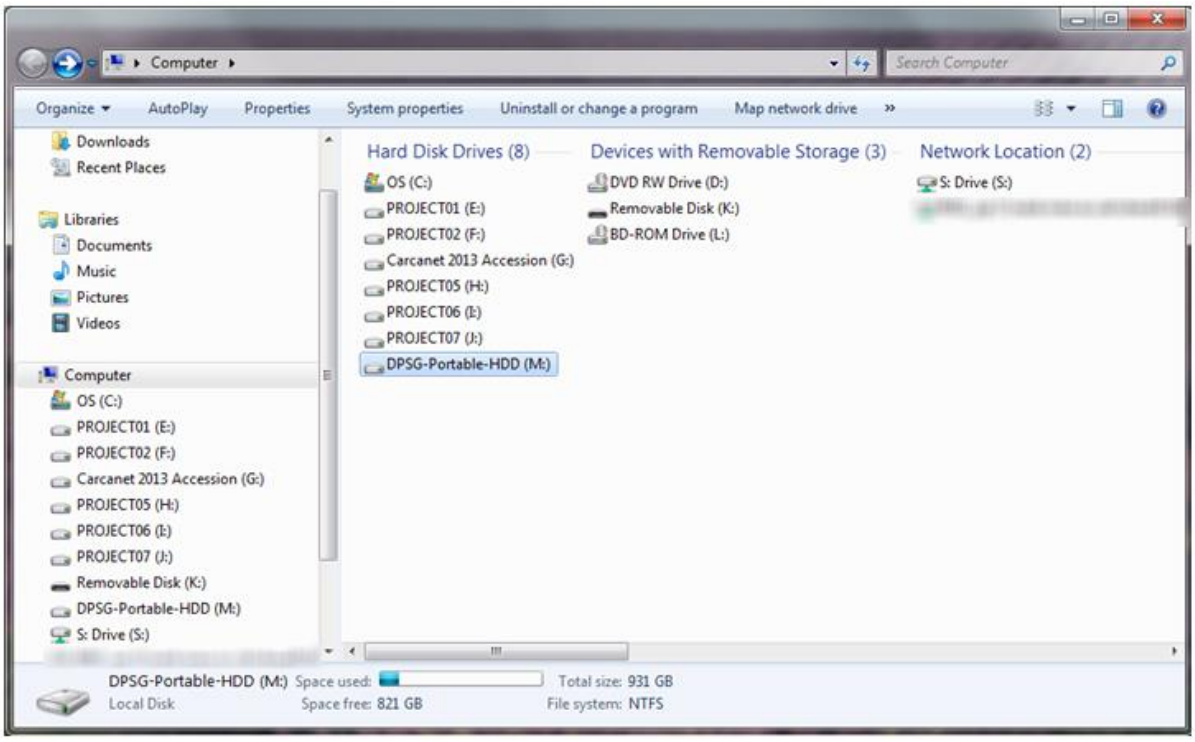
Chapter 1: Archiving emails at email sequence level

Step 1.1: Locate and/or create PST files

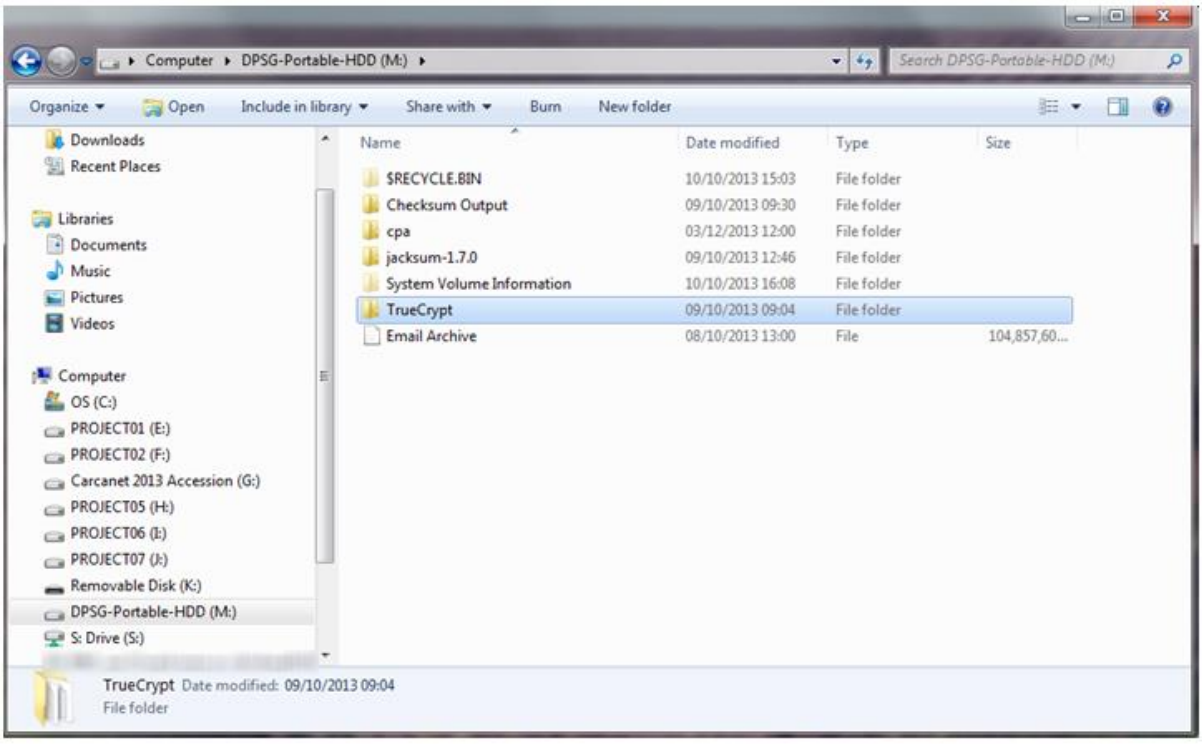
1.1.1 Configuring the removable hard drive

PST files are transferred from the creator's PC to the Library by means of a removable hard drive. This hard drive needs to encrypted to ensure safe and secure transport of files. One way to do this is to use TrueCrypt's Portable Mode software (see http://www.truecrypt.org/docs/truecrypt-portable). It is advisable to encrypt only a portion of the hard drive to allow other software to be installed on the non-encrypted sections. When TrueCrypt encrypts part of a hard drive you are asked for a file name and password. You should call the encrypted file 'Email Archive'. The password used MUST be retained securely. This encrypted file acts as a lockable container for all the archival files and metadata you wish to transfer. Once the removable hard drive is encrypted you will need to install the TrueCrypt executable (which 'unlocks' the encrypted file so you can save material to it) on the unencrypted part of the drive, along with Jacksum software, which runs checksums on the archival files.

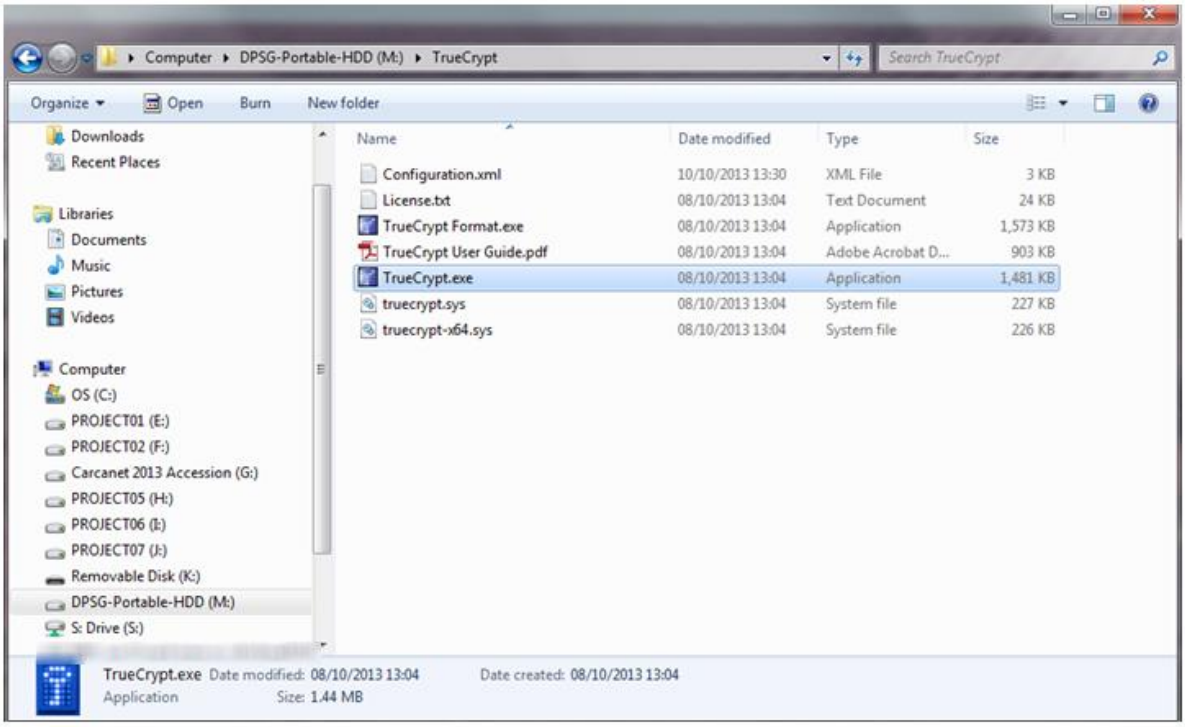
Once onsite, connect the removable hard drive to the creator's PC; when connected, navigate to the hard drive, which will appear in the list of local and network drives on the creator's PC, as in the example below:



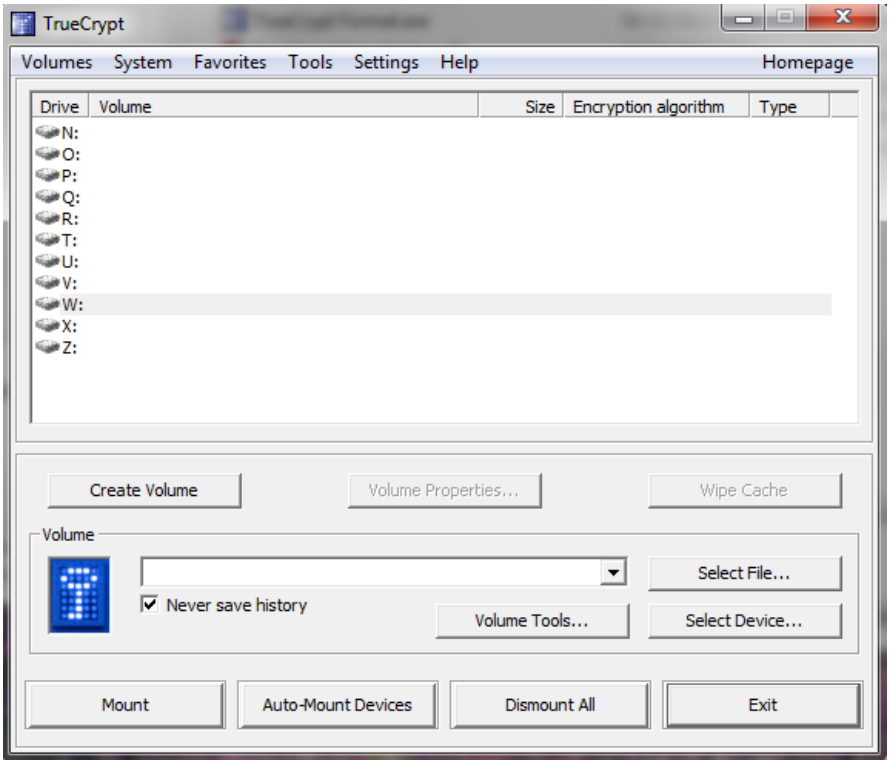
From here, find the TrueCrypt executable, which is located in <drive letter>:\TrueCrypt:



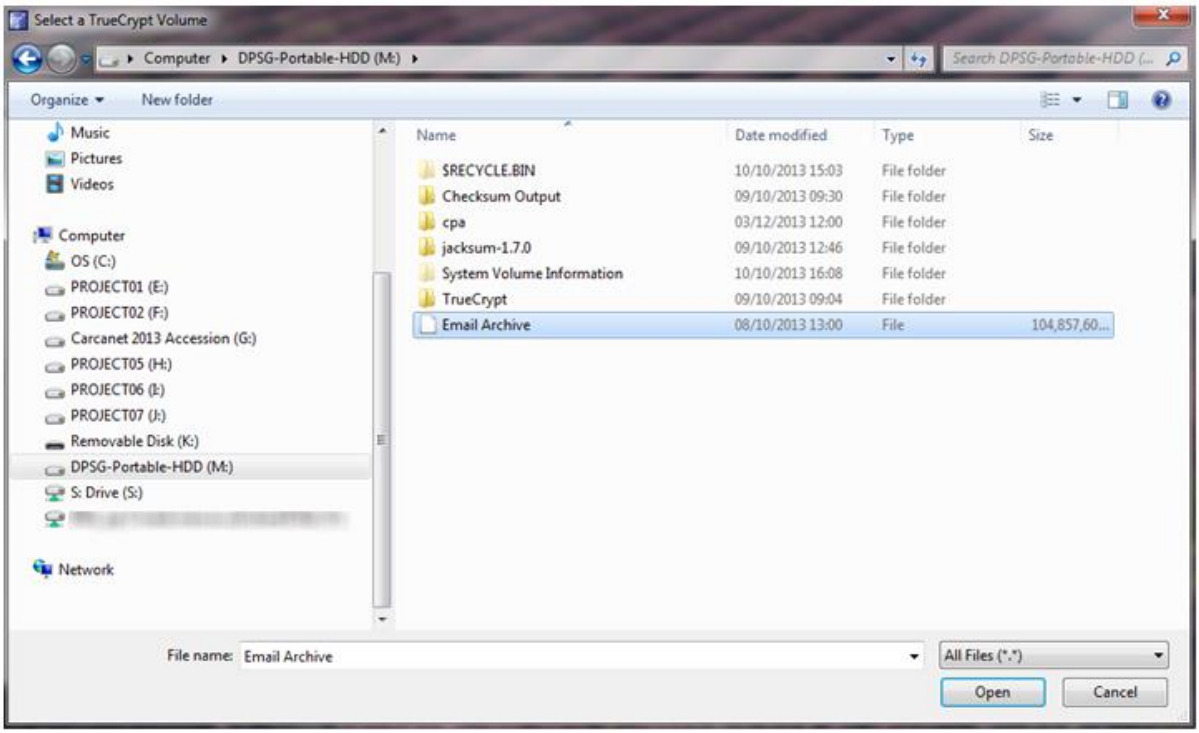
Run ‘TrueCrypt.exe’ by double clicking it:



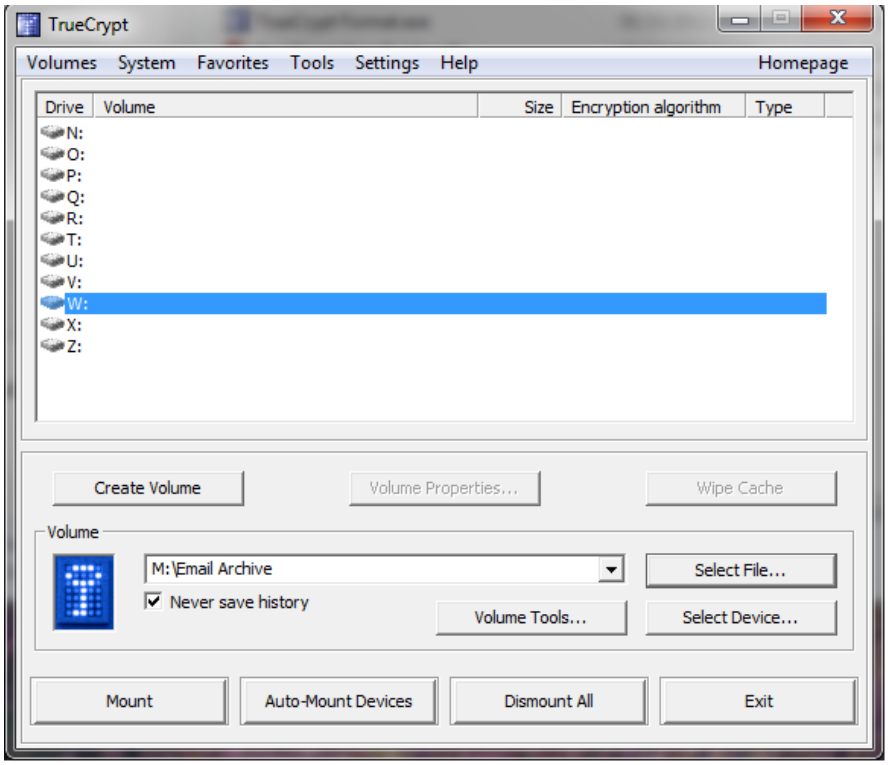
Choose ‘Select file’:



Browse to the remote drive and find the file 'Email Archive'. Select this file and click <Open>.

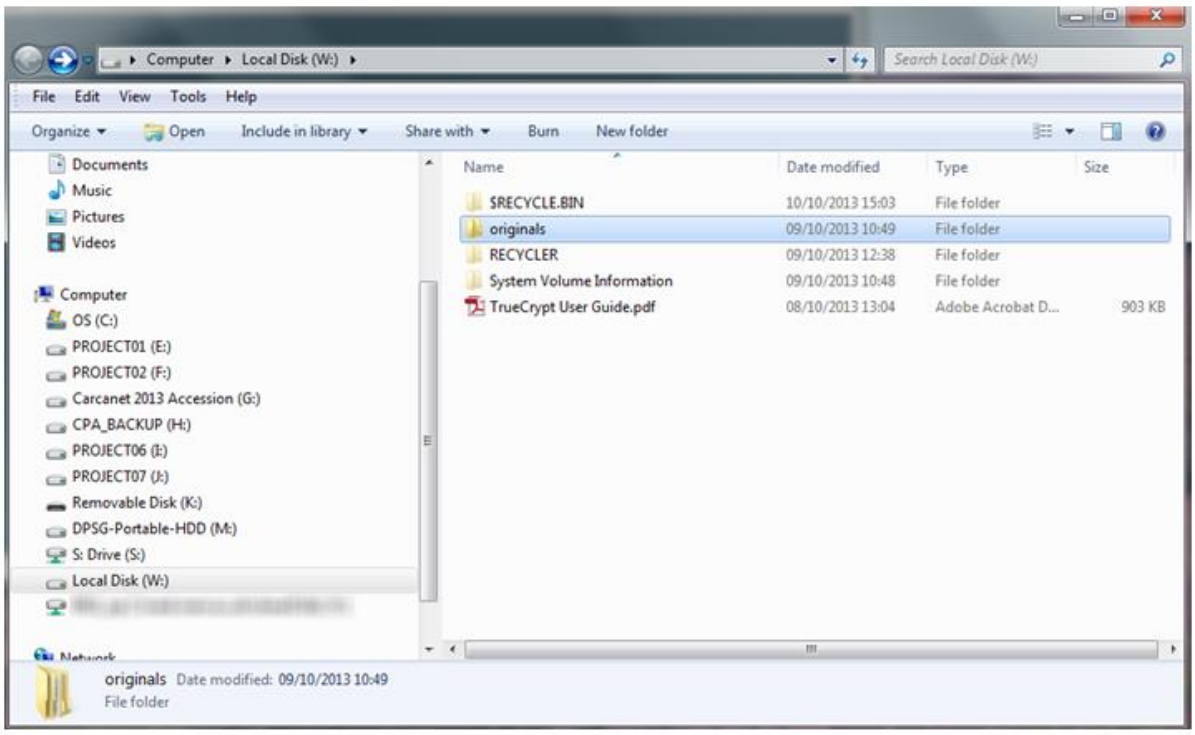


Choose any available drive letter from the list presented and click <Mount>.



When you are prompted for a password, enter the password that was used when the hard drive was first encrypted then click <Ok>.

Create a folder on the removable hard drive called ‘originals’ on the drive letter that you ‘mounted’ in the previous section. This will hold all the PST files to be transferred.



If you know how many accessions you are dealing with at this stage, create one or more subfolders within 'originals', one per accession; in the absence of an accession number, these should be named with the temporary reference 'a01', 'a02' etc. Within the accession folders, there will be one or more subfolders for each PST created or transferred, named 001, 002 etc. See Step 1.1.5 for an overview of naming conventions.

1.1.2 How to locate and/or create PST files

If 'archived' PST files already exist on the creator's PC (i.e. they have already manually or auto-archived some of their email), fixity checks need to be run on these prior to their transfer to the removable hard drive. If there are no existing PSTs, you must create one or more using the procedures outlined below; PSTs created by Library staff can be saved straight to the removable hard drive and the first fixity check taken on the removable drive (see Section 1.1.4 for instructions).

Do existing 'archived' PST files exist on the creator's PC?

1.1.2.1 If ***yes***, locate these by **one** of the following methods:

a. Search the local hard disk for *.pst (e.g. in MS Windows, load Windows Explorer (WIN+E), search for file with extension .pst).
b. Open up MS Outlook (instructions are version specific) and look for the Auto Archive Settings option, then select Advanced and identify the folder/filename used for the auto archive.
c. Open up MS Outlook (instructions are version specific) and look for any additional 'personal folders'; select a personal folder and choose 'open file location'; extract the folder/filename from this location.

Move to Step 1.1.3 below.

1.1.2.2 If ***no***, create a PST as follows:

a. Choose File menu item.
b. Choose Import/Export (in Outlook 2010 choose Open then Import).
c. Choose 'Export to file' from the list of import/export options in the 'Import and Export Wizard'.
d. Choose Personal Folder File (.pst).
e. There is no straightforward way of selecting individual components within a Mailbox to export (e.g. omitting calendar, contacts etc), so at this stage, the whole Mailbox will be exported. Choose 'include subfolders' at this point if necessary.
f. Apply any filtering using the Filter option e.g. date range filters. To filter by date, click the 'Filter' button; go into the 'Advanced' tab, and using the 'Field' button select a combination of filters as appropriate. This will usually be used only for filtering by date: from the list of options, choose 'Date/time fields' option; and check 'Received' (this will also filter the 'sent' items). If taking in yearly accruals, filter by using the 'on or after' option – based on the date of the last transfer. The date should be given in the format: dd/mm/yyyy.
g. Create a folder on the removable hard drive for each PST, named 001, 002, …, 00n. If you are able to name the PST itself, name it 001.pst, 002.pst, …, …00n.pst. Otherwise rename it after saving it (see Section 1.5 for naming conventions).
h. Select 'allow duplicate items to be created'.
i. Choose any encryption and/or password necessary to secure the PST.
j. Save the PST to the corresponding folder (001, 002 etc) on the removable hard drive.

1.1.3 How to use Jacksum to run checksums on PSTs

Where PSTs have been created by Library staff, the first fixity check on these will be run on the removable hard drive. However, where PSTs already exist on the creator's PC, a fixity check should be run prior to copying the PST to the removable hard drive, and again after copying.

Jacksum software (for running checksums) is installed on the portable hard drive in the folder called 'jacksum-1.7.0'. There is no need to install this program on the PC that is being analysed; Jacksum can be executed from the portable hard disk. Save the Jacksum output text file with the name of the PST file it has been run against, and add the suffix _f1.txt. It should be temporarily saved to the same location as the PST it relates to on the creator's PC.

A sample Jacksum command is as follows:

e:\java –jar jacksum.jar –a sha1+crc32 –s "\t" –t –m –f –o <PST name>_f1.txt archive.pst

In the above example:

- The portable hard drive has been mounted as partition 'e:' on a PC with a primary hard drive 'c:'
- Java is installed and is running from the root of the removable hard drive. This may not always be the case: the filepath will depend on where it is installed on the hard drive (e.g. it could be within a dedicated 'java' folder); and as most PCs are likely to have Java installed, it may be possible to run Java using the creator's PC (the filepath can be found automatically by starting to type 'java' in the command).

- The PST file being analysed by Jacksum is 'archive.pst'. If written as above, it assumes the filepath of archive.pst to be at the root of the c: partition. In reality, it is likely to reside in a folder structure that must be specified on the above command line.
- Jacksum is using a combination of the SHA1 and CRC32 checksums and outputting the results in a text file. The full path of the resulting output file must be specified. For example, if you wished the output to be placed directly on the portable hard drive, the destination file name would be 'e:\checksum-output.txt'.

This outputs the hash data in a tab delimited format, in order to make it easier to copy and paste into an Excel spreadsheet.

The resulting output would be:

```
sha1+crc32		00cc2833bafeba7922062faf35d12ba903c0776d3f4feb84		38618112
archive.pst
```

In the case of PST files created by Library staff and saved directly to the removable hard drive, the first fixity check should be run on this drive: the Jacksum output file should be named with the PST number and suffix _f1.txt, and saved in the folder with the PST file it relates to. A sample command (assuming the hard drive is mounted as drive e: on the creator's computer) would be:

e:\java -jar jacksum.jar -a sha1+crc32 -s "\t" -t -m -f -o e:\originals\a01\001\001_f1.txt e:\originals\a01\001\001.pst

1.1.4 How to transfer files to the removable hard drive and checksum again

The basic folder structure on the removable hard drive has already been created in the previous steps; PST files created by Library staff, and their associated Jacksum outputs, have also already been saved to the relevant numbered folder (001, 002 etc).

PST files which already existed on the creator's PC, and their associated Jacksum outputs, should now also be transferred to the removable hard drive.

Create suitably named folders for the files to be copied to: each folder (named 001, 002, 003 etc) should contain a single PST file and its Jacksum output in a text file.

Simply copy and paste each PST and Jacksum output file from their location on the creator's PC into the relevant folder on the removable hard drive.

These files themselves can now be renamed appropriately, incorporating the folder name, e.g. as 002.pst, 002_f1. However, it is important that the original name of the PST is recorded somewhere; if this was a name assigned by the creator it might have some significance. Check that the Jacksum output includes the original name and filepath. If not, ensure that you record this somewhere.

A second fixity check must be run on all PSTs which were not created by Library staff directly onto the removable hard drive. This is to ensure no changes have taken place in the process of copying and pasting the files from the creator's PC. See Section 1.1.3 for instructions on running Jacksum to create fixity checks. Ensure that all filepaths are correct.

Save the Jacksum output for this second fixity check in the same folder as the PST it relates to and the first fixity output. Save it as <sequence number>_f2.txt, e.g. 002_f2.txt.

1.1.5 Naming conventions

When transferring PST files to the removable hard drive you may encounter situations where multiple PST files exist with the same name. It is also important to be able to track back from the copied file to the original source file. The folder structure and naming system has been covered in the previous steps, but can be summarised as follows:

1. A folder called 'originals' should be created at the root of the removable hard drive to store all the archival files which are being transferred.

2. Within this folder, a folder for each accession should be created. If you don't know the accession number at this stage (which is likely to be the case) then create folders named a01, a02, …, a0n, one per accession if dealing with multiple accessions.

3. It may be necessary to copy multiple files in parallel, especially when the PST files are large. To avoid confusion, one subfolder for each PST should be created in the relevant accession folder. Name these subfolders 001, 002, 003, …, 00n.

4. In the case of PST files created by Library staff: each PST file created should be saved in the relevant subfolder and named with that subfolder's title, i.e. 001.pst, 002.pst, …, 00n.pst. The Jacksum output relating to each PST should be stored alongside it in each subfolder, and named accordingly: 001_f1.txt, 002_f1.txt, …, 00n_f1.txt.

5. In the case of PST files which already exist on the creator's PC: these will be copied into the relevant subfolder on the removable hard drive, along with their associated Jacksum output. They should then be renamed using the same system outlined in point 4 – *ensuring that the creator's original name is recorded somewhere*.

6. In the case of PST files which have been transferred as in point 5, a second fixity check must be run on them after transfer to the removable drive. Name these 001_f2.txt, 002_f2.txt …, 00n_f2.txt etc.

1.1.6 Dismount removable hard drive

When you have all the archival and metadata files saved on the hard drive, you will need to 'dismount' the TrueCrypt application. Find the TrueCrypt programme and run it as outlined in Section 1.1.1.

Select the drive letter of the encrypted space and click <Unmount> (see screenshot on p. 12). Safely disconnect the hard drive from the creator's PC.

Finally, delete the text file containing the first Jacksum output from the creator's PC. Do NOT delete the PST file itself unless this has been agreed with the creator.

1.1.7 Complete transfer list

Ensure that you record details of the material transferred on a transfer list (a pro-forma can be found in the Appendix to this manual), and that this is signed both by the donor/depositor or their representative, and the relevant member of curatorial staff. This includes some information about hardware and technical environment which can be transferred to the EAD record at a later stage.

The transfer list should be scanned and stored in EMU collection management software as a multi-media record, linked to the accession record to which it relates.

1.1.8 Step 1.1 folder and file structure

At the end of this stage the following folder and file structure should exist on the removable hard drive:

For PST files created by Library staff:

\originals\<temporary accession reference>\<email sequence number>\<email sequence number>.pst
\originals\<temporary accession reference>\<email sequence number>\<email sequence number>_f1.txt

For PSTs which already existed on the creator's PC:

\originals\<temporary accession reference>\<email sequence number>\<email sequence number>.pst
\originals\<temporary accession reference>\<email sequence number>\<email sequence number>_f1.txt
\originals\<temporary accession reference>\<email sequence number>\<email sequence number>_f2.txt

e.g.

\originals\a01\001\001.pst
\originals\a01\001\001_f1.txt
\originals\a01\001\001_f2.txt
\originals\a01\002\002.pst
\originals\a01\002\002_f1.txt
\originals\a01\002\002_f2.txt

Step 1.2: Transfer and verify PST files

1.2.1 How to transfer PST files from removable hard drive and run a checksum

This assumes that TrueCrypt and Jacksum are installed on the removable hard drive and that all PST files have previously been copied to an encrypted partition on the hard drive.

1. Choose an empty drive for the collection on the Quarantine PC. If you are dealing with an accession to an existing collection, there may already be a drive with the relevant collection folder in it. If not, you will need to create a new collection folder named with the three-letter code of the collection if you know it at this stage, e.g. cpa for Carcanet Press Archive; use lower case letters.

2. If no collection folder exists, create a new folder at the root of this drive of the Quarantine PC, named with the collection reference code. This will be the three-letter mnemonic assigned to the collection in accordance with in-house practice, but use **lower case** letters. Before assigning a new reference code, check the REFCODES document in the s:\archive folder to avoid duplication, and remember to record any new code in the REFCODES document as well as the EMU record at the point of accessioning (see Step 1.3). Keep folder names short, all lower-case letters, no spaces and use underscore characters to delimit words as necessary.

3. Connect the removable hard drive to the Quarantine PC (see Section 1.1.1 for instructions on mounting the encrypted hard drive on a desktop computer); once connected, navigate to

the TrueCrypt executable on the hard drive (within the TrueCrypt directory), and run TrueCrypt.exe. The TrueCrypt application will remain open in the background. Select a partition letter (e.g. 'z') to mount the encrypted drive. Now click on <Select File>, navigate to the portable hard drive and select the encrypted file called 'Email Archive'. Click <Open>, and then click <Mount> at which point you will be prompted for the password. A partition will now exist at the selected drive letter (e.g. 'z:') on the Quarantine PC.

4. Copy and paste the accession folder or folders from the 'originals' folder (but **not** the 'originals' folder itself), from the encrypted drive, e.g. z: drive (using naming conventions listed below), to the collection folder created in step 1 above. Verify that all folders and files have been copied.

5. On completion, click on <dismount> from the TrueCrypt application, close TrueCrypt and safely disconnect the removable hard drive. Store the removable hard drive somewhere secure (NB: leave files on the encrypted partition for secure deletion at a later stage).

6. Run Jacksum on the copied PST files. See instructions in Section 1.1.3 for running Jacksum. When creating the Jacksum output, name the file 001_f2.txt or 001_f3.txt, 002_f2.txt or 002_f3.txt, …, 00n_f2.txt …, 00n_f3.txt (as appropriate, depending on whether the PST was created by Library staff or archive creator) so that subsequently you can track which Jacksum output applies to which original PST file. Check that no changes have occurred in the checksum reading.

1.2.2 Record actions in Event Log and transfer master templates and code

The fixity checks and certain other actions need to be recorded in an Event Log for each PST file/email sequence. The master Event Log template and other essential templates and code are stored on a network, but the processing of each collection takes place locally on the Workbench PC. At this stage, you need to copy the master documents onto the Workbench PC so you are working from your own local copies. NO archive material should be stored on any networked drive.

1. Select a partition drive from the unused drives on the Workbench PC for your project/collection. Create a folder on it named with the three-letter collection code (use **lower-case** letters).

2. Open the folder <drive letter>:\EmailArchivingMasterCode\ and copy the four subfolders within this folder, i.e. 'BAT-Scripts', 'Aid4Mail-Scripts', 'XML-Templates', and 'XSLT-Transforms'.

3. Paste these subfolders into your collection folder; these will act as the templates for your current project.

4. In order to record the fixity checks you have run up to this point, you will need a copy of the Event Log template for each separate PST file/email sequence. Within your collection folder, create another subfolder called eventlog. Go to the XML-Templates folder and copy the file called PREMIS_Eventlog_EmailSequenceTemplate.xml. You will need a copy of this for each PST file you are processing, so paste as many copies as you need into the eventlog folder. Name each template with the accession number (or temporary reference), sequence number and suffix 'eventlog', e.g. a01_001_eventlog.xml, a01_002_eventlog.xml.

5. We are using a local version of the PREMIS schema for the Event Log; this has been modified to include tailored drop-down menus etc. In order to ensure that each Event Log validates, a copy of the local schema needs to be present in the same folder, so copy and paste this from

the XML-Templates folder into the eventlog folder. It is called premis-localextension-eventlog.xsd.

6. See Step 1.19 for detailed instructions on filling in the Event Log for each PST.

1.2.3 How to run a virus check

Once all the PST files have been safely transferred to the Quarantine PC, a virus check should be run. The only means of doing this *within* a PST file is to use an anti-virus application that embeds itself into Outlook. A dummy Outlook account is installed on the Quarantine PC.

1. Firstly, you MUST ensure that the reading pane in MS Outlook is turned off; if this is turned on it can change the status of unread messages to 'read'. Reading panes can be disabled globally by running Outlook in safe mode with a ':1' option. Do not attempt to launch Outlook by clicking on the desktop shortcut. Safe mode Outlook needs to be launched via the Run command (command line), by going into the Start menu, and entering 'cmd' in the search box to bring up the command prompt. Type in the following: outlook.exe /safe:1

2. Open Outlook, and load the first master PST as a data file. Instructions are version-specific, but in MS Outlook 2010, use: File -> Open Outlook Data File -> browse for file. This will be the first master PST, at \<collection code>\<temporary reference>\<email sequence number>\<email sequence number>.pst

3. To run the virus check, go to the McAfee E-mail Scan menu at the top of the screen, and click on the left-hand icon ('On-Demand E-mail Scan'). This process checks every message body and attachment, and can take some time to run.

4. Currently we are unable to modify this tool with our preferences, and it deletes any files containing identified viruses or other threats. However, where this happens, the tool can generate reports of any deleted files and the reason for deletion. Create a subfolder within the accession folder on the Quarantine PC called 'reports'; this will be used for storing the McAfee, and subsequent Email Examiner, reports. Use this filepath: \<collection code>\<temporary reference>\<reports>\

5. Save the McAfee report as a text file in this folder, using the email sequence number and suffix _virus_report as its name, i.e. \<collection code>\<temporary reference>\<reports>\<email sequence number>_virus_report.txt. This file will not be ingested, but – along with the appraisal reports – should be stored in the EMU collection file as part of the audit trail (see below).

Record ALL events in the appropriate event_log.xml file on the Workbench which was created at Step 1.2.1 above (see Step 1.19 for detailed instructions).

Save a copy of any virus report as a multi-media record in EMU, linked to the accession record to which it relates.

1.2.4 Step 1.2 folder and file structure

At the end of this stage the following folders and files will exist on the Quarantine and Workbench PCs.

On Quarantine PC :
\<collection code>\<temporary reference>\<reports>\<email sequence number>_virus_report.txt

\<collection code>\<temporary reference>\<email sequence number>\<email sequence number>.pst
\<collection code>\<temporary reference>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<temporary reference>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<temporary reference>\<email sequence number>\<email sequence number>_f3.txt

e.g. \cpa\a01\reports\001_virus_report.txt
\cpa\a01\reports\002_virus_report.txt

\cpa\a01\001\001.pst
\cpa\a01\001\001_f1.txt
\cpa\a01\001\001_f2.txt
\cpa\a01\001\001_f3.txt
\cpa\a01\002\002.pst
\cpa\a01\002\002_f1.txt
\cpa\a01\002\002_f2.txt
\cpa\a01\002\002_f3.txt

On Workbench PC:

\<collection code>\eventlog\premis-localextension-eventlog.xsd
\<collection code>\eventlog\<temporary reference>_<email sequence number>_eventlog.xml
(NB These Event Log files will need to be copied into each email sequence folder after the files have been transferred from Quarantine to Workbench; ensure that a copy of the local PREMIS template is also copied).

e.g. \cpa\eventlog\a01_001_eventlog.xml
\cpa\eventlog\a01_002_eventlog.xml
\cpa\eventlog\premis-localextension-eventlog.xsd

Step 1.3: Accession

1.3.1 Create accession record

Use EMU to record digital accessions in the same way as other archive accessions, as follows:

1. Open the Accession Lot module in EMU and select New Record.

2. In the Summary tab, designate the Accession (in Collection type field) as ‘Archive’. Other fields in this tab are automatically filled in when the corresponding fields in the Details tab are completed and saved, so ignore these.

3. Open the Details tab.

4. Ensure that the accession has a unique accession number; this is supplied automatically by the system. The format is the same with the transition to EMU, but the running number following the slash is no longer prefaced with zero (i.e. 2013/32 rather than 2013/032).

5. Record the date of the accession in the form dd/mm/yyyy; only one date can be included so if the same accession was acquired on more than one date (e.g. if the copying of a large PST file is left running overnight), record the last date.

6. Record the owner of the material using the look-up list for this field.

7. In the Method of Acquisition field, record the type of acquisition (donation, deposit etc) using the look-up list.

8. Record the title of the accession in the relevant field, e.g. 'Digital Archive of xx'; 'Accession 3 of the Digital Archive of xx'.

9. Record the three-letter mnemonic code for the collection.

10. Create a donor reference for the accession in the relevant field. This field must link to a Parties record. If the donor/depositor has an existing Parties record (e.g. if the accession is an accrual to an existing collection), find it by adding part of the name and searching for the record; when you find it, click the link button to attach it. If a new Parties record is required, complete the record and click the link button to attach it. Similarly if there is a specific main contact for the accession, link to or create another Parties record accordingly. Remember to give the Role in the Parties record as 'Collection Donor' or 'Collection Depositor'.

11. EMU has a dedicated field for recording the size of a digital accession: 'Extent (digital)'. Record the size of the accession. Jacksum outputs size in bytes, so record this; however, it is helpful also to record the approximate size in a more user-friendly form – e.g. gigabytes – taking the size from the transfer list, or from viewing the PST files in Windows Explorer. This records digital size before any appraisal has taken place, so usually a digital accession will reduce in size after processing. This field should therefore be updated after appraisal has occurred.

12. In the Access Conditions field, record any restrictions or embargoes agreed with the donor/depositor, plus any legal or regulatory restrictions (copyright will apply to all digital archives, and data protection to many of them).

13. Leave the Condition Check field empty as this is not directly applicable to digital archives.

14. Use the Summary field to provide a summary description of the accession, primarily a brief note of content, and information about provenance/creator.

15. Use the Full Description field to provide a more detailed description of the nature and contents of the accession.

16. Use the Notes field to provide supplementary and technical information about the accession which is not more appropriately recorded elsewhere, e.g.: information about the technical environment in which the material was created; its format; how transfer

was made; checksum readings where feasible; and a note indicating whether the material will be subject to appraisal at a later stage.

17. If the accession is related to an existing collection, which has a record in the Catalogue Module, link the Accession Lot record to this record by dragging and dropping the Accession record into the Catalogue's Accession field.

18. If the digital accession is for a completely new collection, you will need to consider whether a Catalogue record is compiled to 'register' the collection. This is recommended for anything more than very small/single items. For guidance on Registration (creating a core Catalogue record), refer to the EMU for Archivists guidelines.

19. Once the accession number is assigned, use this number (without slash) as a parent folder for each set of PST folders (and hence files) that make up the accession, e.g. for PST file 001 in folder 001 that belongs to accession 2012/010, use: \<collection code>\2012010\001\001.pst. For an accession folder named a01 in Step 1, simply rename to the correct accession number. NB: DO NOT change the PST folder names, PST filenames or Jacksum report filenames. Renaming the accession folder will normally be done prior to transfer of folders and files from Quarantine to Workbench.

20. You will also need to rename any report files created on the Quarantine PC, and the event logs created on the Workbench PC in Step 1.2, using the given accession numbers.

1.3.2 Step 1.3 folder and file structure

The folder and file structure at the end of this stage is as follows:

On Quarantine PC :

\<collection code>\<accession number>\<reports>\<email sequence number>_virus_report.txt

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt

e.g. \cpa\2012010\reports\001_virus_report.txt
\cpa\2012010\reports\002_virus_report.txt

\cpa\2012010\001\001.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\002\002.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt

On Workbench PC :

\<collection code>\eventlog\premis-localextension-eventlog.xsd

\<collection code>\eventlog\<accession number>_<email sequence number>_eventlog.xml (NB these will need to be copied into email sequence folders after files have been transferred from Quarantine to Workbench; ensure that a copy of the local PREMIS template is also copied).

e.g. \cpa\eventlog\a01_001_eventlog.xml
\cpa\eventlog\a01_002_eventlog.xml
\cpa\eventlog\premis-localextension-eventlog.xsd

Step 1.4: Appraisal and compaction of PST Files

1.4.1 Appraisal using Paraben's Email Examiner

1.4.1.1 Creating a 'case'

Paraben's Email Examiner (v.8) is installed on the Quarantine PC to facilitate examination and appraisal of archival email. The identification of folders for appraisal will be carried out using this software. Some of the terminology is unfamiliar as the software is generally intended for police/law enforcement purposes. Follow these instructions for opening and interrogating each PST file.

1. Open Paraben program and select 'Yes' to the question about installation.

2. Select 'New Case' -> 'Next', or click on 'New' in the toolbar. Work through the steps shown in the New Case wizard.

3. Name the case with the appropriate accession and PST/sequence number (each PST/sequence will form a separate case).

4. Fill in the 'description' field with some information about the case/PST if considered necessary.

5. In the 'Additional Information' box, add your name as the case investigator.

6. On clicking 'Finish', the software will ask you where you want to save the case: each case will be stored at the top level within the Accession folder; name the case file with the accession number and relevant sequence number, i.e. <collection code>\<accession number>\<accession number>_<sequence number>.cemx The .cemx file extension is generated by the software.

7. Select the option to add 'Evidence' to the case.

8. When prompted to select 'Source type', choose MS Outlook database.

9. The software then allows you to browse to the location of the PST file you wish to examine for appraisal and prompts you to name it. Name it with the PST filename, i.e. 001, 002 etc.

10. When prompted, DO NOT select 'Raw mode', or 'Scan database for deleted messages'.

11. Email Examiner allows you to open one or multiple sets of 'Evidence' (i.e. PST files) at once – this is useful for searching, but for appraisal it is advised to open only one PST at a time.

12. Use the software to interrogate and assess the content of each PST file. A tree-view of the directory structure is shown in the Case Explorer pane; right-clicking enables you to perform various actions, including adding and viewing bookmarks. The Data View pane (the large pane to the right) displays the content of the items selected in the Case Explorer pane – e.g. subject lines of emails, with information like sender, recipient, date etc. The Email Data pane (viewer) allows you to view email message contents in different formats; it can render both emails and attachments, but not image files which are embedded in email messages.

1.4.1.2 Bookmarking

Folders (and individual messages) can be bookmarked for any purpose, and the same items can be bookmarked multiple times for different purposes if necessary. For appraisal, bookmarking will usually be used to highlight folders which the curator decides should be securely deleted for the purpose of running an appraisal report.

1. To bookmark a folder, highlight the relevant folder in the Case Explorer screen and right click on it.

2. You will be asked to give the bookmark a name: this could be the name of the action the bookmark is denoting, or the name of the folder, or both. Even if the bookmark is not given the folder name, it is linked to the relevant folder: right-clicking on the bookmark and selecting 'Edit bookmark', and 'Properties' shows the folder path for the folder it refers to. Double-clicking the bookmark will take you to the contents of the folder itself.

3. You can include the reason for identifying the folder for deletion in the bookmark's 'Description' field, e.g. 'DPA', 'Not of long-term research value'.

4. You can organise bookmarks into folders if required. Just right-click on the Bookmarks pane; choose 'Create folder', and click and drag contents into folders.

1.4.1.3 Generating an appraisal report

An appraisal report can be generated based on the bookmarks which have been created.

1. The appraisal report should be stored in the <reports> folder which **may** have been created at Step 1.2.2 if any virus reports were generated at that stage. If this folder does **not** already exist, create it now, i.e.: \<collection code>\<accession number>\<reports>\

2. From the toolbar, select 'Generate Report'. This runs the Reports Wizard.

3. In the first window, select 'HTML Evidence Summary Report'. Although this type of report is not ideal for presenting to donors/depositors in terms of its user-friendliness, it is the best option that Email Examiner offers.

4. In the same window, browse to the destination folder created in step 1 above.

5. In the next window, add your name as the investigator.

6. Click ‘Finish’ and the report will start running. It can take some time to complete.

7. On completion, Email Examiner will have created a subfolder within the <reports> destination folder. The subfolder takes the name of the Case (e.g. 001) as its name. Inside the subfolder is the report itself in HTML format, i.e.\<collection code>\<accession number>\<reports>\<accession number>_email sequence number>.html; and a separate checksum file. The HTML file includes:
    - investigator information;
    - a list of every bookmark, including: its ‘source’ (i.e. the folder path which allows you to identify the location of the folder or item); and the ‘description’ of why the item was bookmarked for appraisal;
    - and a report on the PST file content as a whole, which consists of folder name; total messages (including content of subfolders); total messages directly stored in the folder; number of unread messages; size of each folder in bytes; and number of attachments.

8. This report can then be used by the curator to navigate to and delete the folders identified for appraisal using Outlook (see Step 1.4.2 below); it can also be used as a formal appraisal report for presenting to the donor/depositor. The report will not be transferred to the Workbench PC along with the appraised files, but should be securely stored in the relevant collection file.

NB: in future, the Email Examiner report might also be considered as an alternative to PST Reporter (a free tool – see Step 1.6) for extracting structured metadata about PST files.

1.4.2 Deletion and compaction using Outlook

As a forensic tool, Paraben’s Email Examiner does not enable the deletion of folders and messages within a PST file. Currently, this must be done by opening the PST file in a dummy Outlook account, deleting the relevant folders, and then ‘compacting’ the PST.

1. Go into the dummy Outlook account, and load the first master PST file for appraisal. It is recommended that only one PST data file is open in Outlook at any one time. Instructions for loading data files are version specific. In MS Outlook 2010, use: File -> Open -> Open Outlook Data File -> browse for file. This will be the first master PST, at \<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst

2. Referring to the appraisal report generated by Email Examiner, delete the folders (and, in exceptional cases, individual emails) flagged for disposal.

3. Delete all messages and mailbox folders from the ‘Deleted items’ folder.

4. Now the file must be compacted to ensure that none of the deleted items are retained in the preservation master. In MS Outlook 2010, go to: File -> Account Settings -> In Data Files tab, select relevant PST -> select Settings option -> click on Compact Now button.

5. The compaction process can take some time and may need to be left running overnight.

6. Once the compaction is complete, unload the PST data file from Outlook.

7. A fixity check should now be run on all the PSTs prior to their transfer to network storage. See Section 1.1.3 for instructions on running Jacksum. Name the fixity check file according to naming conventions with the suffix …_f3.txt or …_f4.txt, as appropriate.

8. Record ALL events in the appropriate Event Log on the Workbench PC (see Step 1.19 for instructions).

9. Go back to the accession record in EMU, and alter the digital size of the accession to reflect the newly compacted size of each PST (see Step 1.3.1, point 11).

1.4.3 Step 1.4 folder and file structure

At the end of this stage the following structure will exist on the Quarantine PC:

(NB: Only the sequence-level subfolders are transferred to the Workbench PC at Step 1.5, NOT any reports or other temporary files).

\<collection code>\<accession number>\<accession number_email sequence number>.cemx

\<collection code>\<accession number>\<reports>\<email sequence number>_virus_report.txt
\<collection code>\<accession number>\<reports>\<accession number>_email sequence number>.html
\<collection code>\<accession number>\<reports>\<accession number>_email sequence number>.html.MD5

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt

e.g.

\cpa\2012010\2012010_001.cemx
\cpa\2012010\2012010_002.cemx

\cpa\2012010\reports\001_virus_report.txt
\cpa\2012010\reports\2012010_001.html
\cpa\2012010\reports\2012010_001.html.MD5

\cpa\2012010\reports\002_virus_report.txt
\cpa\2012010\reports\2012010_002.html
\cpa\2012010\reports\2012010_002.html.MD5

\cpa\2012010\001\001.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt

\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt

On Workbench PC :

\<collection code>\eventlog\premis-localextension-eventlog.xsd
\<collection code>\eventlog\<accession number>_<email sequence number>_eventlog.xml (NB these will need to be copied into email sequence folders after files have been transferred from Quarantine to Workbench; ensure that a copy of the local PREMIS template is also copied).

e.g. \cpa\eventlog\a01_001_eventlog.xml
\cpa\eventlog\a01_002_eventlog.xml
\cpa\eventlog\premis-localextension-eventlog.xsd

Step 1.5: Transfer and verify PST files

1.5.1 Transfer and verification process

The compacted PST files must now be moved to the networked Workbench PC by means of the encrypted partition on the removable hard drive, using the following instructions.

1. Connect the removable hard drive to the Quarantine PC, and follow the instructions for 'mounting' the the drive using TrueCrypt given in Step 1.1.1.

2. Navigate to the 'Email Archive' container created in Step 1.1.1. Ignore the 'originals' folder which you created to transfer originals from the creator's PC to the hard drive. Create a new folder at the root named with the three-letter collection code, e.g. cpa. Within this create one or more subfolders for each accession.

3. Copy and paste each email sequence folder into the relevant accession folder on the removable drive, so that each new preservation master PST, and all its associated fixity check records are copied. DO NOT copy any of the reports or other temporary files.

4. 'Dismount' the TrueCrypt application as outlined in Step 1.1.6, and safely disconnect the hard drive from the Quarantine PC.

5. Log on to the Workbench PC, connect the removable hard drive, and mount the encrypted container as above. Navigate to the collection folder on the chosen partition of the Workbench PC which was created at Step 1.2.2. Do NOT store any archive material on a network drive; ensure you use a partition of the Workbench hard drive.

6. Copy and paste each accession folder (if more than one) from the removable hard drive to the collection folder on the relevant partition.

7. A further fixity check should then be run on each PST. At this stage, the combined CRC/checksum will be run for verification purposes, but a second checksum (using the SHA-384 algorithm) will also be run, and the latter will form the principal means of verifying the files from this point on because it is compatible with other tools used on the Workbench PC and in archival storage. The two fixity checks should be saved as 001_f4.txt, 001_f5.txt, etc. OR 001_f5.txt, 001_f6.txt etc, depending on how many checks have already been run on

each PST. Run Jacksum on the copied PST files on the networked drive using the following command:

a. Load up a MS Windows command line window (type 'cmd' under the Start menu -> Run).

b. Type the following into the command line window (NB: you need to press return and wait for the first sha1+crc32 command to run before you can enter the second sha384 command):

<drive letter>: [enter]

cd \<collection code>\<accession number\<email sequence number>\ [enter]

java –jar c:\jacksum\jacksum.jar –a sha1+crc32 –m –o "<email sequence number>_f5.txt" "<email sequence number>.pst" [enter]

java –jar c:\jacksum\jacksum.jar –a sha384 –m –o "<email sequence number>_f6.txt" "<email sequence number>.pst" [enter]

8. Confirm that Jacksum ran correctly by viewing the output files using a text editor such as Windows Notepad and comparing the pre- and post-transfer sha1+crc32 readings.

9. Finally, create a **copy** of each PST file in the same folder and rename this <email sequence number>_copy.pst. This file is IMPORTANT and will be used by Aid4Mail and PST Reporter to extract metadata. These tools need to run on a copy because they change a PST file when they access it and we wish to archive the master file unaltered.

10. Record actions in event log (see Step 1.19 for instructions).

11. At this stage the Event Logs for each PST file should also be cut and pasted from the event log folder into each PST folder, and renamed so the file title takes the form <email sequence number>_eventlog.xml (i.e. the accession number is now omitted from the filename). Remember also to copy the local version of the PREMIS schema into each folder. The empty eventlog folder can then be deleted.

1.5.2 Step 1.5 folder and file structure

At the end of this stage the following file structure should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst

\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd

Step 1.6: Metadata extraction (PST Reporter)

1.6.1 Installing PST Reporter

If not already installed on the Workbench PC, download Kernel Outlook PST Reporter from http://www.nucleustechnologies.com/outlook-pst-reporter.html

Run the downloaded file "freekerneloutlookpst reporter.exe" to install PST Reporter. Once installed, you can run PST Reporter from the desktop icon.

1.6.2 Running PST Reporter

Assuming PST Reporter is preinstalled on the Workbench desktop, do the following:

1. Before running PST Reporter ensure a connection to the pst files that you wish to process exists (Start menu -> Control Panel -> Mail icon -> Email accounts -> Data files tab -> Click on 'Add' and browse for relevant PST file). Also you MUST ensure you connect to the COPY PST (named <email sequence number>_copy.pst) created in Step 1.5 and NOT the master copy. This is because PST Reporter changes the PST file during processing and we wish to archive the original file unaltered.

2. Start PST Reporter from the shortcut on the Workbench desktop.

3. Unselect all PSTs (which is how it starts up) except the one(s) you are interested in. Click "Generate Report".

4. Select HTML as the output format when prompted.

5. Enter the output folder which needs to be \<collection code>\<accession number>\<email sequence number.

6. Name the output file using the convention <email sequence number>_pstr.html.

7. Generate the Report.
8. When PST Reporter has finished, the HTML file will load into the default web browser (NB: this can be slow to load).
9. Close down PST Reporter (click “Finish” then “Exit”, confirm “Yes” when prompted).
10. Unload the connection to the data file.

1.6.3 The PST Report

The report from PST Reporter is used as one of two key sources of information about the contents of each PST file. It is used to check that anything extracted from the PST to the file system is correct and to verify that migrations are successful and accurate.

The figure below illustrates an example PST Reporter report.

Main report of OST/PST folder(s) details created at 11:03:31 31/07/2012

S.No	Folder Name	Total Item Count	Total Item Size	Item's Type	Number of Read Item	Number of unread Item	Total Attachment Count	Embedded Attachment Count	Shortcut Attachment Count	Total Attachment Size
1	Phil.Butler-3@manchester.ac.uk\Inbox\Mail Subscriptions\	19	207 Kb	Email (19)	19	0	0	0	0	0 Kb

Report of Folder "Phil.Butler-3@manchester.ac.uk\Inbox\Mail Subscriptions" of S.No. "1"

Sender Name List

S.No	Sender Name	Sender Email	Total Mails	Total Mail Size	Attachment Count	Attachment Size
1	dspace-tech-bounces@lists.sourceforge.net	dspace-tech-bounces@lists.sourceforge.net	2	23 Kb	0	0 Kb
2	dspace-general-bounces+p.butler=manchester.ac.uk@mit.edu	dspace-general-bounces+p.butler=manchester.ac.uk@mit.edu	1	10 Kb	0	0 Kb
3	dspace-general-bounces@mit.edu	dspace-general-bounces@mit.edu	1	10 Kb	0	0 Kb
4	Windows Live ID	support@passport.msn.com	1	11 Kb	0	0 Kb
5	LISTSERV@LISTSERV.INDIANA.EDU	LISTSERV@LISTSERV.INDIANA.EDU	1	13 Kb	0	0 Kb
6	JISCMAIL LISTSERV Server (14.5)	LISTSERV@JISCMAIL.AC.UK	5	64 Kb	0	0 Kb
7	confirmations@emailenfuego.net	confirmations@emailenfuego.net	1	8 Kb	0	0 Kb
8	majordomo@ecs.soton.ac.uk	majordomo@ecs.soton.ac.uk	5	49 Kb	0	0 Kb
9	SPARC-IR administration	SPARC-IR-request@arl.org	1	12 Kb	0	0 Kb
10	SPARC-IR administration	SPARC-IR-digest@arl.org	1	7 Kb	0	0 Kb

Attachment Type List

S.No	Attachment Type Name	Attachment Type Count	Total Attachment Type size

1.6.4 Step 1.6 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt

\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html

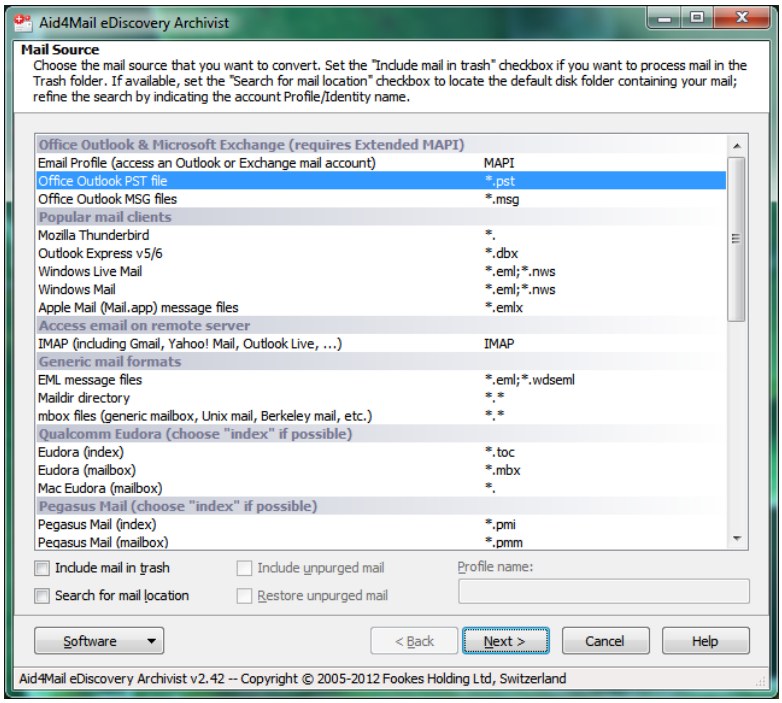
Step 1.7: Metadata extraction (Aid4Mail)

1.7.1 Aid4Mail requirements:

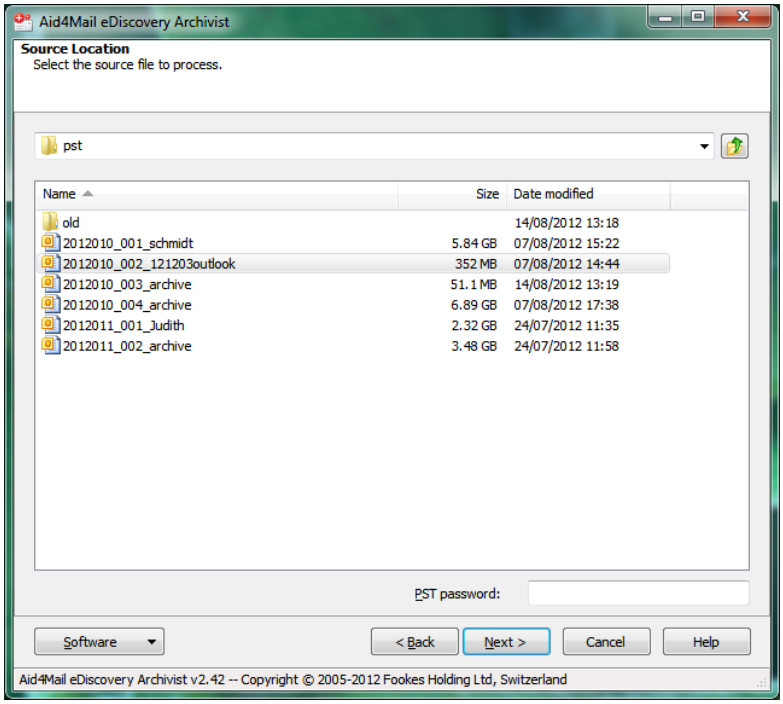
- Requires the current version of Aid4Mail eDiscovery Archivist (http://www.aid4mail.com/) to be preinstalled onto the Workbench PC.
- We have implemented a custom Aid4Mail script which extracts email metadata in a suitable format for email sequence objects. This script outputs the required metadata as XML. The script needs to be loaded into Aid4Mail once after its installation. The script is located in the folder \Aid4Mail-Scripts\ExtractMetadataForSequencePreservation_v1-6.s4o. To load this script into Aid4Mail, run Aid4Mail, click the “Software” button at the bottom left, select “Install Scripts…”, browse to the folder and file name above, and click “Open” .

1.7.2 Running Aid4Mail

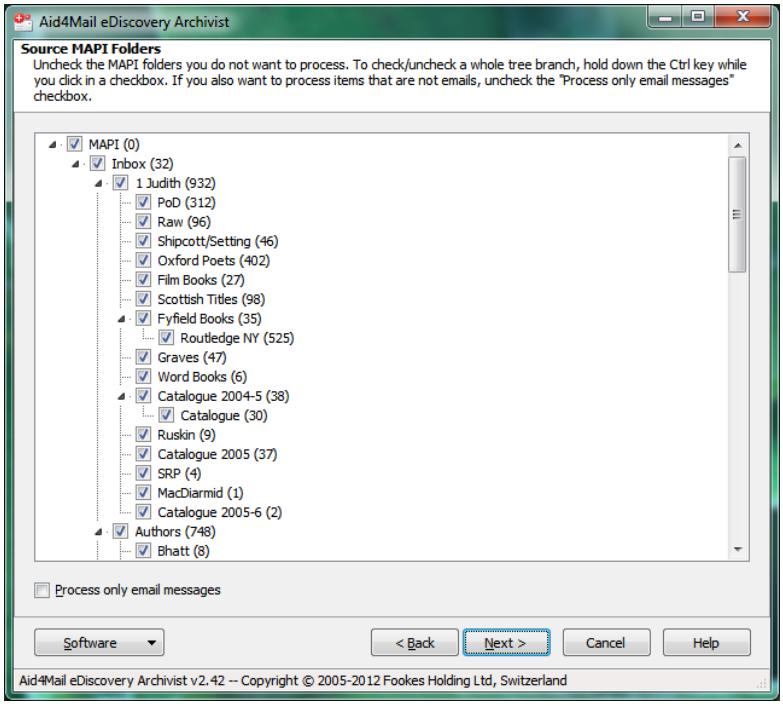
1. Run Aid4Mail (Normally located in Windows start menu under the submenu Fookes Software, or in tool bar at bottom of screen).
2. Make sure in the Advance Options that “Header-based MD5 File Names” is NOT ticked, select “Software” button -> “Advanced Options” -> “Header-based MD5 File Names” (NB: this is a toggle so if already unticked DO NOT select it).
3. Select Mail Source: “Office Outlook PST file”, click “Next >”.



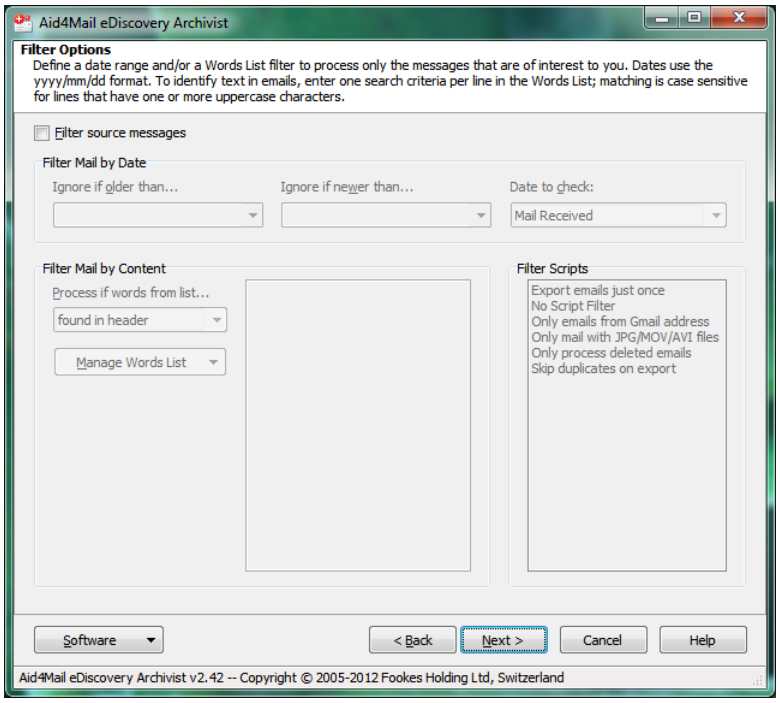
4. Select Source Location: browse folder structure for source PST file. Select one PST file. Click “Next > ”. Make sure you connect to the COPY PST (named <email sequence number>_copy.pst) created in Step 1.5 and NOT the master copy. This is because Aid4Mail changes the PST file during processing and we wish to archive the original file unaltered.



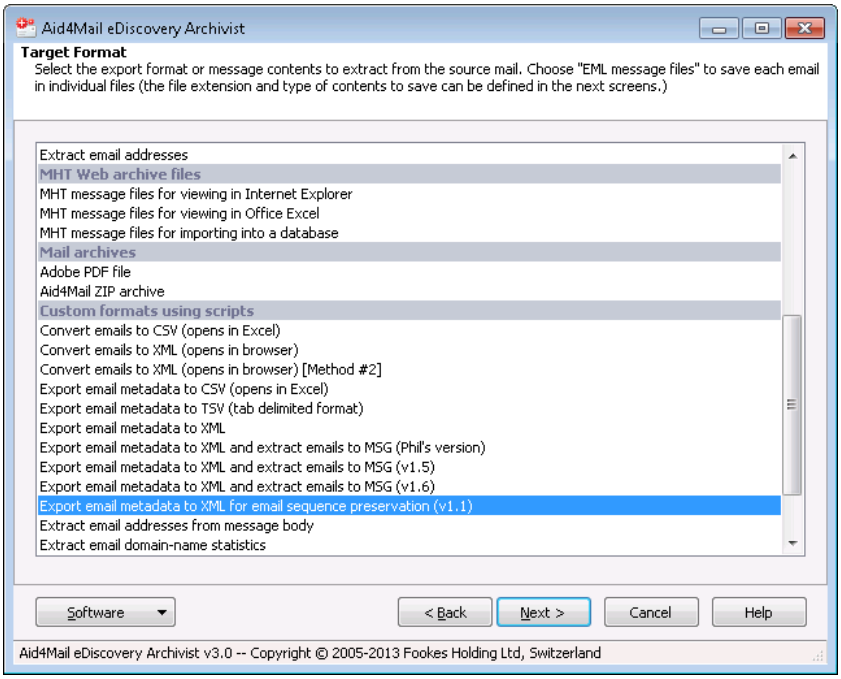
5. Under “Source MAPI Folders”, select and deselect the source email folders that you wish to extract metadata for.



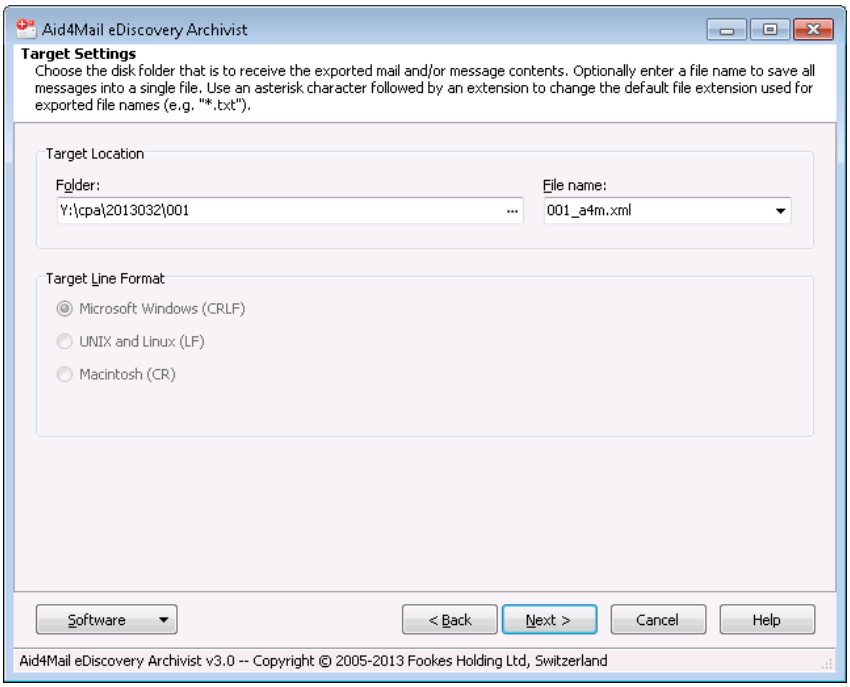
6. Filter Options: Enter any filtering options you wish to apply to the emails that you want to extract metadata about; this option will not usually be used. Click “Next >”.



7. Target Format: Under the “Custom formats using scripts” heading, select “Export email metadata to XML for email sequence preservation (v1.1)”. Click “Next >”.

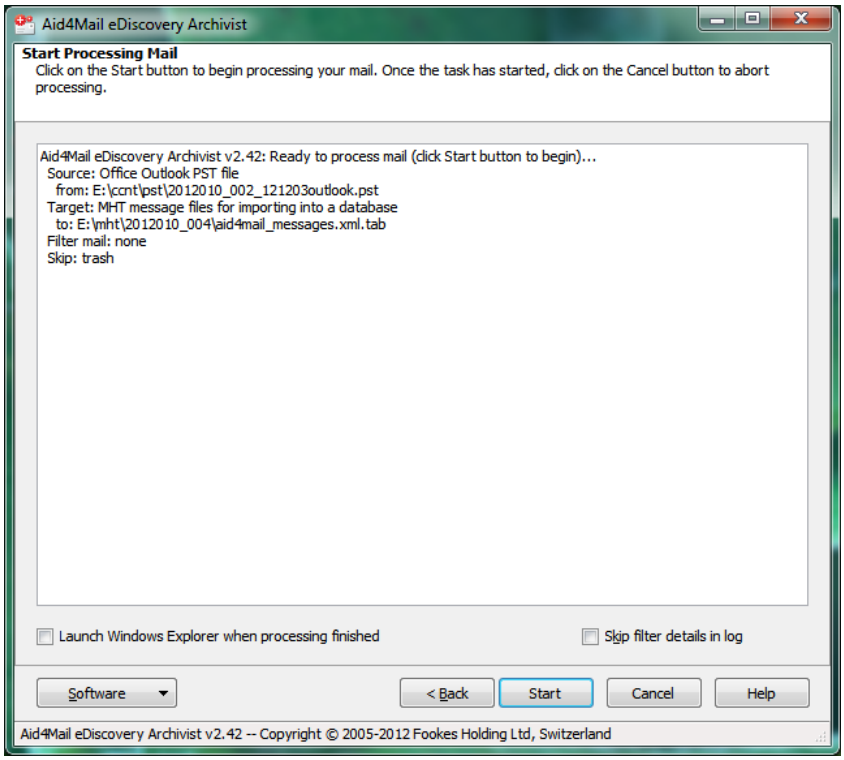


Target Settings: Enter a folder path and file name for the metadata extract. Use the same folder as used to store the PST file i.e. \<collection code>\<accession number>\<email sequence number> Name the output file using the naming convention <email sequence number>_a4m.xml. Click “Next >”.



8. Start Processing Mail: This stage may take some time depending on the size of the PST file. Aid4Mail reports progress during processing. You can leave the process running overnight but you must ensure that your machine does not shutdown automatically; lock the screen if

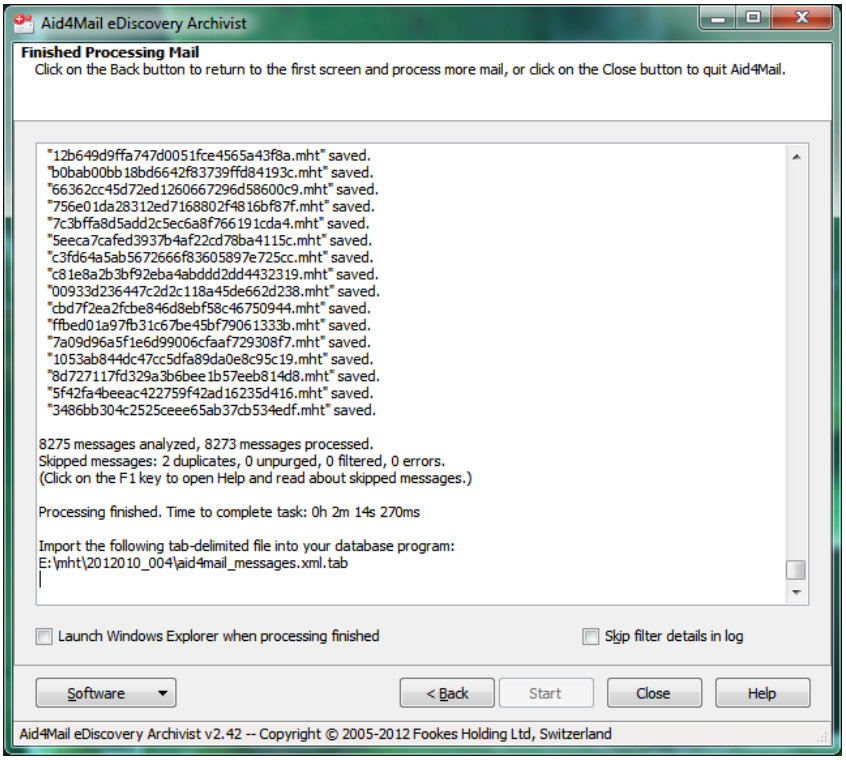
leaving overnight (by pressing the windows button and 'l' keys). Click "Start" to begin processing Aid4Mail.



9. Finished Processing Mail: Once Aid4Mail has completed processing the PST file, it will summarise how many messages it processed, how many it skipped and the number of duplicates (duplicates occur when the MD5 checksum file name is not unique; this can happen when the emails are exact duplicates, which might occur as a result of auto-archiving).

10. Locate and copy the Email sequence-level EAD template. This can be found at \XML-Templates\EAD_EmailSequence_Template.xml. Paste it into the relevant email sequence folder and rename it to <email sequence number>_eadv1.xml.

11. Copy the lines from the Aid4Mail output which summarise the number of processed messages and number of duplicates skipped. This needs to be added to the EAD record for the sequence: there is a paragraph within the <scopecontent> element of the Email Sequence EAD template which refers to this information; simply paste the figures as indicated in the template and save the file.



12. Aid4Mail names the output file with a .tab. extension. Rename this file so that it has a .xml extension

1.7.3 Step 1.7 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt

\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml

Step 1.8: Metadata extraction (FITS)

FITS produces some essential technical and preservation metadata for the PST files. This step, and the following steps (1.9-1.10), can be run separately, but there is also a batch file which can run all three steps together. This can be found at \<collection code>\BAT-Scripts\EmailSequence_Steps_8-9-10.bat.

1.8.1 Creating a batch file for running Steps 1.8-1.10

1. Load the DOS command window (enter 'cmd' at Start menu -> run).

2. Type "y:\<collection code>\BAT-Scripts\EmailSequence_Steps_8_9_10.bat" y: <collection code> <accession code> <email sequence number>

   e.g. for a single PST in collection 'cpa', accession '2012010' and sequence code '001', type the following "y:\cpa\BAT-Scripts\EmailSequence_Steps_8_9_10.bat" y: cpa 2012010 001 (NB: note the presence of space character between the collection code, accession code and email sequence number).

   NB: Wait until this command finishes running before you type anything else into the command window. If you would like to process more than one PST simultaneously then open up a new command window for each PST.

   The above example uses the drive letter "y"; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.

1.8.2 Creating a batch file for running FITS on multiple PST files

The following instructions apply if you choose NOT to run steps 1.8-1.10 using the batch file.

If you are running FITS on multiple email sequence PST files at once, you can create a smaller batch file to carry out this operation, as follows:

1. Load up notepad.

2. Create a file in the BAT-Scripts folder for your collection, named \<collection code>\BAT-Scripts\pst2fits.bat.

3. In this file, create a line with the following details for each PST you want to run FITS against (NB: this example uses the drive letter "y"; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project):

"fits.bat" -i "y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst" –o "y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml"

For example if you are processing two PST files, 001.pst and 002.pst, in the accession 2012010, and one PST file, 001.pst, in the accession 2012011, you would enter the following into the pst2fits.bat file (ALL within the collection CPA):

"fits.bat" -i "y:\cpa\2012010\001\001.pst" -o "y:\cpa\2012010\001\001_fits.xml"

"fits.bat" -i "y:\cpa\2012010\001\002.pst" -o "y: \cpa\2012010\001\002_fits.xml"

"fits.bat" -i "y: \cpa\2012011\001\001.pst" -o "y:\cpa\2012011\001\001_fits.xml"

4. Save the batch file pst2fits.bat.

5. Load the DOS command window (enter 'cmd' at Start menu -> run).

6. Change your working folder to the folder where FITS is installed (normally c:\fits or c:\program files\fits) and at the command prompt, enter c: <ENTER> cd \fits <ENTER>

7. Type in the batch file name, pst2fits.bat, and press <ENTER>.

8. The batch file should run and create a FITS report (.xml) in the specified accession folders - one report for each PST file. Verify that the FITS reports have been created correctly.

9. Enter exit to complete the process.

1.8.3 Processing a single PST file

Alternatively if you wish to run FITS against a single PST file, you can skip steps 1 to 5 in Section 1.8.2, and type directly into a DOS command window, choosing one FITS command as described in step 3, i.e.:

1. Load the DOS command window (enter 'cmd' at Start menu -> run).

2. Change your working folder to the folder where FITS is installed (normally c:\fits or c:\program files\fits) and at the command prompt, enter c: <ENTER> cd \fits <ENTER>.

3. Type in the following command: "fits" -i "y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst" –o "y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml"

e.g. "fits" -i "y:\cpa\2012010\001\001.pst -o "y:\cpa\2012010\001\001_fits.xml"

(NB: this example uses the drive letter “y”; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project).

4. Enter exit to complete the process.

1.8.4 Step 1.8 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\001_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml

Step 1.9: Tidy HTML (PST Reporter)

1.9.1 Running the Tidy software

PST Reporter produces invalid HTML. This needs to be 'tidied' in order to validate it prior to ingest into the archive. To do this you will need to install HTML Tidy (This can be downloaded from http://tidy.sourceforge.net/#binaries – pick the first option, for Windows XP, 7, 8) on the Workbench machine (unless it is already installed). It is assumed HTML Tidy is installed in the folder c:\program files (x86)\tidy.

1. Load DOS command window (in Windows go to Start menu -> Run -> cmd, enter).

2. At the command prompt enter the following.

"c:\program files (x86)\tidy\tidy.exe" -asxhtml -o "y: \<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml" "y: \<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html"

For example if you are processing a PST Reporter file, 001.html, in the accession 2012010, all in the collection CPA, you would enter the following:

"c:\program files (x86)\tidy\tidy.exe" -asxhtml -o "y:\cpa\2012010\001\001_pstrtidy.xhtml" "y:\cpa\2012010\001\001_pstr.html"

NB: the above example uses the drive letter "y"; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.

3. Press <ENTER>.

4. The tidy command should run and create a tidied report (_pstrtidy.xhtml) in the specified collection/accession folder. Verify this tidied report file has been created correctly.

5. Exit to complete the process.

1.9.2 Step 1.9 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst

\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml

\cpa\2012010\002\002.pst
\cpa\2012010\002\001_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml

Step 1.10: Verify Aid4Mail metadata extraction and transform to EAD (Email Sequence)

1.10.1 Verification and transformation

Aid4Mail produces XML formatted metadata for each PST file. This needs to be converted to EAD in order to produce a basic catalogue record for the email sequence. The output is also verified against metadata extracted by PST Reporter. The conversion to EAD is a transformation of the Aid4Mail XML using an XSLT and combining this with the already existing EADv1 file for each email sequence. The following assumes that:

- Java is installed on the Workbench machine (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com) has been installed on the Workbench PC.
- The EAD file is available for processing in the folder \<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml (see Step 1.7).
- Aid4Mail email metadata extracts are available (see Step 1.7).
- The transformation may fail the first time you run it. This can be the case when the Aid4Mail extract creates an XML file that has incorrectly encoded UTF-8 characters. This has been seen occasionally in file attachment names and subject lines. To proceed you need to open the <email sequence number>_a4m.xml in an XML editor, e.g. Oxygen, and edit this file by

removing the rogue characters. These can be identified using Oxygen's XML auto-validation and line red-flagging features; they can be as basic as additional blank spaces between words. Once you have valid XML you should resave this file and over-write the original version. This should resolve the character encoding issue. See Step 2.2 for further information about character encoding issues.

1. Load the DOS command window (enter "cmd" at Start menu -> run).
2. Change your working folder to the relevant email sequence folder e.g. y:\<collection code>\<accession number>\<email sequence number> by entering at the command prompt y: <ENTER> cd \cpa\2012010\001<ENTER>. NB: this example uses the drive letter "y"; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.
3. Type in the following command and press <ENTER>.

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml" -xsl:"y:\<collection code>\XSLT-Transforms\ Aid4MailToEmailSequenceEAD.xsl" -o:"y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml" a4mfp=y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml pstrfp=y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml

For example if you are processing an EAD XML file in the folder \cpa\2012010\001, you would enter the following at the command prompt:

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\cpa\2012010\001\001_eadv1.xml" -xsl:"y:\cpa\XSLT-Transforms\ Aid4MailToEmailSequenceEAD.xsl" -o:"y:\cpa\2012010\001\001_eadv2.xml" a4mfp=y:\cpa\2012010\001\001_a4m.xml pstrfp=y:\cpa\2012010\001\001_pstrtidy.xhtml

4. For verification purposes, inspect the <email sequence number>_eadv2.xml file. Consider the following:

- The number of folders reported by PST Reporter and Aid4mail should normally match, but Aid4Mail only counts folders with email messages in them; this means that if any empty folders have slipped through after the appraisal/compaction process, there will be a disparity.
- The list of folders in PST Reporter and not in Aid4Mail may include folders with no emails in them. Any other folders may indicate an issue with the metadata extraction and should be investigated.
- Normally the report should NOT list any folders in Aid4Mail and not in PST Reporter. Again, if such folders are listed this should be investigated.
- Part of the report includes a section that lists folders where there is a disparity between the number of messages found by Aid4Mail and PST Reporter. If Aid4Mail and PST Reporter report the same number of email messages for all folders, this section will be empty. You may want to check the folders where a disparity exists.

- The number of attachments reported by PST Reporter and Aid4Mail may differ in particular circumstances (this is due to the way these tools count the number of attachments). If particular emails or folders are of special significance and these numbers do not match then it may be necessary to check those entities manually. In general terms Aid4Mail seems to report higher numbers of attachments than PST Reporter so clearly if this is reversed then further checks may be warranted.

5. Once verified you can delete the <email sequence number>_copy.pst. However, if you intend to proceed to archiving at email level then you should retain this file.

1.10.2 Step 1.10 folder and file structure

At the end of this stage the following files should exist on the Workbench PC.

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt

\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml

Step 1.11: Edit EAD (Email Sequence)

1.11.1 Editing the EAD record

Once the automatically generated sections of the EAD record have been populated, it is necessary for the curator to enhance the record manually.

1. Run Oxygen XML editor.
2. Open the EAD file created in Step 1.10.
3. Manually edit sections with comments (NB: do NOT change the structure of the file by adding or removing elements or attributes).
4. NB: size given in the <extent> element is only indicative as Aid4Mail and PST Reporter count different things.
5. Save the resulting EAD file as <email sequence number>_eadfinal.xml

1.11.2 Step 1.11 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst

\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml

Step 1.12: Edit PREMIS (Email Sequence)

1.12.1 Editing the PREMIS record

The PREMIS record includes key preservation metadata about the email sequence. Much of this still needs to be created manually using templates which contain a mixture of drop-down controlled vocabulary elements and free-text fields.

1. Run Oxygen XML editor.

2. Open the PREMIS template file: <collection code>\XML-Templates\PREMIS_Representation-file_EmailSequenceTemplate.xml.

3. Manually edit the sections with comments. For some elements there are drop-down menus, which can be accessed by clicking ctrl+space bar while cursor is inside the element. The content of some elements can be found in the FITS report for the relevant PST file: copy and paste the information from that source. NB: do NOT change the structure of the file by adding or removing elements or attributes.

4. Save this file to \<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml.

1.12.2 Step 1.12 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml

\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml

Step 1.13: Transform EAD to DC (Email Sequence)

1.13.1 Creating a batch file to transform EAD for multiple PST files

Digital objects stored in Manchester eScholar must include a Dublin Core (descriptive metadata) datastream as this is required by Fedora software. A very basic Dublin Core record for each email sequence is created automatically by extracting data already stored in the EAD record.

This stage assumes that:

- Java is installed on the Workbench machine (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com), has been installed on the Workbench PC.
- The EAD source file should have been created at the end of Step 1.11. The file should exist in the folder \<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml.

1. Load up notepad.
2. Create a batch file in the BAT-Scripts folder as follows: \<collection code>\BAT-Scripts\ead2dc_emailseq.bat.
3. Edit this file and create a line with the following details for each EAD record you want to transform to DC. NB: this example uses the drive letter “y”; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml” -xsl:”y:\<collection code>\XSLT-Transforms\EmailSequenceEADToDC.xsl” -o:”y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml”

For example if you are processing two EAD files, 001_eadfinal.xml and 002_eadfinal.xml, in the folder y:\cpa\2012010, you would enter the following into the ead2dc_emailseq.bat file (NOTE: one line per command; do not separate parameters by a carriage return, ONLY enter a carriage return at the end of the whole of each command):

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\cpa\2012010\001\001_eadfinal.xml” -xsl:”y:\cpa\ XSLT-Transforms\EmailSequenceEADToDC.xsl” -o:”y:\cpa\2012010\001\001_dcfinal.xml”

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\cpa\2012010\002\002_eadfinal.xml” -xsl:”y:\cpa\ XSLT-Transforms\EmailSequenceEADToDC.xsl” -o:”y:\cpa\2012010\002\002_dcfinal.xml”

4. Save the batch file ead2dc_emailseq.bat.
5. Load DOS command window (Windows Start menu -> Run -> cmd, enter).
6. Change your work folder to the batch file program folder y:\<collection code>\dc\ by entering at the command prompt, y: <ENTER> cd <collection code>\ dc <ENTER>)
7. Type in the batch file name ead2dc_emailseq.bat and press <ENTER>.
8. The batch file should run and create a series of DC email sequence files in the specified collection/accession folders, one for each EAD file. Verify that the files have been created correctly.
9. Exit to complete the process.

1.13.2 Transforming EAD for a single PST file

Alternatively, if you only wish to transform the EAD for a single PST file, you can run the same command directly into a command window, i.e.:

1. Load the DOS command window (enter ‘cmd’ at Start menu -> run).
2. Enter the following command:

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml” -xsl:”y:\<collection code>\XSLT-Transforms\EmailSequenceEADToDC.xsl” -o:”y:\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml”

3. Press <ENTER> to run the command. Once the process is complete, exit the window.

1.13.3 Step 1.13 folder and file structure

At the end of this stage the following files should exist on the Workbench PC.

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml

\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

Step 1.14: Transform email metadata extract to EAD (Accession)

1.14.1 Transform metadata

Accession-level digital objects do not contain any PST files directly; they are conceptual objects which include descriptive metadata only. They will not always be present, but have been included because the arrangement of the Carcanet Press Archive is based on accessions. The Accession EAD record includes some elements which are automatically populated with metadata from Aid4Mail. The conversion is a transformation of the Aid4Mail XML using an XSLT and combing this with an EAD template. The following assumes that:

- Java is installed on the Workbench PC (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com), has been installed on the Workbench PC.
- The Accession-level EAD template is available for processing at \<collection code>\XML-Templates\EAD_Accession_Template.xml.
- Aid4Mail email metadata extracts are available (see Step 1.7).
- This steps requires valid Aid4Mail email metadata extract files, see step 1.10.1 for comments on non-UTF-8 characters.

1. Load up notepad.
2. Create a file named \<collection code>\<accession number>\file_list.xml
3. Edit this file and create a line with the details of the file path for each Aid4Mail metadata extract file you want to transform to EAD (there will be one extract file per PST/email sequence). For example, an accession in the collection CPA with two Aid4Mail extractions would look like this:

   ```
   <files>
         <file>/cpa/2012011/001/001_a4m.xml</file>
         <file>/cpa/2012011/002/002_a4m.xml</file>
   </files>
   ```

4. Save the file.
5. Load DOS command window (Windows Start menu -> Run -> cmd, enter).
6. Change your working drive letter and folder to the folder where you saved the file in step 3. For instance, if this file was saved in y:\<collection code>\<accession number>\ you would enter y: <ENTER> cd <collection code >\<accession number> <ENTER>.
7. Type in the following command and press <ENTER>.

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\<collection code>\XML-Templates\EAD_Accession_Template.xml” -xsl:”y:\<collection code>\XSLT-Transforms\Aid4MailToAccessionEAD.xsl” -o:”y:\<collection code>\<accession number>\<accession number>_eadv1.xml” filesfilepath=”/<collection code>/<accession number>/file_list.xml”

8. Exit to complete the process.

1.14.2 Step 1.14 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml
\<collection code>\<accession number>\<accession number>_eadv1.xml

\<collection code>\<accession number>\file_list.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt

\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

\cpa\2012010\2012010_eadv1.xml

Step 1.15: Edit EAD (Accession)

1.15.1 Edit EAD

Once the automatically generated sections of the EAD record have been populated, it is necessary for the curator to enhance the record manually.

1. Run Oxygen XML editor.
2. Open the EAD file (ead_v1) created in Step 1.14.
3. Manually edit sections with comments (NB: do NOT change the structure of the file by adding or removing elements or attributes).
4. NB: size given in the <extent> element is only indicative as Aid4Mail and PST Reporter count different things.
5. Save the file to the folder \<collection code>\<accession number>\<accession number>_eadfinal.xml.

1.15.2 Step 1.15 folder and file structure

At the end of this stage the following files should exist on the Workbench PC.

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml

\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml

\<collection code>\<accession number>\<accession number>_eadv1.xml
\<collection code>\<accession number>\<accession number>_eadfinal.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

\cpa\2012010\file_list.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml

Step 1.16: Transform EAD to DC (Accession)

1.16.1 Creating a batch file to transform EAD for multiple accessions

Digital objects stored in Manchester eScholar must include a Dublin Core (descriptive metadata) datastream. A very basic Dublin Core record for each accession is created automatically by extracting data already stored in the EAD record.

This stage assumes that:

- Java is installed on the Workbench PC (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com), has been installed on the Workbench PC.
- The EAD template source file has been created at the end of Step 1.15. The file should exist in the folder \<collection code>\<accession number>\<accession number>_eadfinal.xml.
- To process one accession you may, instead of creating a batch file, run the below commands directly within a command window.

1. Load up notepad.
2. Create a file in the Bat-Scripts folder as follows: \<collection code>\BAT-Scripts\ead2dc_accession.bat.

   Edit this file and create a line with the following details for each EAD record you want to transform to DC. This example uses the drive letter "y"; in reality this is likely to be a different letter, and should be the drive letter on the Workbench PC which is dedicated to your current project. NOTE: when entering the following, use one line per command; do not separate parameters by a carriage return, ONLY enter a carriage return at the end of the whole of each command:

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\<collection code>\<accession number>\<accession number>_eadfinal.xml" -xsl: y:\<collection code>\XSLT-Transforms\ AccessionEADToDC.xsl" -o:"y:\<collection code>\<accession number>\<accession number>_dcfinal.xml"

   For example, if you were processing two different accessions (2012010 and 2012011) in the collection CPA, you would type the following:

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\cpa\2012010\2012010_eadfinal.xml" -xsl: y:\cpa\XSLT-Transforms\ AccessionEADToDC.xsl" -o:"y:\cpa\2012010\2012010_dcfinal.xml"

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\cpa\2012011\2012011_eadfinal.xml" -xsl: y:\cpa\XSLT-Transforms\ AccessionEADToDC.xsl" -o:"y:\cpa\2012011\2012011_dcfinal.xml"

3. Save the batch file ead2dc_accession.bat.
4. Load DOS command window (Windows Start menu -> Run -> cmd, enter).
5. Change your work folder to the batch file program folder \<collection code>\BAT-Scripts\ by entering at the command prompt, y: <ENTER> cd <collection code>\BAT-Scripts <ENTER>.
6. Type in the batch file name ead2dc_accession.bat and press <ENTER>.

7. The batch file should run and create a series of DC accession files in the specified collection folder, one for each EAD file. Verify that the files have been created correctly.

8. Exit to complete the process.

1.16.2 Transforming EAD for a single accession

You can create a batch file as in step 1.16.1 even for a single accession. Alternatively, if you are only dealing with a single accession, you can run the same command directly into the command window, as follows:

1. Load the DOS command window (enter 'cmd' at Start menu -> run).

2. Enter the following command (NB this example uses the drive letter "y"; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project).

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\<collection code>\<accession number>\<accession number>_eadfinal.xml" -xsl: y:\<collection code>\XSLT-Transforms\AccessionEADToDC.xsl" -o:"y:\<collection code>\<accession number>\<accession number>_dcfinal.xml"

For example, if you were processing the accession 2012010 in the collection CPA, you would type the following:

java -Xms512m -Xmx512m -jar "c:\saxon\saxon9.jar" -s:"y:\cpa\2012010\2012010_eadfinal.xml" -xsl: y:\cpa\XSLT-Transforms\AccessionEADToDC.xsl" -o:"y:\cpa\2012010\2012010_dcfinal.xml"

3. Press <ENTER> to run the command. Once the process is complete, exit the window.

1.16.3 Step 1.16 folder and file structure

At the end of this stage the following files should exist on the Workbench PC.

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml

\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml

\<collection code>\<accession number>\<accession number>_eadv1.xml
\<collection code>\<accession number>\<accession number>_eadfinal.xml
\<collection code>\<accession number>\<accession number>_dcfinal.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml

\cpa\2012010\file_list.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml
\cpa\2012010\2012010_dcfinal

1.17: Create/edit EAD (Collection)

1.17.1 Creating an EAD record for a new collection

Collection-level digital objects do not contain any PST files directly; they are conceptual objects which contain descriptive metadata only. At collection-level the EAD record is created entirely manually using a pre-existing template.

1. Run Oxygen XML editor.
2. Open the EAD Collection template file from the Templates folder \<collection code>\XML-Templates\EAD_Collection_Template.xml.
3. Manually edit the template following instructions given in comments (NB: do NOT change the structure of the file by adding or removing elements or attributes).
4. NB: size given in the <extent> element is only indicative as Aid4Mail and PST Reporter count different things.
5. Save the file to the folder \<collection code>\<collection code>_eadfinal.xml.

1.17.2 Editing an EAD record for an existing collection

When adding a new accession to an existing collection, the collection-level EAD record will need to be changed to take into account the addition to the collection. If the changes are relatively minor, they can be made directly to the EAD datastream using the Fedora client (see Step 2.8.4, 'Update datastream in existing Fedora digital object').

If more extensive changes are required, you will need to download a copy of the existing EAD record from Fedora and manually edit it in Oxygen XML Editor. See Step 2.8.4.

Elements which will need editing include:

- "normal" attribute in <unitdate> - amend the covering date. The element content does not need to be edited as it gives later date as [ongoing].
- <extent> - add new number of accessions and overall digital size.
- <materialspec> - only if dealing with different formats from previous accessions.
- <custodhist>
- <acqinfo>
- <scopecontent>
- <arrangement>
- <controlaccess> if relevant.

Save the edited file as \<collection code>\<collection code>_eadfinalv2.xml. Make a note of where you have saved this file, as you will need to navigate to it separately at the point of ingest.

1.17.3 Step 1.17 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml
\<collection code>\<accession number>\<accession number>_eadv1.xml
\<collection code>\<accession number>\<accession number>_eadfinal.xml
\<collection code>\<accession number>\<accession number>_dcfinal.xml

\<collection code>\<collection code>_eadfinal.xml ***OR***
\<collection code>\<collection code>_eadfinalv2.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt

\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

\cpa\2012010\file_list.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml
\cpa\2012010\2012010_dcfinal.xml

\cpa\cpa_eadfinal.xml ***OR***
\cpa\cpa_eadfinalv2.xml

1.18: Transform EAD to DC (Collection)

1.18.1 Creating a batch file to transform EAD

Digital objects stored in Manchester eScholar must include a Dublin Core (descriptive metadata) datastream. A very basic Dublin Core record for each collection is created automatically by extracting data already stored in the EAD record.

This stage assumes that:

- Java is installed on the Workbench PC (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com) has been installed on the Workbench PC.
- The EAD source file should have been created at the end of Step 1.17. The file should exist as the folder \<collection code>\<collection code>_eadfinal.xml ***OR*** \<collection code>\<collection code>_eadfinalv2.xml. Make sure you use the right source and output filenames when running the commands, depending on whether you are processing a completely new collection record or a new version of an existing collection record.

1. Load up notepad.

2. Create a file in the BAT-Scripts folder, as follows: \<collection code>\BAT-Scripts\ead2dc_collection.bat.

3. Edit this file and create a line with the following details for each EAD record you want to transform to DC; in this case, it is a new EAD collection record (i.e. there is no version number).This example uses the drive letter “y”; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\<collection code>\<collection code>_eadfinal.xml” -xsl: y:\<collection code>\XSLT-Transforms\CollectionEADToDC.xsl” -o:”y:\<collection code>\<collection code>_dcfinal.xml

For example if you are processing a collection with the EAD file cpa_eadfinal.xml in the folder y:\cpa you would enter the following into the ead2dc_collection.bat file (NOTE: one line per command, do not separate parameters by a carriage return, ONLY enter a carriage return at the end of the whole of each command):

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\cpa\cpa_eadfinal.xml” -xsl:”y:\cpa\XSLT-Transforms\CollectionEADToDC.xsl” -o:”y:\cpa\cpa_dcfinal.xml”

4. Save the batch file as ead2dc_collection.bat.
5. Load DOS command window (Windows Start menu -> Run -> cmd, enter).
6. Change your work folder to the batch file program folder y:\<collection code>\dc\ by entering, at the command prompt, y: <ENTER> cd <collection code>\ dc <ENTER>).
7. Type in the batch file name ead2dc_collection.bat and press <ENTER>.
8. The batch file should run and create a collection-level DC file in the relevant collection folder. Verify that the file has been created correctly.
9. Exit to complete the process.

1.18.2 Transforming EAD using command line

You can create a batch file as in step 18.1 even if you are dealing with only a single collection record, but bear in mind that this would have to be amended to take account of collection records that are being updated rather than created from scratch (i.e. filenames must take account of the version of the EAD and DC).

Alternatively, you can transform the EAD record by running the same command directly into the command window, as follows:

1. Load the DOS command window (enter ‘cmd’ at Start menu -> run).
2. Enter the following command. As in 1.18.1, this example is for a new collection record, so no version numbers are necessary in the EAD or DC filenames. This example also uses the drive letter “y”; in reality this is likely to be a different letter, and it should be the drive letter for the Workbench partition devoted to your current project.

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\<collection code>\<collection code>_eadfinal.xml” -xsl: y:\<collection code>\XSLT-Transforms\CollectionEADToDC.xsl” -o:”y:\<collection code>\<collection code>_dcfinal.xml

For example if you are processing a collection with the EAD file cpa_eadfinal.xml in the folder y:\cpa, you would type the following:

java -Xms512m -Xmx512m -jar “c:\saxon\saxon9.jar” -s:”y:\cpa\cpa_eadfinal.xml” -xsl:”y:\cpa\XSLT-Transforms\CollectionEADToDC.xsl” -o:”y:\cpa\cpa_dcfinal.xml”

3. Press <ENTER> to run the command. Once the process is complete, exit the window.

1.18.3 Step 1.18 folder and file structure

At the end of this stage the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml

\<collection code>\<accession number>\<accession number>_eadv1.xml
\<collection code>\<accession number>\<accession number>_eadfinal.xml
\<collection code>\<accession number>\<accession number>_dcfinal.xml

\<collection code>\<collection code>_eadfinal.xml ***OR***
\<collection code>\<collection code>_eadfinalv2.xml

\<collection code>\<collection code>_dcfinal.xml ***OR***
\<collection code>\<collection code>_dcfinalv2.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt
\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

\cpa\2012010\file_list.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml
\cpa\2012010\2012010_dcfinal.xml

\cpa\cpa_eadfinal.xml ***OR***
\cpa\cpa_eadfinalv2.xml

\cpa\cpa_dcfinal.xml ***OR***
\cpa\cpa_dcfinalv2.xml

Step 1.19: Event Log

1.19.1 PREMIS

The Event Log template file will be stored on one of the partitions of the Workbench hard drive; it can be copied to whichever partition is being dedicated to the current accession of material and filled in accordingly.

The Event Log for each PST file/email sequence is created at Step 1.2, and renamed with permanent accession number at Step 1.3.

Event information is stored in the PREMIS XML schema.

Each Event document needs some basic PREMIS Object metadata in order to identify the Object (or Manchester eScholar datastream) to which all of the recorded Events relate. This will be created automatically or included in the template document, and consists of:

- <objectIdentifier> This is the eScholar digital object datastream PID
- <objectCharacteristics> <compositionLevel> Defaults to 0 for no bundling or compression

- <objectCharacteristics> <format> <formatDesignation> <formatName>

1.19.2 PREMIS Event metadata

Each Event affecting an Object should be recorded in a separate <event> wrapper within the same overall document.

<eventIdentifier> is locally defined, and entered manually: each Event must have a separate identifier, and these should be a simple numerical sequence, 001, 002 etc.

<eventType> is defined in a drop-down list – simply select the relevant Event from the list. In Oxygen XML Editor use CTRL-space over the element to show the drop-down list of controlled values.

Definitions of eventTypes are as follows:

- Attachment extraction: to record results of extracting attachments from pst files (or other email formats).
- Deletion: i.e. secure deletion of material targeted for disposal.
- Fixity check: running checksum or other algorithm.
- Ingest: ingest of the material into Fedora repository system.
- Metadata extraction: to record the results of running any metadata extraction tools.
- Migration: to record results of migration actions.
- Rename: to record any changes in file or folder names.
- Replication: the process of creating a copy of an object that is (bit-wise) identical to the original.
- Transfer to repository: to record transfer of digital material from the creator to the Library.
- Transfer to quarantine machine: i.e. when material is uploaded from transfer media.
- Transfer to network storage: to record transfer from quarantine PC to secure network storage.
- Validation: i.e. act of comparing an object with a standard and noting compliance or exceptions (e.g. DROID/FITS).
- Virus check: running manual or automatic virus check.

Use <eventDetail> to record any optional additional information about the event (but *not* its outcomes).

<eventOutcome> can be selected from drop-down list.

Use <eventOutcomeDetailNote> to record a plain English description of the outcome of each Event, including any information extracted as part of the Event which isn't held elsewhere. For guidance, the following information might be recorded for each <eventType> identified in the drop-down list:

- Attachment extraction: number of attachments extracted from pst files.
- Deletion: extent of material deleted in bytes and logical extent; reference to where documentation about disposal actions can be found, which would list specifics about what was deleted (usually archive collection file).
- Fixity check: any information generated by checksum tool, e.g. checksum value; file size; original filepath and filename; location of source data; PC used to run tool.
- Metadata extraction: type of metadata extracted (e.g. file names, formats).
- Migration: original and migrated formats.

- Rename: new name.
- Replication: identifier and/or name for the replicated object; identify purpose of replicated object – e.g. working copy.
- Transfer to repository: means of transfer; location to which material transferred.
- Transfer to network storage: details of network drive/location.
- Validation: format name and version as verified by tool.
- Virus check: identify any viruses/problems picked up by virus check.

1.19.2 Step 1.19 folder and file structure

By the end of this stage, the following files should exist on the Workbench PC:

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f6.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml

\<collection code>\<accession number>\<accession number>_eadv1.xml
\<collection code>\<accession number>\<accession number>_eadfinal.xml
\<collection code>\<accession number>\<accession number>_dcfinal.xml

\<collection code>\<collection code>_eadfinal.xml ***OR***
\<collection code>\<collection code>_eadfinalv2.xml

\<collection code>\<collection code>_dcfinal.xml ***OR***
\<collection code>\<collection code>_dcfinalv2.xml

e.g. \cpa\2012010\001\001.pst
\cpa\2012010\001\001_copy.pst
\cpa\2012010\001\001_f1.txt
\cpa\2012010\001\001_f2.txt
\cpa\2012010\001\001_f3.txt
\cpa\2012010\001\001_f4.txt
\cpa\2012010\001\001_f5.txt
\cpa\2012010\001\001_f6.txt

\cpa\2012010\001\001_eventlog.xml
\cpa\2012010\001\premis-localextension-eventlog.xsd
\cpa\2012010\001\001_pstr.html
\cpa\2012010\001\001_a4m.xml
\cpa\2012010\001\001_eadv1.xml
\cpa\2012010\001\001_fits.xml
\cpa\2012010\001\001_pstrtidy.xhtml
\cpa\2012010\001\001_eadv2.xml
\cpa\2012010\001\001_eadfinal.xml
\cpa\2012010\001\001_premisfinal.xml
\cpa\2012010\001\001_dcfinal.xml

\cpa\2012010\002\002.pst
\cpa\2012010\002\002_copy.pst
\cpa\2012010\002\002_f1.txt
\cpa\2012010\002\002_f2.txt
\cpa\2012010\002\002_f3.txt
\cpa\2012010\002\002_f4.txt
\cpa\2012010\002\002_f5.txt
\cpa\2012010\002\002_f6.txt
\cpa\2012010\002\002_eventlog.xml
\cpa\2012010\002\premis-localextension-eventlog.xsd
\cpa\2012010\002\002_pstr.html
\cpa\2012010\002\002_a4m.xml
\cpa\2012010\002\002_eadv1.xml
\cpa\2012010\002\002_fits.xml
\cpa\2012010\002\002_pstrtidy.xhtml
\cpa\2012010\002\002_eadv2.xml
\cpa\2012010\002\002_eadfinal.xml
\cpa\2012010\002\002_premisfinal.xml
\cpa\2012010\002\002_dcfinal.xml

\cpa\2012010\file_list.xml
\cpa\2012010\2012010_eadv1.xml
\cpa\2012010\2012010_eadfinal.xml
\cpa\2012010\2012010_dcfinal.xml

\cpa\cpa_eadfinal.xml ***OR***
\cpa\cpa_eadfinalv2.xml

\cpa\cpa_dcfinal.xml ***OR***
\cpa\cpa_dcfinalv2.xml

Step 1.20: Transform and Package

1.20.1 Requirements

Prior to running the transform and pack operations the following highlighted files need to exist in the email sequence folder.

\<collection code>\<accession number>\<email sequence number>\<email sequence number>.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_copy.pst
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f1.txt

\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f2.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f3.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f4.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_f5.txt
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_fits.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eventlog.xml
\<collection code>\<accession number>\<email sequence number>\premis-localextension-eventlog.xsd
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstr.html
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_pstrtidy.xhtml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_a4m.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv1.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadv2.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_eadfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_premisfinal.xml
\<collection code>\<accession number>\<email sequence number>\<email sequence number>_dcfinal.xml

\<collection code>\<accession number>\file_list.xml

This stage assumes that:

- Java is installed on the Workbench machine (this is normally installed as part of Windows).
- The XML/XSLT processor, Saxon (www.saxonica.com) has been installed on the Workbench PC.

Steps 1.21-1.22 outline how to create and then ingest the collection, accession and email sequence FOXML files and datastreams as created in the steps above.

Steps 1.21-1.23 (creation of the FOXML files) require a licenced version of Saxon. If this is unavailable it is necessary to use the Oxygen XML editor to perform the transformations as described in the Addendum to sections 2.6-2.7. When using Oxygen you should set the parameter values according to the values given in the Saxon commands.

Once you have created the necessary FOXML files you can ingest them. It is possible to ingest FOXML files into a Fedora repository via a number of routes. These are fully documented on the Fedora Commons website at https://wiki.duraspace.org/display/FF/Documentation.

The following steps assume that the FOXML file and all datastreams have been extracted and generated for the collection, accession and/or email sequences being considered.

To execute an ingest, the managed datastream files as described below (e.g. in the case of a collection object this is the file <collection code>_EADfinal.xml) need to be first uploaded to the Fedora enabled server on some temporary file space; doing this makes the files available to Fedora. On Manchester eScholar the temporary file space is limited to <10GB which limits the size of a batch ingest. All files must be removed from this temporary space once the ingest is complete.

Uploading of files is required whenever a FOXML file references a managed datastream. This also applies to collection, accession and email sequence digital objects.

Ingest of the Collection, Accession, Email Sequence and FOXML files can be achieved using the same approach as described in section 2.8 for Email Folder and Email Message FOXMLs.

Step 1.21: Create and ingest Collection Object

If a licenced version of Saxon is unavailable, it is necessary to use the Oxygen XML editor to perform this transformation as described in the Addendum to sections 2.6-2.7.

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>
3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -xsl:”<drive letter>:\<collection code>\XSLT-Transforms\CreateCollectionFOXML.xsl” drive=<drive letter> collection=<collection code> <ENTER>

   To run this transformation using the Oxygen XML Editor you should modify the following XLST parameters

   ```
   <xsl:param name="drive" select="'<drive letter>'"/>
   <xsl:param name="collection" select="'<collection code>'"/>
   ```

4. This command will create a FOXML XML file called uk-ac-man-col-<collection code>.xml for the collection. The file is stored in a subfolder named ‘foxml’ under the collection folder. Check this FOXML file to see it has been created successfully.

To ingest the created FOXML you need to upload the following file to the fedora enabled temporary file space:

<collection code>_eadfinal.xml

Step 1.22: Create and ingest Accession Object/s

If a licenced version of Saxon is unavailable, it is necessary to use the Oxygen XML editor to perform this transformation as described in the Addendum to sections 2.6-2.7.

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>
3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -xsl:”<drive letter>:\<collection code>\XSLT-Transforms\CreateAccessionFOXML.xsl” drive=<drive letter> collection=<collection code> accession=<accession number> <ENTER>

   To run this transformation using the Oxygen XML Editor you should modify the following XLST parameters

   ```
   <xsl:param name="drive" select="'<drive letter>'"/>
   <xsl:param name="collection" select="'<collection code>'"/>
   <xsl:param name="accession" select="'<accession number>'"/>
   ```

4. This command will create a FOXML XML file called uk-ac-man-ema-<accession number>.xml for the accession. The file is stored in a subfolder named 'foxml' under the accession folder. Check this FOXML file to see it has been created successfully.

To ingest the created FOXML you need to upload the following files to the fedora enabled temporary file space:

<accession number>_eadfinal.xml

Step 1.23: Create and ingest Email Sequence Object/s

If a licenced version of Saxon is unavailable, it is necessary to use the Oxygen XML editor to perform this transformation as described in the Addendum to sections 2.6-2.7.

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>
3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailSequenceFOXML.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

   To run this transformation using the Oxygen XML Editor you should modify the following XLST parameters

   ```
   <xsl:param name="drive" select="'<drive letter>'"/>
   <xsl:param name="collection" select="'<collection code>'"/>
   <xsl:param name="accession" select="'<accession number>'"/>
   <xsl:param name="emailsequence" select="'<email sequence number>'"/>
   ```

4. This command will create a FOXML XML file called uk-ac-man-ems-<email sequence number>.xml for the email sequence. The file is stored in a subfolder named 'foxml' under the email sequence folder. Check this FOXML file to see it has been created successfully.

To ingest the created FOXML you need to upload the following files to the fedora enabled temporary file space:

<email sequence number>.pst
<email sequence number>_eadfinal.xml
<email sequence number>_eventlog.xml
<email sequence number>_fits.xml
<email sequence number>_premisfinal.xml
<email sequence number>_pstrtidy.html

Step 1.24 – 1.26: Index Collection, Accession and Email Sequence Objects

Indexing of objects automatically follows the ingest of digital objects. For the Collection, Accession and Email Sequence objects, in addition to the object properties (e.g. the PID, creation date, last modified date) only fields in the Dublin Core and the RELS-EXT datastreams are indexed and made searchable. Fields available for searching using SOLR are

PID	Persistent identifier for object
d.creator	Creator as stored in dc:creator
d.date	Date range content created as stored in dc:date
d.identifier	Content identifier(s) as stored in dc:identifier
d.title	Content title as stored in dc:title
r.iscreatedby.pid	PID of individual who created digital object
r.isderivationof.pid	PID of base digital object for class of objects
r.islastmodifiedby.pid	PID of individual who last modified object
r.ismemberof.pid	PID of related collection object (only in accession and email sequence objects)
x.createddate	Date object was created
x.label	Label for digital object
x.lastmodifieddate	Date object was last modified
x.ownerid	PID of digital object owner
x.state	Access state of object (I=Inactive, A=Active)

Step 1.27: Access (secure)

Currently, the Carcanet Press Email Archive is embargoed to researchers for data protection, IPR and technical reasons. Access to the archive is limited to the curator and a small number of system administrators, and requires authentication by username and password. Read access to metadata is available using Manchester eScholar's SOLR index. This is comprehensive, and includes the full text of every email as well as extensive metadata.

Access for the purpose of editing or deleting any of the digital objects and/or their component datastreams is via the Fedora Administrative Client or the Fedora Application Programming Interface, and is restricted to a small number of system administrators.

Currently, a curatorial tool is under development, due for completion in August/September 2014: this will provide a user-friendly mechanism for the curator to access, interrogate and manage the archive.

Step 1.28: Secure deletion of removable hard drive

Deletion of files on the removable hard drive should not take place until an accession is safely ingested and in archival storage.

Archival files should ONLY be stored in the encrypted container on the removable hard drive. This means that the risk of unauthorised access to sensitive data is so low that the secure deletion software is not necessary for deleting files on this device. Simply follow these steps:

1. Connect the removable hard drive to the Quarantine PC.

2. Navigate to the hard drive.
3. ‘Mount’ the ‘Email Archive’ encrypted container by following the instructions given in Step 1.1.1.
4. Navigate to the ‘originals’ folder which you created in Step 1.1.1, but do not open it.
5. Delete this entire folder and all its content, simply using the delete key.
6. ‘Dismount’ the removable hard drive by following the instructions in Step 1.1.6.

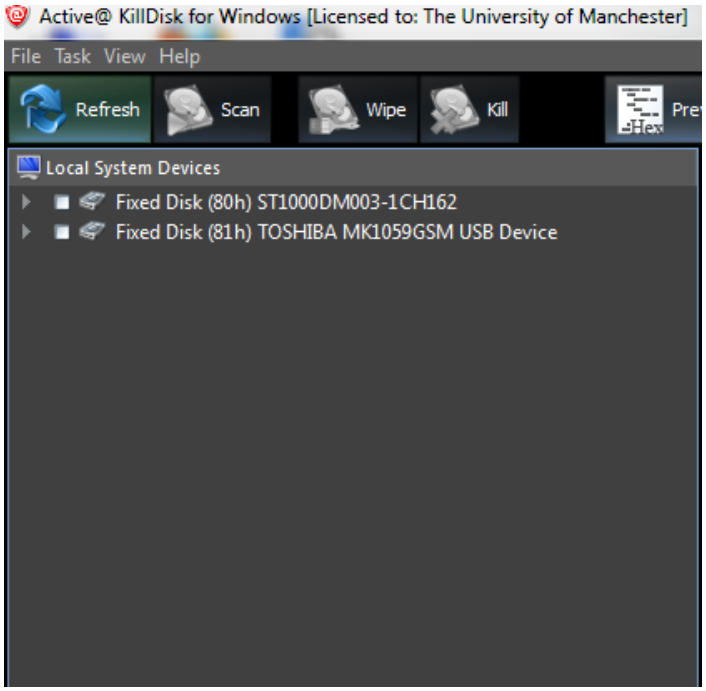
Step 1.29: Secure deletion of Quarantine PC

Deletion of files from the Quarantine PC should not take place until an accession is safely ingested and in archival storage.

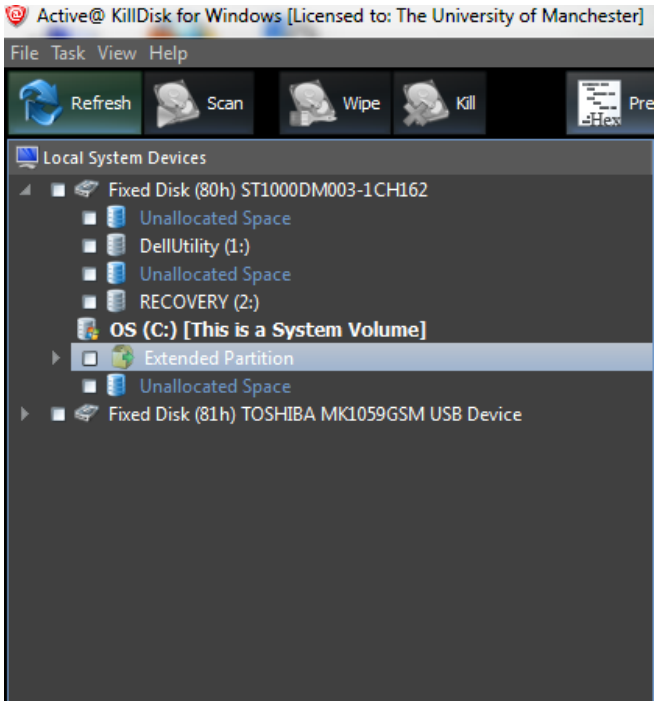
It is crucial that a secure erasure tool is used to delete data. Using the Windows DELETE command simply changes the file names so that the operating system will not look for the files; it does not delete the files themselves.

Secure deletion software called Active@ KillDisk is installed on both the Quarantine and Workbench PCs, and this will be used to delete data by overwriting. It conforms to 15 standards for clearing and sanitizing data. We are using US DoD 5220.22-M, in which the write head passes over each disk sector three times, the first tim with zeros (0x00), the second time with 0xFF, and the third time with random characters. There is one final pass to verify random characters by reading.

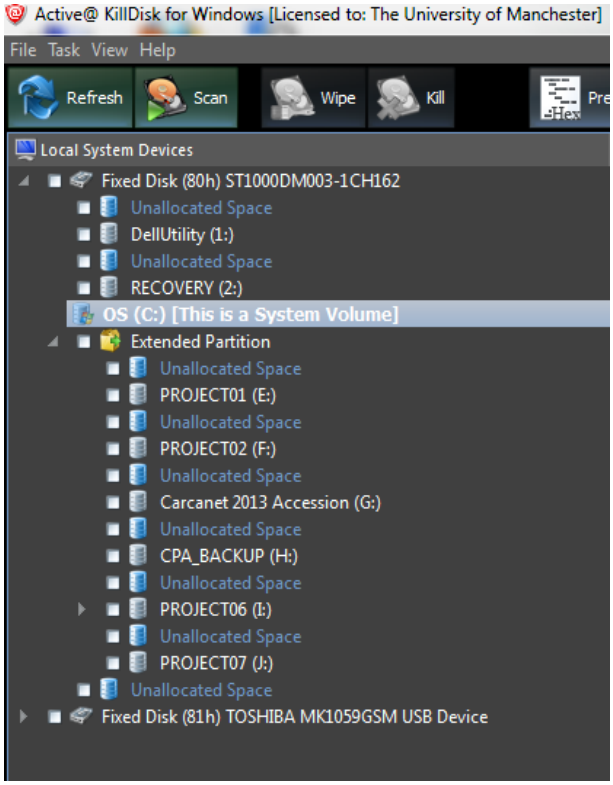
In order to run the software, open KillDisk using the shortcut icon. It shows the structure of the disks in the left-hand pane – the entire hard drive (shown as Fixed Disk 80h in the figure below), and any removable devices if the are connected to the PC (a removable hard drive is shown as Fixed Disk 81h in the figure below).



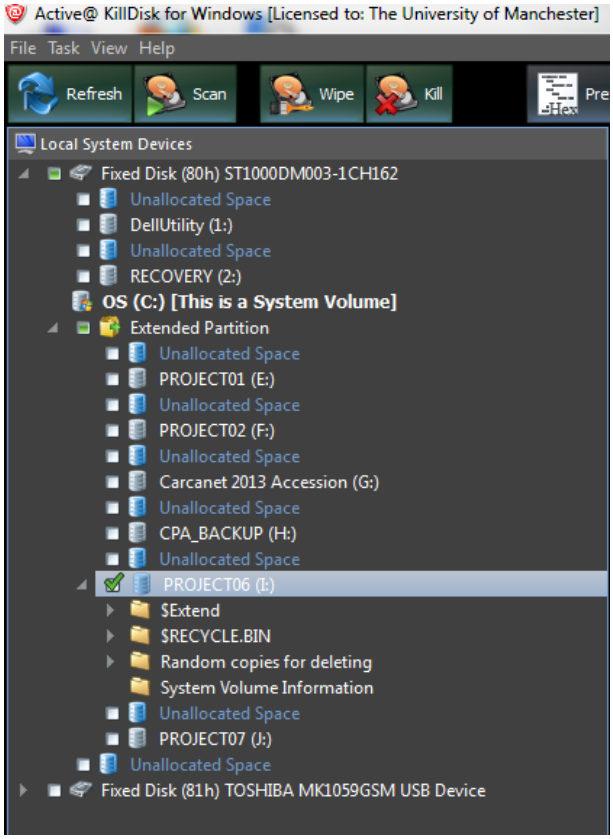
You can expand and contract this structure to focus on specific disk partitions to be deleted. Ignore the c: partition, and the sections marked (1:) and (2:) as you will not need to delete anything from these. The ‘Extended Partition’ contains all the other disk partitions.



Expand the ‘Extended Partition’ to view all the partitions on the hard disk, as below. The software also shows (and overwrites) ‘unallocated’ disk space.



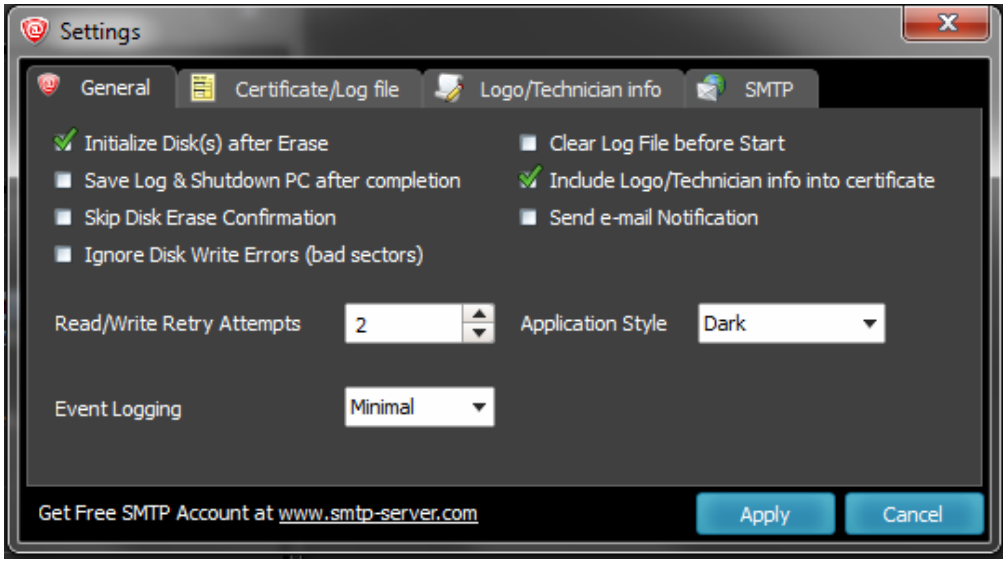
Select (by highlighting the tick box) the partition whose content you wish to delete, then click on the 'kill' icon at the top of the screen.



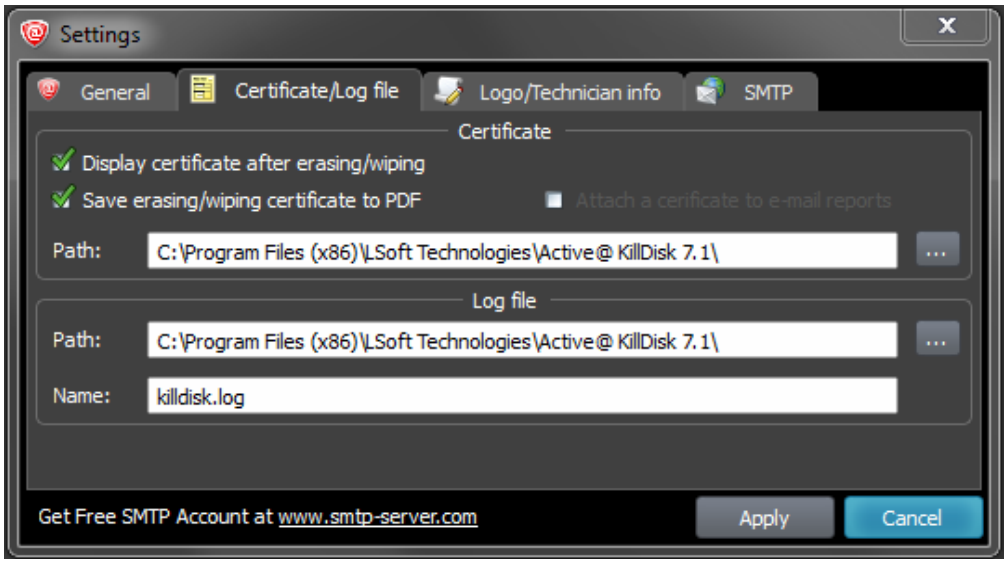
This brings up a box asking you which method of erasing you wish to choose. The chosen method is likely to be highlighted, but if not, select: US DoD 5220.22-M (3 passes, verify). You can direct the

software to perform verification of the surface on the drive to be sure that the overwriting was performed properly. Because this is a long process, you can specify a percentage of the surface to be verified. Leave the 'verification' setting at the default of 10%.

Then click on 'more options' which opens the 'General' tab in the 'Settings' box (below). Ensure that 'Include Logo/Technician info into certificate' is selected; the other selection is by default.



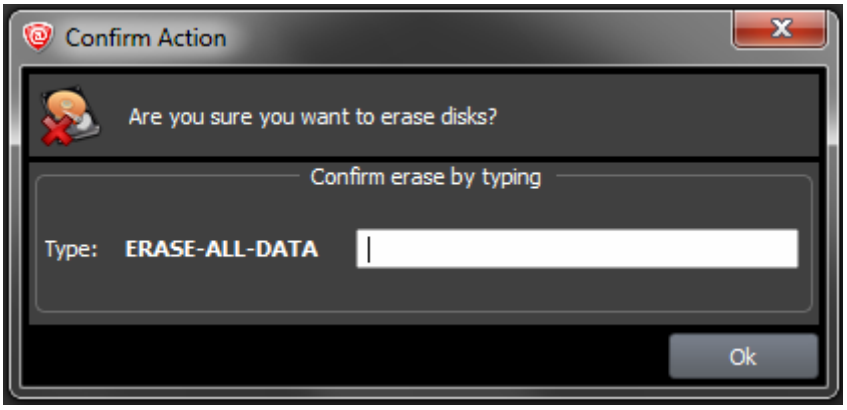
Choose the 'Certificate/Log file' tab. This determines the filepaths for: the certificate which is produced after the erasure has taken place; and for the log file which records details of the erasure process. By default these are stored on the c: drive of the PC in the Program Files (x86) folder, although you can change the location if you wish.



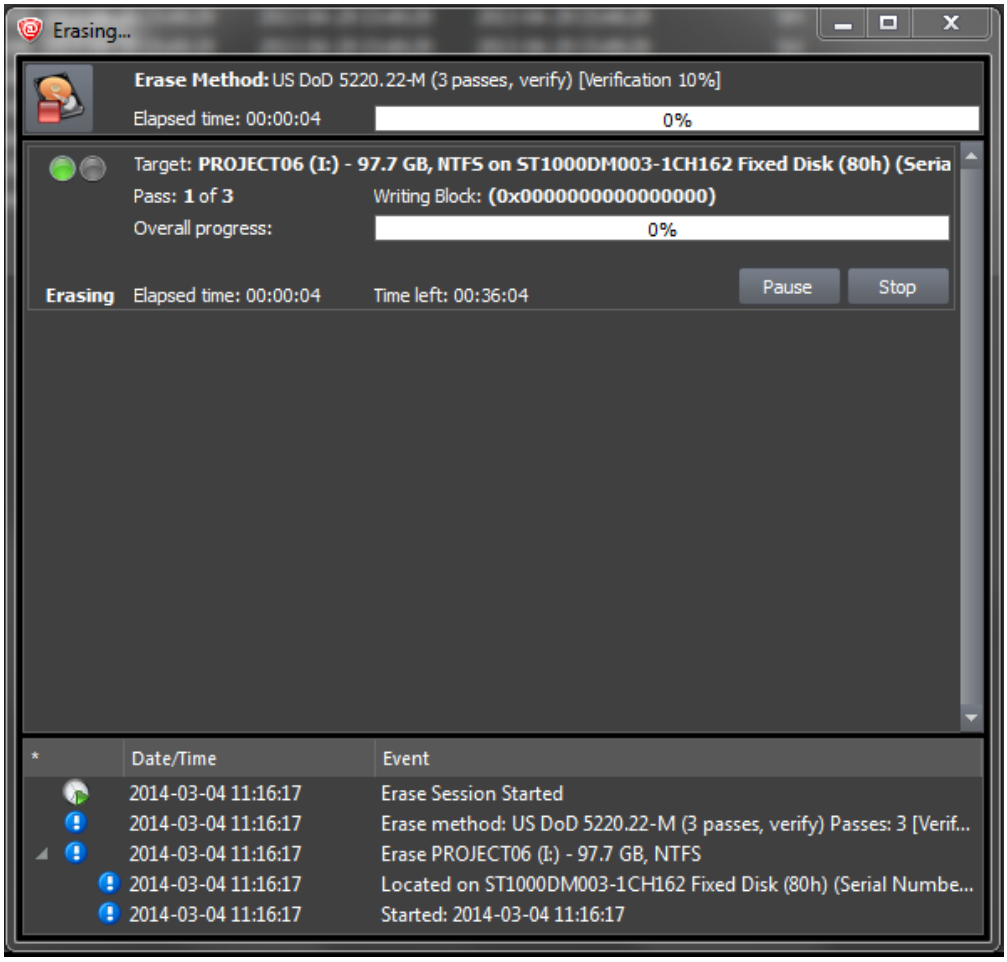
Select the 'Logo/Technician info' tab (below). This displays the Library's logo and some key metadata for inclusion in the deletion certificate which the software produces at the end of the process. You can edit the data on the right. If this is to be presented to a donor/depositor as evidence of secure deletion, you can enter their details as the 'Client' if necessary.

Ignore the 'SMTP' tab on the right, click on the 'Apply' button, and then click on 'Start' in the first box.

This will bring up a confirmation box requesting you to type in 'ERASE-ALL-DATA' (see below) before the erase process begins.



A box (below) then opens which shows you the progress of the erase process.



On completion, the Certificate will open to indicate that the process is successfully completed.

Step 1.30: Secure deletion of Workbench PC

The partition(s) you have been using for processing the archival files on the Workbench PC should be erased using Active@ KillDisk following exactly the same procedures as outlined for the Quarantine PC in Step 1.29. Ensure that you select ONLY the partition(s) you have been using.

Chapter 2: Archiving emails at individual email level

Email-level processing will usually take place after ingest at PST or Sequence level is complete, so the guidance in this chapter is based on the assumption that:

- PST file/s have been acquired from the creator;
- they have been transferred to the Library and verified;
- they have been subject to initial archival appraisal;
- a working copy of each PST file is available on the Workbench PC;
- the PSTs have been archived as digital objects along with related accession and collection objects;

- the Manchester eScholar PIDS for all PSTs, accession and collection objects are available.

Step 2.1: Email extraction and migration (Aid4Mail)

In this first step you will extract and migrate email messages, their metadata and file attachments. Email messages are migrated into four formats: .mht, .eml, .msg and .xml. Migrated email files and their attachments are generated in subfolders named with the Aid4Mail md5 checksum, one folder per email message. Descriptive metadata for all messages within each PST is saved to a single file named 'metadata.xml' stored in the email sequence folder.

1. Load Aid4Mail
2. In the Mail Source option select "Office Outlook PST file", click "Next>"
3. Browse for and select the PST file you wish to extract metadata and files from, click "Next >"
4. Ensure all required Source MAPI Folders are ticked (untick any you wish to exclude), e.g. Calendar, Tasks, Notes. Click "Next>"
5. Ignore Filter Options, click "Next >"
6. Select "Export email metadata to XML and extract emails to MSG (v1.6)", click "Next >"
7. In "Target Settings", browse (by clicking the ... next to the folder path) to the folder where you wish to extract the metadata and files to, this should be the PST email sequence folder for the relevant accession.

   <drive letter>:\<collection code>\<accession number>\<email sequence number>

   For example for the Carcanet Archive, accession 2012010, email sequence 001 the folder path would be

   <drive letter>:\cpa\2012010\001

   Click "Next>"

8. In "Start Processing Mail", click "Start>".
9. Make a note of any files Aid4Mail could not process and retain for collection file.
10. Exit Aid4Mail by clicking "Close".

Step 2.2: Clean XML metadata

Aid4Mail metadata is stored in XML format. This needs to be correctly encoded and valid before it can be ingested. Aid4Mail does not necessarily complete these steps fully for you. The problem arises because the XML metadata is being taken from a potentially non-XML-compatible source, i.e. a PST file. XML can contain any character-encoding, but in order to retain all possible characters, it should be at least encoded UTF-8. Certain UTF-8 characters are invalid XML, e.g. &x011; These are mainly control characters and have no visible rendering.

As a result, it is recommended that you run a separate 'clean XML' step by converting all the extracted XML files (messages and metadata) to UTF-8 and removing all invalid XML characters. This can take a LONG time.

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.

2. Change to relevant drive letter by typing <drive letter>: <Enter>

3. At the command prompt, type <drive letter>:\BAT-Scripts\cleanXMLfiles.bat \<collection code>\<accession number>\<email sequence number> <Enter> e.g.

   e:\BAT-Scripts\cleanXMLfiles.bat \cpa\2012011\001

4. Wait for the process to run. It will scroll one line per email processed. Once completed, additional files are created in each individual email folder. These additional files are:

   a. email_tmp.xml: this is a temporary file created as part of the processing, and is not ingested.

   b. email_utf8.xml: this is the final UTF-8-encoded valid XML for ingest.

   The original XML version of the email is not retained for ingest.

   Similarly, a further two XML files will also be created in the email sequence-level folder. These are:

   a. metadata_tmp.xml

   b. metadata_utf8.xml

Step 2.3: Extract technical metadata from email message files

Technical metadata for individual email messages (in all four formats – msg, eml, mht, xml) consists of:

- Filesize
- Mimetype
- MD5 checksum

This is extracted using Jacksum software to an XML file, from which it is exported to PREMIS. There is a script which iterates across all subfolders and runs Jacksum against any .msg, .eml, .mht, and .xml files. It creates a metadata file for each email format, taking the filename and extending it with .jacksum.xml, e.g. for the file email.mht in the folder \cpa\2012010\001\ffecd16a6d98985dcac22e73fd63ee94, Jacksum would create a file called: email.mht.jacksum.xml in the same folder.

Run Jacksum on message files as follows (NB: this takes a LONG time):

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.

2. Change to relevant drive letter by typing <drive letter>: <Enter>

3. At the command prompt, type <drive letter>:\BAT-Scripts\createJacksums.bat <collection code> \<collection code>\<accession number>\<email sequence number> <Enter> e.g.

   e:\BAT-Scripts\createJacksums.bat cpa \cpa\2012011\001

NB: ensure that you enter the accession AND sequence folder names correctly, otherwise the script will create checksums against ALL files in ALL subfolders. If you realise you have made a mistake, you can abort the script by entering CTRL-c.

Step 2.4: Extract technical metadata from file attachments

The FITS tool is run against all attachments in order to verify their format and extract technical metadata. The FITS for attachments is stored in full, inside the PREMIS record for each attachment.

Create FITS for every attachment by doing the following (NB this takes a VERY LONG time).

2.4.1 Create batch file for running FITS against each file attachment

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to relevant drive letter by typing <drive letter>: <Enter>
3. At the command prompt, enter:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:”<drive letter>:\<collection code>\<accession number>\<email sequence number>\metadata_utf8.xml” -xsl:”<drive letter>:\<collection code>\XSLT-Transforms\CreateFITSScriptForFileAttachments.xsl” -o:”<drive letter>:\<collection code>\<accession number>\<email sequence number>\createFitsFileAttachments.bat” drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>
4. Close down the command window by typing Exit.
5. Check that the batch file has been successfully created in the relevant email sequence folder. Open it in Notepad and verify that the correct FITS commands have been entered. It should look like this example:

```
call fits.bat -i "e:\cpa\2013032\001259eb87d2f26167de2ea8b54c573ac95\all\SKMBT_C25312083110140.pdf" -o
"e:\cpa\2013032\001259eb87d2f26167de2ea8b54c573ac95\all\SKMBT_C25312083110140.pdf.fits.xml"
echo created e:\cpa\2013032\001259eb87d2f26167de2ea8b54c573ac95\all\SKMBT_C25312083110140.pdf.fits.xml"
call fits.bat -i "e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\Author questionnaire The Gypsy and the Poet.doc" -o
"e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\Author questionnaire The Gypsy and the Poet.doc.fits.xml"
echo created e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\Author questionnaire The Gypsy and the
Poet.doc.fits.xml"
call fits.bat -i "e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\The Gypsy and the Poet Final Version 2.doc" -o
"e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\The Gypsy and the Poet Final Version 2.doc.fits.xml"
echo created e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\The Gypsy and the Poet Final Version 2.doc.fits.xml"
call fits.bat -i "e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\DavidM3-smuse.jpg" -o
"e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\DavidM3-smuse.jpg.fits.xml"
echo created e:\cpa\2013032\0018aaced59d81df9d39f8f43024b9c7fca\all\DavidM3-smuse.jpg.fits.xml"
```

2.4.2 Run the batch file to produce FITS output for each attachment

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. The command must be run from the FITS folder, which is on the c: drive. Change to this drive by typing c: <Enter>
3. Change working folder by typing cd \fits <Enter>
4. At the command prompt, enter:

   <drive letter>:\<collection code>\<accession number>\<email sequence number>\createFitsFileAttachments.bat

e.g.

e:\cpa\2013032\001\createFitsFileAttachments.bat

5. Once the process has finished running (it takes a LONG time), check a random sample of email message folders. Where messages have attachments, there will be an additional subfolder within each message folder named “all”. All the attachments are saved inside this subfolder; alongside each attachment should now also be a FITS XML output, based on the attachment file title and mimetype but with the suffix .fits.xml, e.g.

   \cpa\2013032\001\f2918acc03508657531d2405b0276af3\all\Deane New and Selected Poems style sheet.doc
   cpa\2013032\001\f2918acc03508657531d2405b0276af3\all\Deane New and Selected Poems style sheet.doc.fits.xml
   \cpa\2013032\001\efea319a174d854607a486d0756279f1\SlowlyAsIfPROOF2.pdf
   \cpa\2013032\001\efea319a174d854607a486d0756279f1\SlowlyAsIfPROOF2.pdf.fits.xml

---

Step 2.5: Create persistent identifiers (PIDs) and descriptive metadata for folder and email objects

Unique identifiers must be created for every digital object that will be ingested. A PID is made up of two parts: a namespace (e.g. uk-ac-man-emm); and an alphanumerical set of characters, separated from the namespace by a colon. The namespaces used for identifying email-related objects are:

- uk-ac-man-col: a collection object.
- uk-ac-man-ema: an accession object.
- uk-ac-man-ems: an email sequence object.
- uk-ac-man-emf: an email folder object.
- uk-ac-man-emm: an email message object.
- uk-ac-man-emh: an email attachment indexed document (or, if in the future, attachments are separated from their emails, an email attachment object).

The alphanumeric sets of characters following these namespaces are unique WITHIN that namespace. For example, for an accession of the Carcanet Archive, the PIDs would take the following form:

- uk-ac-man-col:cpa
- uk-ac-man-ema:2013032
- uk-ac-man-ems:2013032001
- uk-ac-man-emf:2013032001f2000
- uk-ac-man-emm:2013032001e2000
- uk-ac-man-emh:FILE_2013032001e2000_1

PIDs of the above form are created from the metadata extracted by Aid4Mail in Step 2.1 and stored in the files emailpids.xml and folderpids.xml.

Step 2.5.1: Create PIDs and descriptive metadata for email folder/ subfolder objects

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.

2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\metadata_utf8.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateFolderPIDS.xsl" -o:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\folderpids.xml" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create a single XML document named folderpids.xml for each email sequence (e.g. \cpa\2013032\001\folderpids.xml), which contains folder PIDs and descriptive metadata. Check that this file has been successfully created.

Step 2.5.2: Create PIDS for email objects and file attachments

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.

2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\metadata_utf8.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailPIDS.xsl" -o:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create a single XML document named emailpids.xml for each email sequence (e.g. \cpa\2013032\001\emailpids.xml), which contains email and file attachment PIDs. Check that this file has been successfully created.

Step 2.6: Create datastreams

NOTE: There are two alternative methods for carrying out Steps 2.6 and 2.7. One of these is run from the command line using Java, and the other involves using Oxygen XML Editor. Certain steps can ONLY be run from the command line if a licenced version of Saxon (Saxon-PE or Saxon-EE) is available. If you do not have a licenced version of Saxon then Oxygen XML Editor MUST be used for these steps; it is optional for the other steps. Steps that require a licenced version of Saxon are indicated below.

The instructions set out in Steps 2.6.1-4 and 2.7.1-2 below are based on using the command line/Java. See the Addendum at the end of Step 2.7 (p. 89) for alternative instructions based on using Oxygen. These are generic instructions which can be slightly modified to carry out each step in the same way.

Step 2.6 creates the metadata datastreams for each email digital object, which consist of:

- Descriptive metadata.

- Preservation metadata (in PREMIS) for message files.
- Preservation metadata and event log (in PREMIS) for file attachments.
- Event log for message files.

Each of the following transformations relies on the existence of the emailpids.xml as created in step 2.5.2. The transformations check for the existence of the appropriate support files. In particular it checks for the file attachment ensuring it can resolve the file name correctly. Situations can arise where the character encoding of the file name prevents the transformation accessing the file. As a result no datastream of preservation metadata and event log for the file attachment is created.

Step 2.6.1: Create descriptive metadata for email message files

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>
3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:”<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml” -xsl:”<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailA4MDatastream.xsl” drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create an XML file called a4m.xml for each email object. The file is stored in the relevant subfolder named with the MD5 checksum. Check a selection of subfolders to see that this file has been successfully created in each one.

Step 2.6.2: Create preservation metadata for email message files

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>
3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:”<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml” -xsl:”<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailPREMISMessageFilesDatastream.xsl” drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create an XML file called premis-representationfile.xml for each email object. The file is stored in the relevant subfolder named with the MD5 checksum. Check a selection of subfolders to see that this file has been successfully created in each one.

Step 2.6.3: Create preservation metadata and event log for file attachments

1. Run cmd (by typing “cmd” into Windows “Start” menu search box) and click <Enter>.
2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailPREMISAttachedFilesDatastream.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create an XML file for each email object, which is named with the prefix FILE_ followed by the email PID and _1, _2, _n, if there is more than one attachment, followed by –premis-file.xml. For example, FILE_2012010001e51372_1-premis-file.xml. This file is stored in the "all" subfolder within the message folder named with the MD5 checksum. Check a selection of subfolders (make sure you select messages with attachments) to see that this file has been successfully created in each one.

Step 2.6.4: Create event log for email message files

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.

2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

   java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailPREMISEventLogDatastream.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create an XML file called premis-eventlog.xml for each email object. The file is stored in the relevant subfolder named with the MD5 checksum. Check a selection of subfolders to see that this file has been successfully created in each one.

Step 2.7: Transform and package

FOXML files need to be created prior to ingest. These wrap the separate datastream XML files produced in Step 2.6, and the file attachments extracted in Step 2.1, in an XML container which is suitable for ingest into Fedora. A FOXML file is created for each folder PID and email PID as created in Steps 2.5.1 and 2.5.2 respectively. These form the digital objects stored in Fedora.

There are two stages: Step 2.7.1 creates email folder FOXML files and Step 2.7.2 creates email message FOXML files. To facilitate ingest of large numbers of digital objects it is best to split up the collection into batches of 20,000 or fewer FOXML files. This is especially the case with email message FOXML files because of the large number of messages that can exist in a single PST file.

Step 2.7.1 Create email folder FOXML files

If a licenced version of Saxon is unavailable, it is necessary to use the Oxygen XML editor to perform this transformation as described in the Addendum at the end of Step 2.7.

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.

2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

    a. java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\folderpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateFolderFOXML.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

4. This command will create a FOXML XML file called <pid>.xml for each email folder. The file is stored in a subfolder named 'foxml' under the email sequence folder where the pst and folderpids.xml are located. Check a selection of FOXML files to see that they have been successfully created, one for each folder identified in the file folderpids.xml.

Step 2.7.2 Create email message FOXML files

If a licenced version of Saxon is unavailable, it is necessary to use the Oxygen XML editor to perform this transformation as described in the Addendum at the end of Step 2.7.

1. Run cmd (by typing "cmd" into Windows "Start" menu search box) and click <Enter>.

2. Change to the relevant drive by typing <drive letter>: <Enter>

3. Type in the following command:

    java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailMessageFOXML.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

    OR

    java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"<drive letter>:\<collection code>\<accession number>\<email sequence number>\emailpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateEmailMessageFOXML.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> start=<batch start record number> end=<batch end record number><ENTER>

    The first version of these commands will create a FOXML for every email PID identified in the file emailpids.xml.

    The second version will create a subset of FOXML files according to the numbers entered into the start and end parameters. For example, the following command would create 1,000 FOXMLs from record 50 to 1050 from the emails PIDS found in the file e:\cpa\2013032\001\emailpids.xml. Records are sorted by the value of their MD5 folder name to help match up batches of FOXML files with the datastreams and file attachments stored in the MD5 subfolders.

    java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"e:\cpa\2013032\001\emailpids.xml" -xsl:"e:\cpa\XSLT-Transforms\CreateEmailMessageFOXML.xsl" drive=e collection=cpa accession=2013032 emailsequence=001 start=50 end=1050<ENTER>

If the end parameter is not entered then the transformation will create a subset of FOXML from the start record number to the last record. For example the following command will create a FOXML file for the last 158 records identified in an emailpids.xml containing 7158 records.

java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -s:"e:\cpa\2013032\001\emailpids.xml" -xsl:"e:\cpa\XSLT-Transforms\CreateEmailMessageFOXML.xsl" drive=e collection=cpa accession=2013032 emailsequence=001 start=7000<ENTER>

4. This command will create a FOXML XML file called <pid>.xml for each email message. The files are stored in a subfolder named 'foxml<start>-<end>' under the email sequence folder where the pst and emailpids.xml are located. Check a selection of FOML files to see that they have been successfully created, one for each email message identified in the file emailpids.xml. NOTE that records are sorted by the value of their MD5 folder name to help match up batches of FOXML files with the datastreams and file attachments stored in the MD5 subfolders. They are NOT sorted by their PIDs, so do not expect to see a running series of PID numbers from 1-xx.

Addendum: Running transformations for Steps 2.6-2.7 using Oxygen XML Editor

Oxygen XML Editor provides an alternative mechanism to perform the transformations outlined above in Steps 2.6 and 2.7. This may be needed if a licenced version of Saxon is unavailable. The following instructions are generic and may be applied to any of these transformations. Screenshots are from Oxygen XML Editor v14.2.

1. Load Oxygen XML Editor (normally available from the Start menu under "All Programs" -> "Oxygen XML Editor v14.2"; choose the 64-bit version where this is available).

2. Select from the menu bar "File" -> "Open" (or enter the characters CTRL-o).

3. Browse to the folder <drive letter>:\<collection code>\XSLT-Transforms\, select the file "CreateEmailA4MDatastream.xsl" and click "Open". You may choose any of the following xsl files depending on which step you wish to complete.

    - for step 2.6.1 open "CreateEmailA4MDatastream.xsl",
    - for step 2.6.2 open "CreateEmailPREMISMessageFilesDatastream.xsl",
    - for step 2.6.3 open "CreateEmailPREMISAttachedFilesDatastream.xsl"
    - for step 2.6.4 open "CreateEmailPREMISEventLogDatastream.xsl"
    - for step 2.7.1 open "CreateFolderFOXML.xsl"
    - for step 2.7.2 open "CreateEmailMessageFOXML.xsl"

4. Edit the xsl parameters to match the corresponding <drive letter>, <collection code>, <accession number> and <email sequence number> (NB: note the use of single quotes wrapped in double quotes. Be careful to retain these characters.)

```
<xsl:param name="drive" select="'<drive letter>'"/>
<xsl:param name="collection" select="'<collection code>'"/>
<xsl:param name="accession" select="'<accession number>'"/>
<xsl:param name="emailsequence" select="'<email sequence number>'"/>
```

If you are executing the CreateEmailMessageFOXML.xsl transformation (see step 2.7.2), the following parameters are also included in the xsl.

```
<xsl:param name="start" select="'<record start number>'"/>
<xsl:param name="end" select="'<record end number>'"/>
```

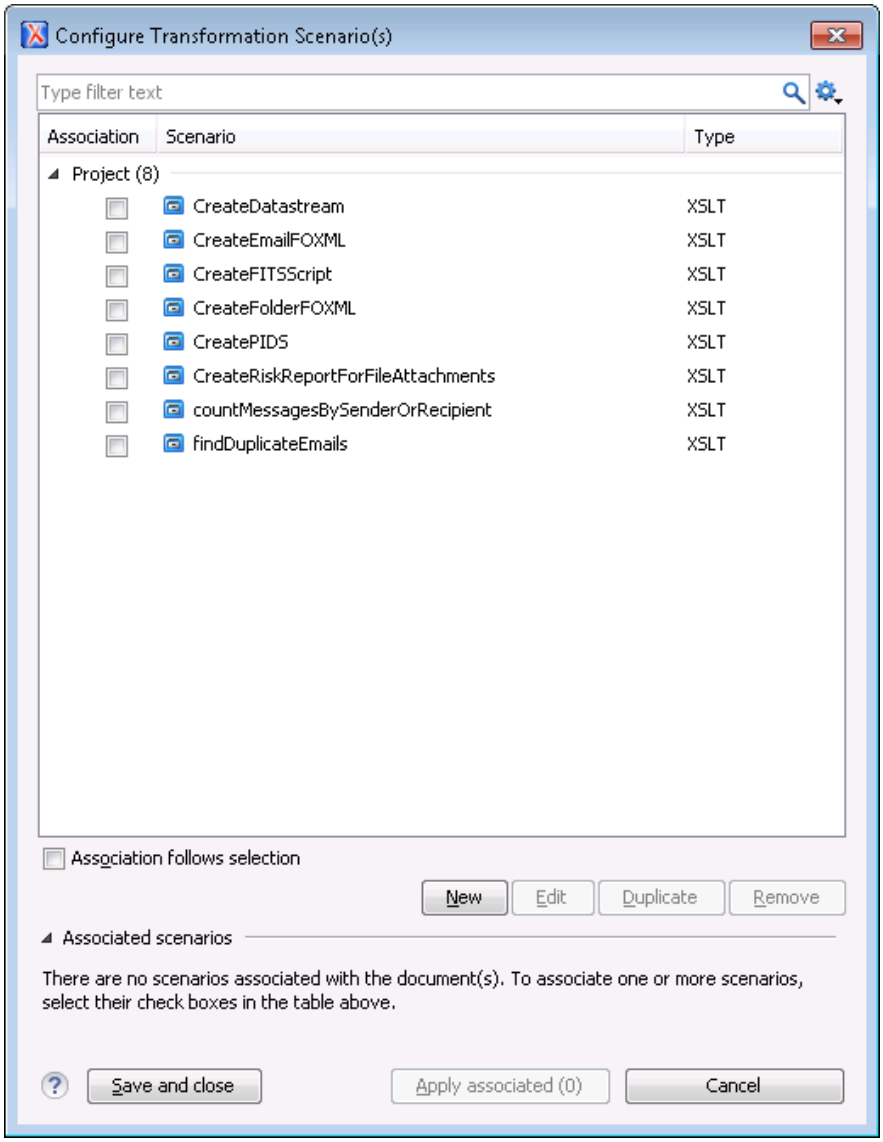
You only need to enter anything into these parameters if you are splitting a sequence of emails in order to ingest in batches. Entering a start number of '1' and an end number of '20,000' will result in a subset of FOXML records from record 1 to 20,000. If no end parameter is entered, the command will run to the end of the sequence. See Step 2.7.2 for an explanation of how the batching and sorting system works.

For example, on the e: drive for the email sequence cpa/2012010/001 (which is not being split up for ingest in batches), the parameters would look like the following:

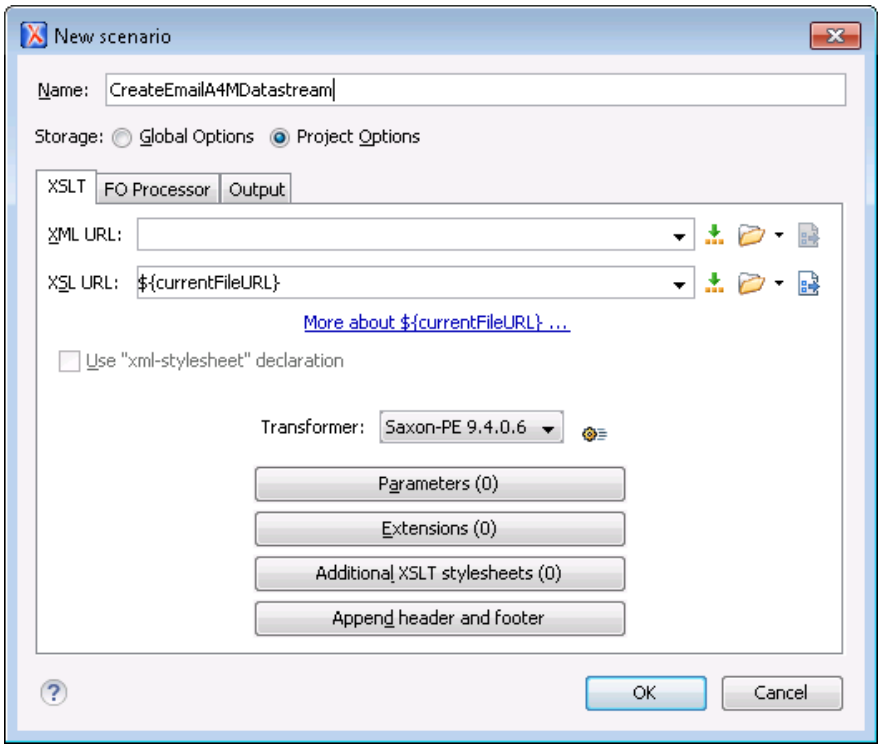
```
<xsl:param name="drive" select="'e'"/>
<xsl:param name="collection" select="'cpa'"/>
<xsl:param name="accession" select="'2012010'"/>
<xsl:param name="emailsequence" select="'001'"/>
```

5. Save the edited xsl by selecting "File" -> "Save" (or enter the characters CTRL-s). It is safe to overwrite the existing xsl file.

6. Next you need to configure the transformation scenario, from the menu bar select "Document" -> "Transformation" -> "Configure transformation scenario(s)…" (or enter the characters CTRL+SHIFT+c) This should open the following dialog (NB: this dialog shows some preconfigured transformation scenarios; if Oxygen is newly installed this list may be blank).

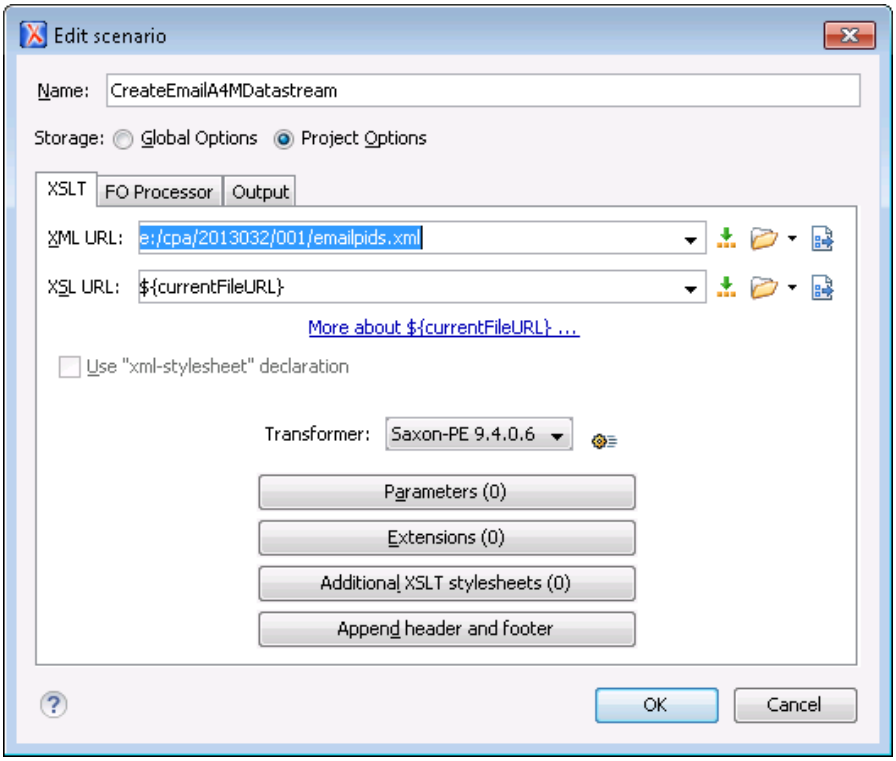


7. Click “New” and select “XSLT Transformation”. This should open the following dialog.

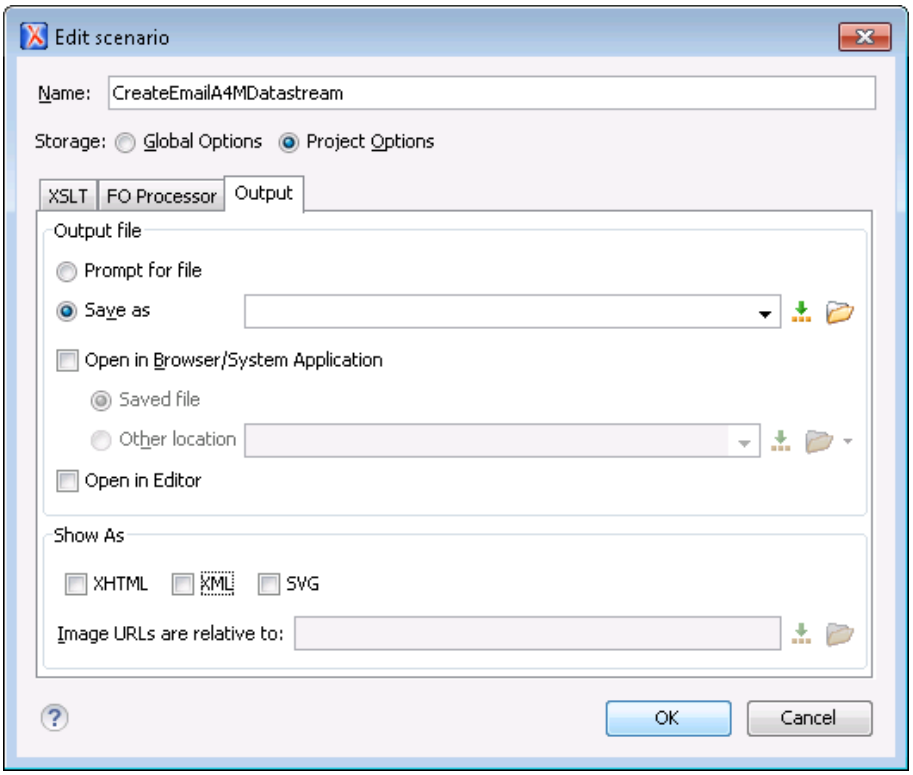


8. In the Name field enter a name for the transformation scenario, the default is the name of the transformation file without the extension; you may keep this or create a new name.

9. In the XML URL field enter the path to the emailpids.xml file (NB: if you are running step 2.7.1 you should enter the path to the folderpids.xml instead). For example, for the 2013032 001 email sequence this would be e:/cpa/2013032/001/emailpids.xml (and for step 2.7.1 e:/cpa/2013032/001/folderpids.xml). You may type this into the field or use the folder icon to browse for the file giving a dialog like the one below.



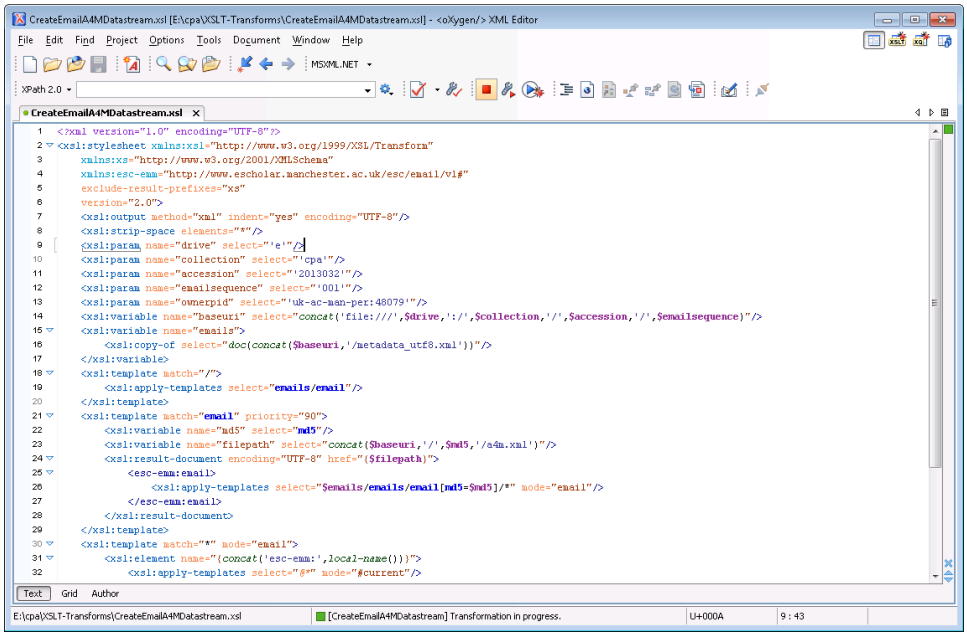
10. Select the Output tab and untick the “Show As” -> XML checkbox. The Output tab should look like the below. Click “OK” to close this dialog and return to the “Configure Transformation Scenario(s)” dialog.



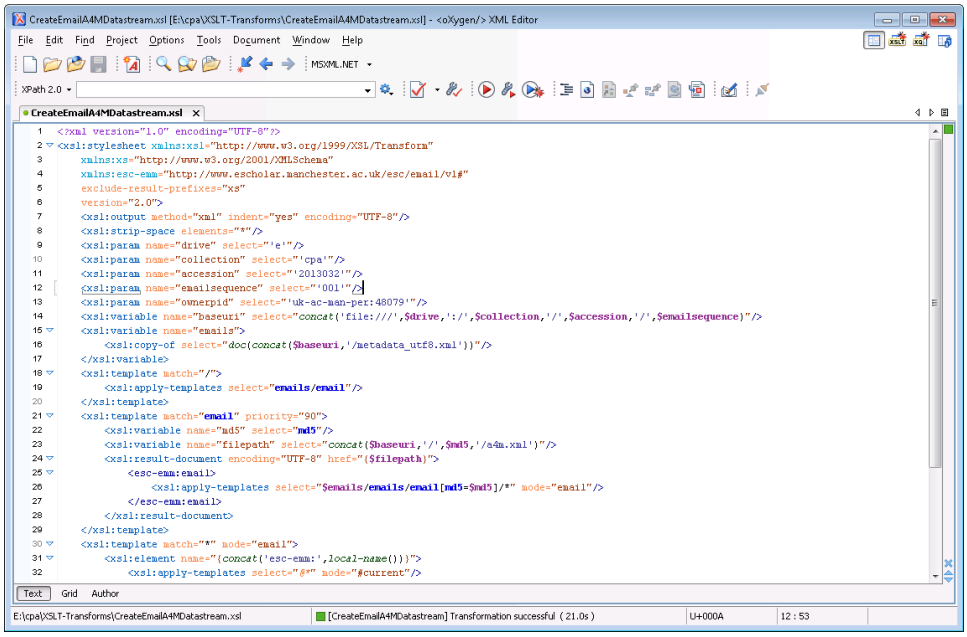
11. Once you have entered your scenario settings click “Save and close” from the “Configure Transformation Scenario(s)” dialog.

12. You are now ready to execute your transformation scenario. From the menu bar Select “Document” -> “Transformation” -> “Apply Transformation Scenario(s)” (or enter the characters CTRL+SHIFT+t).

    Depending on the transformation being executed this may take a number of minutes to finish. While the transformation is running Oxygen will display the message “Transformation in progress” in the status bar. An example is below.



You may do other work while the transformation is running but you should NOT close down the Oxygen XML Editor. Once the transformation is complete Oxygen will display in its status bar “Transformation successful” and the time it took to complete. An example is below.



13. This completes the transformation. You may close down Oxygen or go back to step 1 and repeat these steps for another transformation or another email sequence. NB: you may reuse the same transformation scenario by changing its settings accordingly (select “Edit” rather than “New” in the “Configure Transformation Scenario(s)” box); remember always to change the xsl parameters and align these with the location of the emailpids.xml or folderpids.xml file.

14. Once you have completed one or more transformations you should check the relevant XML files have been created as outlined in steps 2.6.1-4 and 2.7.1-2.

Step 2.8: Ingest folder and email message digital objects

The following steps outline how to ingest the FOXML files and datastreams created in previous steps. It is possible to ingest FOXML files into a Fedora repository via a number of routes. These are fully documented on the Fedora Commons website at https://wiki.duraspace.org/display/FF/Documentation.

The following steps assume that all FOXML files, all datastreams and all file attachments have been extracted and generated for one or more entire PST files. The ingest steps will process many thousands of FOXML files; however, this can take some time. From experience the average time to ingest a single email message FOXML is approximately 8.5 seconds. For a collection of 20,000 email messages it would take around 47 hours 15 minutes. The ingest will run unattended; however, the longer a batch takes, the greater the risk of it being interrupted unexpectedly, e.g. by a network outage, hardware failure or software failure. Recovering from an interruption involves identifying which records remain to be ingested and this can take some effort.

To reduce the risk of unexpected interruptions, it is wise to avoid very large batch ingests. Batch sizes of no more than 20,000 records are sensible. Step 2.7.2 describes how to create a batch of FOXML files.

To execute an ingest, all managed datastream files (e.g. in the case of an email message object, these are the four formatted email message files, the metadata and any file attachments) need to be first uploaded to the Fedora enabled server on some temporary file space; doing this makes the files available to Fedora. All files must retain the same folder structure and filenames. On Manchester eScholar the temporary file space is limited to <10GB which limits the size of a batch ingest. All files must be removed from this temporary space once the batch ingest is complete.

Uploading of files is required whenever a FOXML file references a managed datastream. This applies to collection, accession, email sequence and email message digital objects. It does not apply to email folder digital objects as these use inline datastreams (i.e. all relevant metadata is embedded in the FOXML file).

Step 2.8.1 describes the upload of files to the Fedora enabled server. Step 2.8.2 describes the ingest of FOXML files using the Fedora client software. **Please note, both these steps require system administrator privileges on the Fedora server and should ONLY be done by technical experts.**

Step 2.8.1 Upload of files to the Fedora enabled server

1. It is likely that you will need to upload very many folders and files to the Fedora enabled server before the ingest can proceed. This is best done by first compressing the folders and files into a single compressed archive. You can do this using 7zip or some other compression software. Name the compressed file with the accession number, email sequence number and the range of records; e.g. 2012010004_1-20000.7z would contain records 1 to 20,000 of email sequence 001 in accession 2012010. You should ONLY compress the MD5 folders relevant to the batch ingest being considered. The nth FOXML created in a batch corresponds to the nth MD5 folder sorted in ascending order by name. For example, if you created a batch of FOXMLs from record 501 to 1000 you would need to compress the MD5 folders from the 501st to 1000th when sorted in ascending order by name.

2. To upload the compressed archive you need to use SFTP client software such as FileZilla or WinSCP (NB: You should check that there is enough available temporary disk space to store both the compressed archive file and all uncompressed selected MD5 folders and files). Connect the Fedora production server and locate the file space used to store temporary uploaded files. This folder name should correspond to that referenced in the FOXML files in the managed datastream XMLs. Create a folder in that file space named <accession number><email sequence number>. For example, the 2012010 accession and email sequence 001 would have a folder named 2012010001. Copy the compressed archive to this folder.

3. Once the compressed archive has been copied to the temporary file space on Fedora, uncompress it. On Manchester eScholar you can use the '7z' command (available by logging in using Putty). For example, to uncompress the zip file 2012010004_1-20000.7z, change your working folder to the folder you uploaded the file to and type the command '7z x 2012010004_1-20000.7z'. Uncompressing an archive can take some time.

4. Once all folders and files have been uncompressed you are in a position to run the ingest as described in Step 2.8.2.

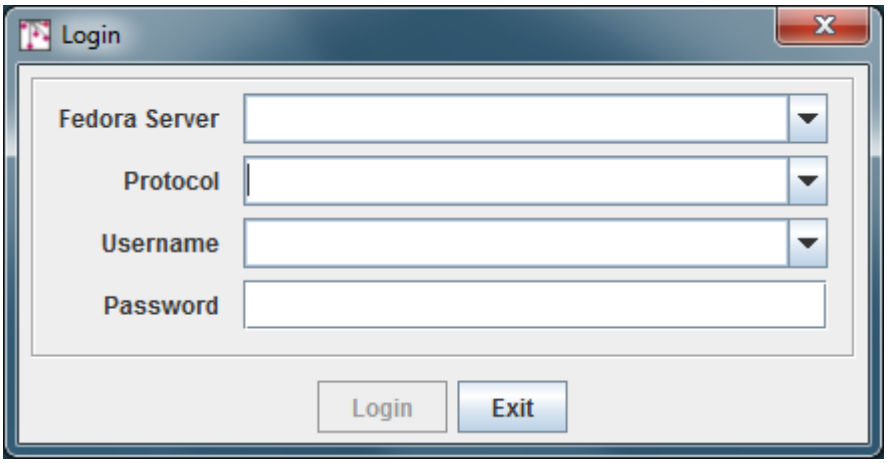
5. Once the ingest is complete you should remove all uncompressed files and the compressed archive from the server. IT IS VERY IMPORTANT THIS IS DONE VERY CAREFULLY. To do this change your working folder to the parent of the folder that holds the uncompressed files. Type the command '\rm –rf <folder name><Enter>'. For example to remove all files and subfolders in the folder 2012010001, type the command

   \rm –rf 2012010001<Enter>

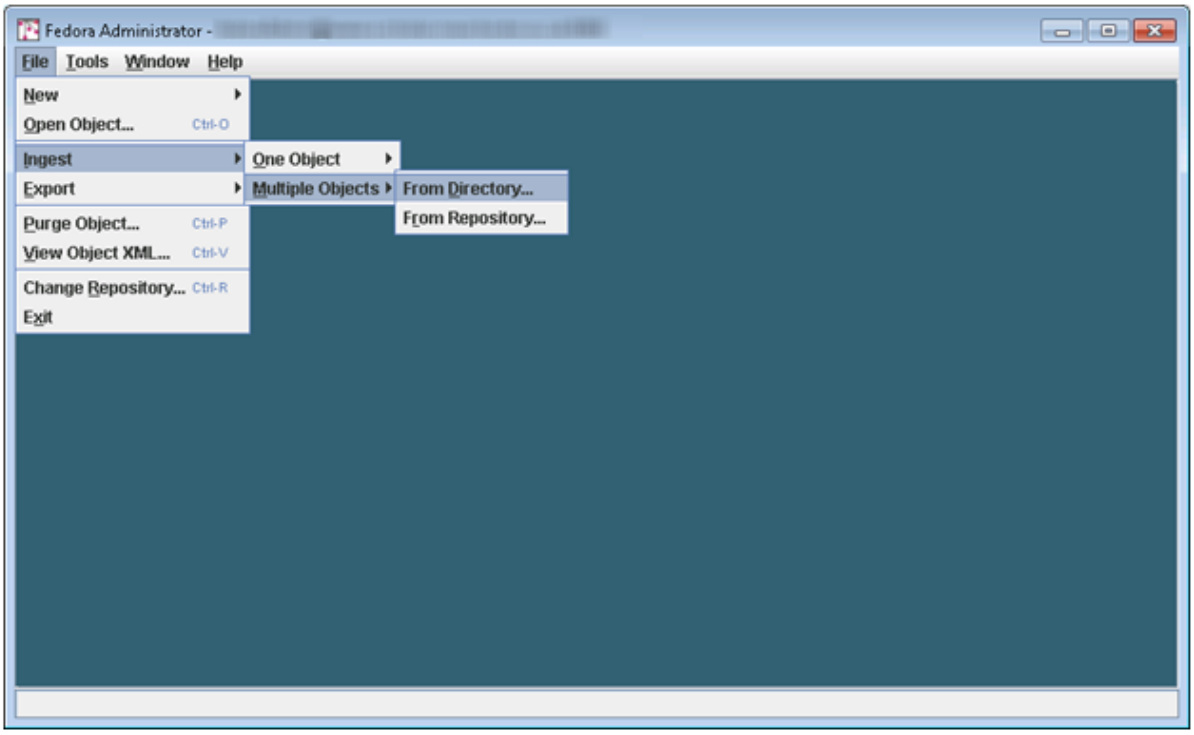
Step 2.8.2 Ingest of FOXML files using the Fedora client software

1. Execute the Fedora client software by running the fedora-admin.bat file; depending on how Fedora was installed this command can be found at <fedora installation folder>\client\bin\fedora-admin.bat> (for example on the Workbench PC the fedora client software is located at C:\fedora\client\bin\fedora-admin.bat); browse to the correct folder and double click the fedora-admin.bat file (alternatively create a shortcut on the desktop or task bar to this file and double click one of those).

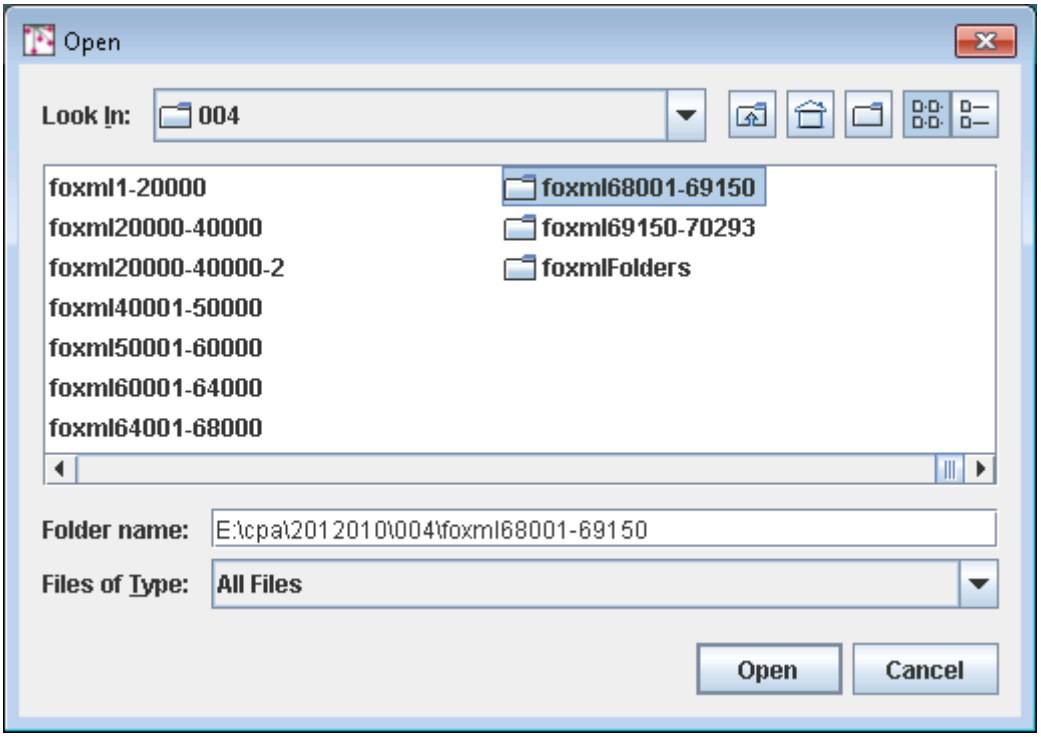
2. Once the Fedora client software is running you are prompted for the repository location and user credentials to make a connection. Below is a screenshot of the dialog where you enter these details. These details need to be provided by a Fedora system administrator and SHOULD NOT be distributed. Enter the details and click <Login>



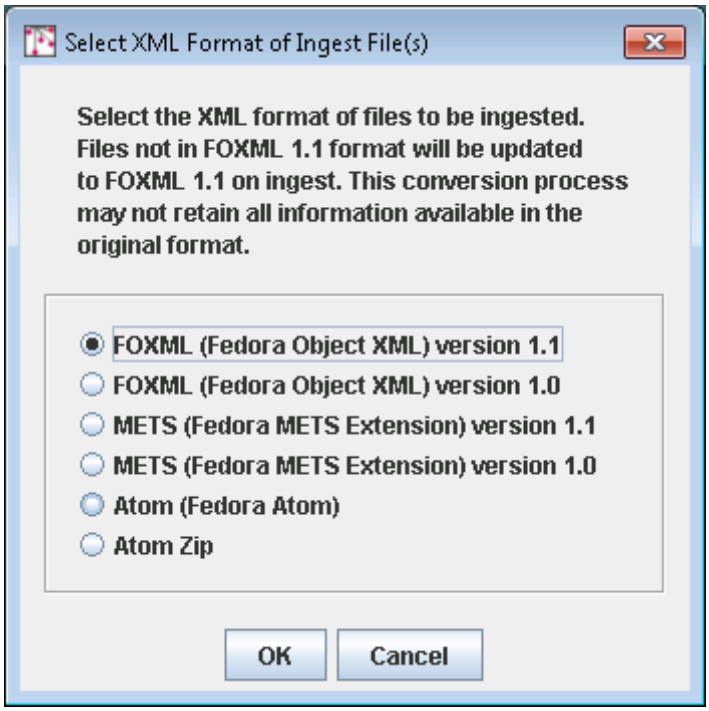
3. Once you are connected to the Fedora repository, select from the menu bar 'File' -> 'Ingest' -> 'Multiple Objects' -> 'From Directory'. This is illustrated below.



4. Browse to the folder that contains the FOXML files that you wish to ingest, select the folder name and then click <Open>. Below is a screenshot of the browse dialog.



5. You are then asked to confirm the type of FOXML file that you wish to ingest. For all types of objects generated so far select the default type 'FOXML (Fedora Object XML) version 1.1' and then click <OK>. See the screenshot below for this dialog.



6. The ingest will start. A progress bar will flicker in the status bar. You should leave the Fedora client software running and should not log-off or shutdown the computer. You can lock the computer using the WIN+L key combination. Alternatively, if you are using remote desktop just close down the connection without logging off.

7. Once the ingest has started it may take some time to complete. The Fedora client software doesn't give any information on progress. As a result, it may be worth running a SOLR query occasionally to review progress of the ingest. The type of SOLR query depends on the types of objects being ingested. Further instructions on running SOLR queries can be found in Section 2.9.

8. Once the ingest has completed, the Fedora client software will give a summary of the number of successfully ingested objects, the number of failed objects and the time taken. The software also produces a detailed log of ingest operations. This log can be used to double-check the ingest and tidy up any issues arising (see Step 2.8.3). Save this data in a text file for later examination.

Step 2.8.3 Audit and tidy up failed ingests

The purpose of the audit is to check that ingests have completed successfully. Ingests can fail for two main reasons:

- issues with the names of file attachments, e.g. they contain odd characters like double spaces or non-Latin characters; and
- failure to index the digital object, i.e. the object has ingested successfully but hasn't been indexed, so it is unfindable.

The audit process has these main elements:

- Identify target metric values, i.e. the expected number of folder objects, email objects and attachment datastreams.

- Identify ingested and indexed objects by analysing ingest logs AND running SOLR queries.

- Compare these actual metrics with the expected ones (e.g. how many email message objects have been ingested and indexed) and identify where differences exist, e.g. the PIDs of email objects that have not been ingested and indexed.

- 'Touch' the index for identified objects then repeat step 2.

- Tidy up any uningested objects: locate and check them; and make a decision about whether to reingest them; this might involve changing filenames, rerunning FITS, rerunning JACKSUM and regenerating FOXML, so is a labour-intensive process.

NOTE: Elements of Steps 2.8.3.1-3 can be automated. Steps 1-3 below outline the manual processes. See Step 2.8.3.4 for how to automate some of these processes.

Step 2.8.3.1 Identify target metric values

Suitable metrics for analysis are as follows:

Number of email folder digital objects – derived from the folderpids.xml file, the number of email folder FOXML files generated, and the email sequence EAD (although note that these metrics may differ because of the presence of non-email folders such as calendar, tasks and notes).

To do this manually (see section 2.8.3.4 for an automated way of doing this):

1) From folderpids.xml:

    a) Open folderpids.xml in Oxygen.

b) Find all root folders (e.g. Inbox, Sent Items, there may be others; these can be identified because they do not contain a slash, and hence are NOT a subfolder of anything else).

c) Write down the number of subfolders, messages and attachments in each root folder, add these up.

2) From email folder FOXML files:

a) Go to the email sequence folder and find the folder where the foxmls are stored (NB: these may be spread across multiple folders because of the batch processing).

b) Enter the keys CTRL-a; this will highlight all files; record the count. Where there is a mix of email folder and email message foxml files, you need to highlight just the folder ones by ordering the files by name, selecting the first folder foxml (e.g. uk-ac-man-emf-2013032001f1.xml), keeping the SHIFT key down, scrolling down to the last folder foxml and selecting that. If the folder ONLY contains folder foxml files then right-click on the folder name and choose properties, this will display the number of subfolders and files in the folder.

3) From the email sequence EAD:

a) Look at the two <extent> elements, one of which contains numbers as calculated by PST Reporter, and the other as calculated by Aid4Mail; these elements record numbers of folders, messages or 'items', and attachments. These are not always the same, because of the way that entities like 'Calendar', 'Contacts' etc are counted.

b) Next, look at the 'List of folders and folder information' section within <scopecontent>. This lists the number of folders identified, and any folders which were identified by one of the two tools but not the other. Examining these usually reveals the source of any discrepancies (e.g. one tool counting 'Calendar' and 'Tasks' as folders when the other doesn't).

c) A combination of the two points above gives a relatively accurate overall estimate of expected folder numbers.

Number of email message digital objects – derived from the folderpids.xml file, the number of email message FOXML files, the number of MD5 folder names in the sequence and the email sequence EAD.

To do this manually:

1) From folderpids.xml:

a) Open folderpids.xml in Oxygen.

b) Find all root folders (e.g. Inbox, Sent Items, there may be others; these can be identified because they do not contain a slash, and hence are NOT a subfolder of anything else).

c) Write down the number of subfolders, messages and attachments in each root folder, add these up.

2) From the email message FOXML files:

a) Do the same as Step 2 above but highlight only email message FOXML files (e.g. uk-ac-man-emm-2013032001e1.xml).

3) From the email sequence EAD:

a) Look at the two <extent> elements, one of which contains numbers as calculated by PST Reporter, and the other as calculated by Aid4Mail; these elements record numbers of folders, messages or 'items', and attachments. These are not always the same, because of the way that entities like 'Calendar', 'Contacts' etc are counted.

b) If the number of messages/items differs, record both, as this should give a rough overall figure.

Number of email file attachment datastreams – derived from the email sequence EAD and the folderpids.xml. NB: this is the most inaccurate of the metrics.

1) From folderpids.xml:

a) Open folderpids.xml in Oxygen.

b) Find all root folders (e.g. Inbox, Sent Items, there may be others; these can be identified because they do not contain a slash, and hence are NOT a subfolder of anything else).

c) Write down the number of subfolders, messages and attachments in each root folder, and add these up.

2) From the email sequence EAD:

a) Look at the two <extent> elements, one of which contains numbers as calculated by PST Reporter, and the other as calculated by Aid4Mail; these elements record numbers of folders, messages or 'items', and attachments. These are not always the same, because of the way that entities like 'Calendar', 'Contacts' etc are counted. Attachment numbers also quite frequently differ.

b) If the number of attachments differ, record both figures, as this should give a rough overall figure.

Total number of digital objects – derived from adding up the figures above and from the email sequence EAD .

Step 2.8.3.2 Identify ingested and indexed objects

To do this manually (this is optional):

1) Run the following queries in SOLR:

a) Access SOLR by going to the URL for the local SOLR index (you MUST be on the University network or using the VPN service for this to work)

b) Execute the following SOLR searches to determine the number of the ingested objects.

To determine the total numbers of folders, messages and attachments across all sequences, use the following:

Number of email folder objects in all email sequences

r.isderivationof.pid:uk-ac-man-emf\:base

Number of email message objects in all email sequences

r.isderivationof.pid:uk-ac-man-emm\:base

Number of email attached files in all email sequences

r.isderivationof.pid:uk-ac-man-emh\:base

Total number of digital objects in entire collection

r.ismemberof.pid:uk-ac-man-col\:<collection code>

To determine the number of folders, messages and attachments in a single accession, use the following:

Number of email folder objects in email accession 2012010

r.isderivationof.pid:uk-ac-man-emf\:base AND r.ispartof.pid: uk-ac-man-ems\:2012010*

Number of email message objects in email accession 2012010

r.isderivationof.pid:uk-ac-man-emm\:base AND r.ispartof.pid: uk-ac-man-ems\:2012010*

Number of email attached files in email accession 2012010

r.isderivationof.pid:uk-ac-man-emh\:base AND r.ispartof.pid: uk-ac-man-ems\:2012010*

To determine the number of folders, messages and attachments in a single email sequence, use the following:

Number of email folder objects in (e.g.) email sequence 2012010/001

r.isderivationof.pid:uk-ac-man-emf\:base AND r.ispartof.pid:uk-ac-man-ems\:2012010001

Number of email message objects in (e.g.) email sequence 2012010/001

r.isderivationof.pid:uk-ac-man-emm\:base AND r.ispartof.pid: uk-ac-man-ems\:2012010001

Number of email attached files in (e.g.) email sequence 2012010/001

r.isderivationof.pid:uk-ac-man-emh\:base AND r.ispartof.pid: uk-ac-man-ems\:2012010001

'Base objects' define different types of object (for the Carcanet Press Email Archive, these are collection, accession, email sequence, email folder and email message).

Use the 'Base object PIDs' for these different classes of object as follows:

a) Collection object - uk-ac-man-col:base

b) Accession object - uk-ac-man-ema:base

c) Email sequence object - uk-ac-man-ems:base

d) Email folder object - uk-ac-man-emf:base

e) Email message object - uk-ac-man-emm:base

f) File attachment - uk-ac-man-emh:base

Use the email collection object PID:

g) Carcanet - uk-ac-man-col:cpa

Use the email accession object PIDs:

h) 2012010 - uk-ac-man-ema:2012010

i) 2012011 - uk-ac-man-ema:2012011

j) 2013032 - uk-ac-man-ema:2013032

Use the email sequence object PIDs as follows:

k) 2012010/001 - uk-ac-man-ems:2012010001

l) 2012010/002 - uk-ac-man-ems:2012010002

m) 2012010/003 - uk-ac-man-ems:2012010003

n) 2012010/004 - uk-ac-man-ems:2012010004

o) 2012011/001 - uk-ac-man-ems:2012011001

p) 2012011/002 - uk-ac-man-ems:2012011002

q) 2013032/001 - uk-ac-man-ems:2013032001

r) 2013032/002 - uk-ac-man-ems:2013032002

s) 2013032/003 - uk-ac-man-ems:2013032003

Step 2.8.3.3 Identify difference between expected and actual ingested and indexed objects

Obtain a list of index PIDs using the following SOLR query (this provides the PIDs of all objects excluding file attachment records):

r.ispartof.pid:uk-ac-man-ems\:<email sequence number>2013032001 AND NOT r.isderivationof.pid:uk-ac-man-emh\:base

e.g.:

r.ispartof.pid:uk-ac-man-ems\:2013032001 AND NOT r.isderivationof.pid:uk-ac-man-emh\:base

You now need to download ONLY the PID field, without any of the other elements.

The above query (taking sequence 2013032001 as an example) should have generated a URL which looks like this:

http://<URL for local SOLR index>/carcanet/select/?q=r.ispartof.pid%3Auk-ac-man-ems%5C%3A2013032001+AND%0D%0ANOT+r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase&version=2.2&start=0&rows=10&indent=on

Edit this URL manually by add the statement &fl=PID&omitHeader=true to the end of the URL, in order to generate a results file that lists ONLY the PID field, as follows:

http://<URL for local SOLR index>/carcanet/select/?q=r.ispartof.pid%3Auk-ac-man-ems%5C%3A2013032001+AND%0D%0ANOT+r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase&version=2.2&start=0&rows=10&indent=on&fl=PID&omitHeader=true

Only a small number of rows (i.e. PIDs) are returned by default. You need to change the number of rows that are returned in order to list all the PIDs in a particular sequence. You should already have the total number of expected PIDs from the searches run in Step 2.8.3.2. Change the rows parameter from the default figure of 10, to the number of PIDs you expect to see, e.g. change rows=10 to rows=99999 where 99999 is the total number of PIDS you wish to analyse. For example:

http://<URL for local SOLR index>/carcanet/select/?q=r.ispartof.pid%3Auk-ac-man-ems%5C%3A2013032001+AND%0D%0ANOT+r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase&version=2.2&start=0&rows=1500&indent=on&fl=PID&omitHeader=true

If you have a very large number of PIDs, you can obtain PIDS for just a subset of records by changing the 'start' and 'rows' parameters, e.g. start=1000&rows=1100 would return 100 records starting at the 1,000th record. Note you can also use the sort= parameter to order records, e.g. "sort=m.from.name asc" (NB note the space delimiter after the m.from.name field) will sort records by the senders name in ascending order.

Once you have run a search for the required PIDs, save the XML data to a file (enter CTRL-s) called <accession code><email sequence number>_actualpids.xml e.g. 2013032001_actualpids.xml.

NB: you should NEVER download the full metadata fields for a large number of records as this will cause server problems: ensure you limit queries to specific fields (as above) for lots of records, or full fields for a small number of records.

Step 2.8.3.4 Automated method for assessing metrics

To avoid running the whole of Steps 2.8.3.1-3 manually, there is a command which automatically produces an audit.xml file. This summarises the expected and actual metrics (number of folders, emails, attachments, and total objects), and lists all the PIDs for every missing message and attachment (although in its current version, these are listed separately rather than grouping attachments with messages, where both are missing).

This script interrogates the folderpids.xml and the emailpids.xml files for each email sequence, and compares the expected figures from these files with the actual numbers found in the SOLR index.

It will not provide figures from the FOXML folders or the EAD records, so it is still worth checking these sources manually as set out in Step 2.3.8.1 to see if the expected figures match those produced in the audit.xml file.

The differences identified in the audit file can then be further investigated as described in Step 2.3.8.6. The audit command is as follows:

> java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateAuditFoldersEmails.xsl" –o:" <drive letter>:\<collection code>\<accession number>\<email sequence number>\audit.xml" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> <ENTER>

Step 2.8.3.5 'Touch' the index for identified PIDS and repeat step 2.8.3.3

This step should resolve situations where objects have been successfully ingested but have not been indexed.

This step MUST be run by a system administrator with the necessary credentials for access to the digital objects.

Run the script:

> java -Xms512m -Xmx512m -jar c:\saxon\saxon9.jar –s: "<drive letter>:\<collection code>\<email sequence number\<accession code><email sequence number>_diffpids.xml" -xsl:"<drive letter>:\<collection code>\XSLT-Transforms\CreateTouchScript.xsl" drive=<drive letter> collection=<collection code> accession=<accession number> emailsequence=<email sequence number> user=<fedora username> password=<fedora password><ENTER>

This will create a file called <accession code><email sequence number>_touch.bat. This file contains commands that will reindex the PIDs that were expected but have NOT been indexed. NB this file contains sensitive user credentials and MUST be kept secure.

Step 2.8.3.6 Tidy up any uningested objects and reingest

Once you have checked that all ingested objects have been indexed correctly there may be some objects left that failed to ingest. It may be worth inspecting the ingest log files for these PIDs and comparing with the list of PIDs identified above. For failed objects a judgement (based on archival appraisal) needs to be made on whether to attempt reingesting the individual objects. In some cases, a decision can be based simply on reviewing the attachment titles. In other cases, objects will need to be interrogated in more detail following the steps below. If a decision is made to reingest objects, it will may be necessary to modify the names of the file attachments, regenerate Jacksum and FITS for these files, edit the datastreams and FOXML files.

Even where it is only attachments that have failed to ingest, it is simpler to reingest entire objects than simply the attachment datastream.

These are the steps involved:

3) Identify PIDs of failed objects from the audit.xml file.

4) Note their MD5 checksums, then find the email messages and file attachments for the relevant objects in the folder structure (i.e. the Aid4Mail extract) using the MD5 checksum; this forms the foldername for each object. Review these objects by opening one or more of the email files (.msg, .eml, .mht, .xml) and/or file attachments, and come to a decision about whether a second ingest should be attempted. If NOT, make a record of the objects omitted and include them in the secure delete schedule at the end of the

project; or save them in a secure folder on the s: drive. If YES, carry out the following steps.

5) Copy the relevant MD5 folders and ALL their contents to a new folder (one per sequence). It is important to retain the MD5 folder name and subfolder structure.

6) If you wish to ingest objects successfully, then you may need to change file attachment names where this is the source of the failed ingest, e.g. removal of double spaces, removal of non-Latin characters etc. Do this on the copy, ensure a version of the original is also kept, and make a note of your 'editorial principles' (e.g. double spaces have been reduced to single space; accented characters have been retyped without the accent).

7) Load the related emailpids.xml or folderpids.xml files for the relevant email sequence

8) Find records in the emailpids.xml/folderpids.xml that correspond to the failed PIDs.

9) Copy these records to a NEW version of emailpids.xml/folderpids.xml.

10) Edit these records, changing any names of file attachments in line with the modifications made above. You may equally remove the reference to any file attachment in the emailpids record. This means you would ingest objects without the file attachments.

11) Rerun FITS (on file attachments if you have changed their titles) and/or Jacksum (on email messages) (see Steps 2.3-4), create datastreams (see Step 2.6), create folder FOXML and email FOXML (see Step 2.7). NB: recreating the folder XML is only necessary if you wish to ingest missing folders; you need to take care with this because the descriptive metadata may be wrong; ensure the number of subfolders, emails etc given is highlighted as being indicative only.

12) Purge the selected objects from eScholar. Run a SOLR query to find the PIDs of the objects you wish to purge; you may wish to edit this manually (NB: ONLY include the PID field in the search results). Then use CreateFedoraBatchModifyPurge.xsl to transform the SOLR results into a Fedora Batch Modify script. Load the Fedora Client tool and process this batch modify script (a system administrator must do this).

13) Regingest these FOXML and MD5 datastreams.

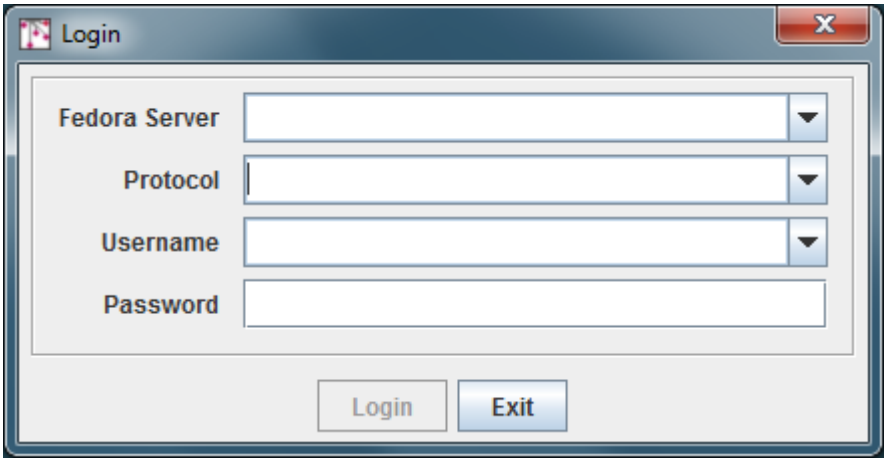
14) Rerun the audit actions outlined in Steps 2.3.8.1-5.

Step 2.8.4 Update datastream in existing Fedora digital object

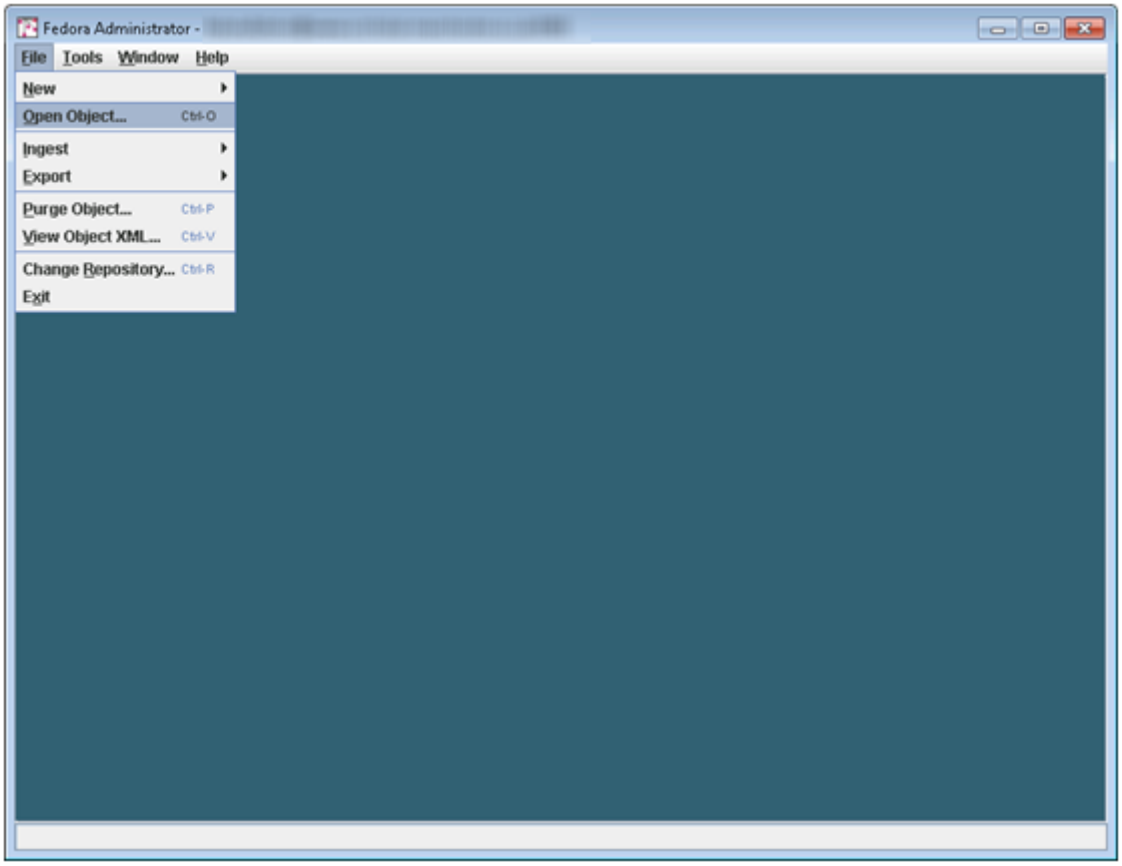
On occasion it may be necessary to update an individual datastream associated with a single Fedora digital object. Fedora offers a number of routes to do this but the most convenient is to use the Fedora Client software. The following example illustrates how to update the EAD XML datastream for the collection digital object (as needed when a new accession is created). This approach may equally be adopted to update other XML based datastreams.

1. Execute the Fedora client software by running the fedora-admin.bat file; depending on how Fedora was installed this command can be found at <fedora installation folder>\client\bin\fedora-admin.bat> (for example on the Workbench PC the fedora client software is located at C:\fedora\client\bin\fedora-admin.bat); browse to the correct folder and double click the fedora-admin.bat file (alternatively create a shortcut on the desktop or task bar to this file and double click one of those).

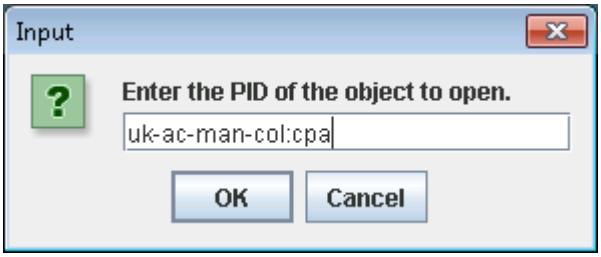
2. Once the Fedora client software is running you are prompted for the repository location and user credentials to make a connection. Below is a screenshot of the dialog where you enter these details. These details need to be provided by a Fedora system administrator and SHOULD NOT be distributed. Enter the details and click <Login>



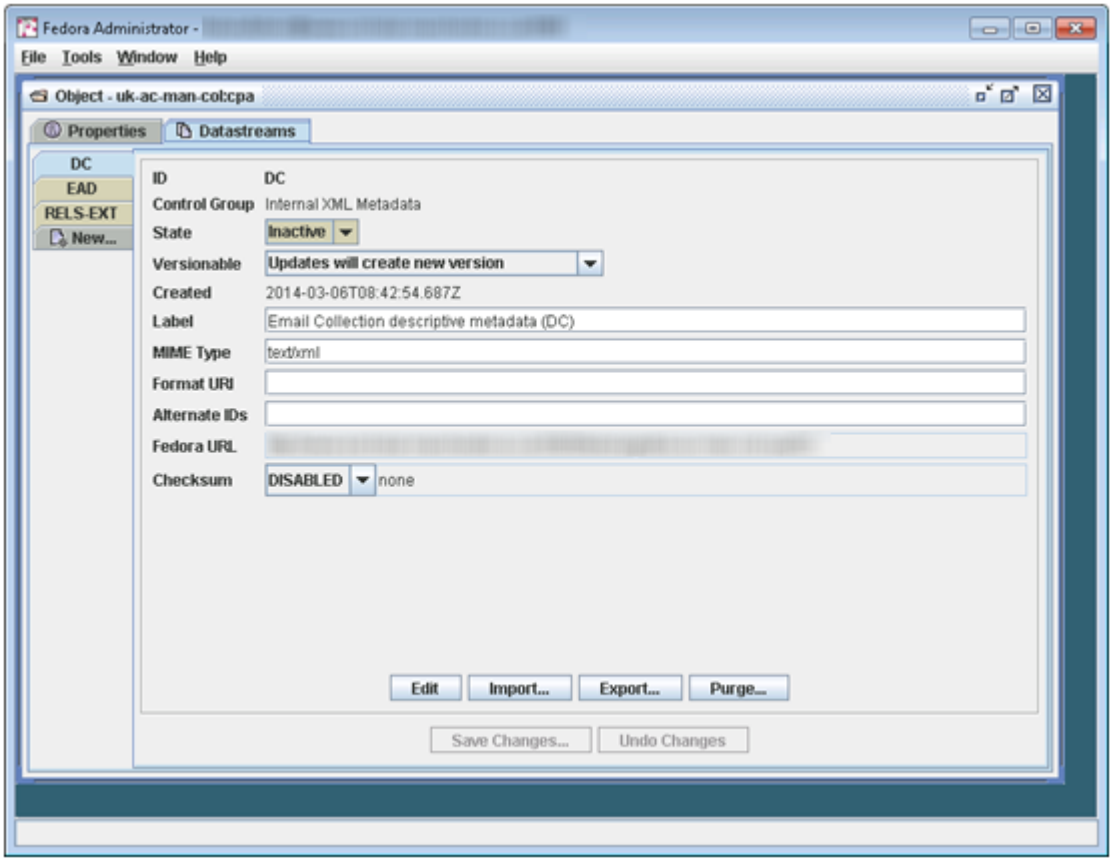
3. Open the digital object you wish to update a datastream for. Select File -> Open Object (alternatively type CTRL-O).



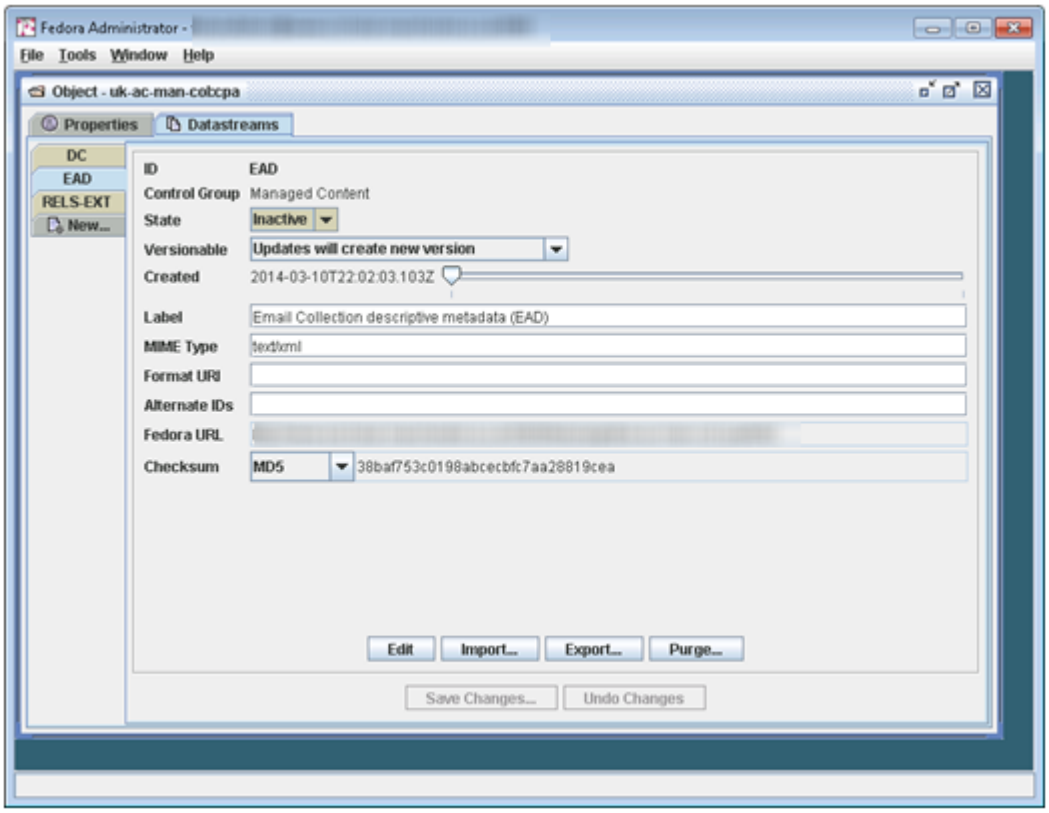
4. At the dialog prompt, enter the PID of the digital object and click “Ok”.



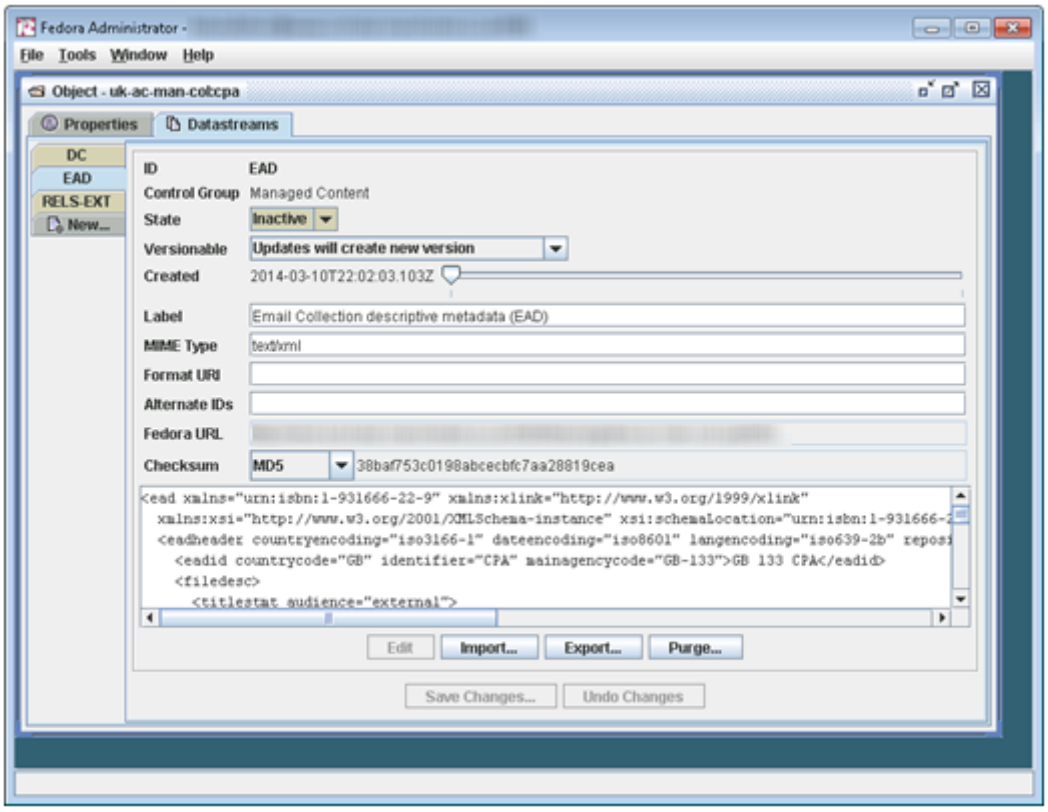
5. Select the “Datastreams” tab.



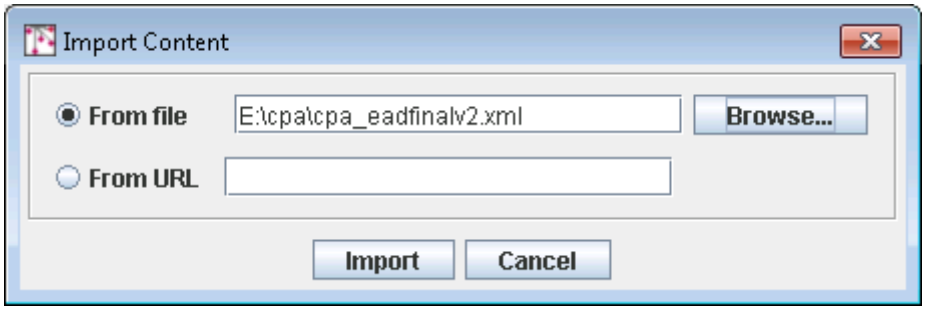
6. Select the datastream you wish to edit from the left had tabs e.g. in this case select “EAD” in the left hand tabs.



7. If you wish to make a small change click the “Edit” button and manually edit the datastream. Once you have edited the datastream click “Save Changes…”.



Alternatively, if there are a lot of changes to the datastream you may wish to generate a new XML file outside the Fedora Client software and then choose “Import”. Choose the “From file” radio button and then click the “Browse” button to find the new XML file



8. Click the “Import” button.
9. Once you have successfully edited the existing datastream or imported a new version you should see the Created slider bar move and a new datetime appearing, indicating a new version of the datastream has been attached to the digital object.
10. To finish this step close the digital object and then close the Fedora Client Software.

Step 2.9: Index

2.9.1 Search examples

General SOLR syntax is as follows:

<field name>:<field value, colon escaped with ‘\’ character>

Which means <field name> = <field value>

Range search syntax is as follows:

<field name>:[<from value> TO <to value>]

The following examples demonstrate some of the common searches you may wish to carry out.

1. Find an individual email message, folder, sequence, accession or collection object

   PID:uk-ac-man-emm\:2012010002e4204

2. Find an individual file attachment

   ID:FILE_2012010002e421_1

3. Find all email messages

   r.isderivationof.pid:uk-ac-man-emm\:base

4. Find all email folders

   r.isderivationof.pid:uk-ac-man-emf\:base

5. Find all file attachments

   r.isderivationof.pid:uk-ac-man-emh\:base

6. Find email messages associated with a particular accession *2012011*

   r.ispartof.pid:uk-ac-man-ems\:2012011001 OR r.ispartof.pid:uk-ac-man-ems\:2012011002

7. Find email messages associated with a particular email sequence *2012011/002*

   r.ispartof.pid:uk-ac-man-ems\:2012011002

8. Find email messages associated with a particular folder (excluding messages in subfolders)

   r.ischildof.pid:uk-ac-man-emf\:2012011001f101

9. Find email messages associated with the folder name ''Inbox\PBS"

   m.folder:Inbox\\PBS

10. Find email folders where the folder name begins with the word 'inbox'

    m.folder:inbox AND r.isderivationof.pid:uk-ac-man-emf\:base*

11. Find all email messages with file attachments

    r.isderivationof.pid:uk-ac-man-emm\:base AND f.isfileattached:Yes

12. Find all file attachments of a particular mimetype (text/html)

    r.isderivationof.pid:uk-ac-man-emh\:base AND f.file.mimetype:text/html

13. Find all email messages received in a day (28th Feb 2013), month (March 2013), year (2008) and a date range (3rd Feb 2012 to 16th Mar 2013)

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.received.day:2013-02-28

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.received.month:2013-03

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.received.year:2018

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.received:[2012-02-03T00:00:00Z TO 2013-03-16T00:00:00Z]

14. Find all email messages sent to and received from a particular correspondent (Michael Schmidt)

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.to.name:"Michael Schmidt"

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.from.name: " Michael Schmidt"

15. Find all email messages that have 'pn review' in the message body

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.body:"pn review"

16. Display ONLY the PID, message subject and message body in search results

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.body:"pn review"

Display specific URL modification:

&fl=PID,m.subject,m.body

Post modification URL:

http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase+AND+m.body%3A%22pn+review%22&version=2.2&start=0&rows=10&indent=on&fl=PID,m.subject,m.body

17. Display the 15th to 21st search result

    r.isderivationof.pid:uk-ac-man-emm\:base

    Display specific URL modification:

    &start=15&rows=6

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase%0D%0A&version=2.2&start=15&rows=6&indent=on

18. Find the top 50 messages most recently sent to a particular correspondent (Michael Schmidt)

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.to.name:"Michael Schmidt"

    Sort specific URL modification:

    &start=0&rows=50&sort=m.received.day desc

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase+AND+m.to.name%3A%22Michael+Schmidt%22%0D%0A&version=2.2&start=0&rows=50&indent=on&sort=m.received.day%20desc

19. Find the top 100 largest file attachments associated with a particular email sequence (2013032001)

    r.isderivationof.pid:uk-ac-man-emh\:base AND r.ispartof.pid:uk-ac-man-ems\:2013032001

    Sort specific URL modification:

    &start=0&rows=100&sort=f.file.size desc

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase+AND+r.ispartof.pid%3Auk-ac-man-ems%5C%3A2013032001%0D%0A&version=2.2&start=0&indent=on&rows=100&sort=f.file.size%20desc

20. Determine the frequency of email messages sent by different individuals

    r.isderivationof.pid:uk-ac-man-emm\:base

Facet specific URL modification:

&facet=true&facet.field=m.from.name.facet

Post modification URL:

http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase&version=2.2&start=0&rows=0&indent=on&facet=true&facet.field=m.from.name.facet

21. Determine the frequency of email messages sent from different individuals

    r.isderivationof.pid:uk-ac-man-emm\:base

    Facet specific URL modification:

    &facet=true&facet.field=m.to.name.facet

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase&version=2.2&start=0&rows=0&indent=on&facet=true&facet.field=m.to.name.facet

    NB: to show multiple facets, amend the URL with &facet.field=<facet name> once for each facet required

22. Determine the frequency of email messages sent and received over time, by day, by month and by year

    r.isderivationof.pid:uk-ac-man-emm\:base

    Facet specific URL modification:

    &facet=true&facet.field=m.received.year&facet.field=m.received.month&facet.field=m.received.day

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase&version=2.2&start=0&rows=0&indent=on&facet=true&facet=true&facet=true&facet.field=m.received.year&facet.field=m.received.month&facet.field=m.received.day

23. Find the 1000 most frequently used words in the email body for emails received in a particular month (Feb 2011, NB: don't run this query against too many emails; always limit it to a few thousand or less)

    r.isderivationof.pid:uk-ac-man-emm\:base AND m.received.month:2011-02

    Facet specific URL modification:

    &rows=0&indent=on&facet=true&facet.field=m.body &facet.mincount=1&facet.limit=1000

    Post modification URL:

http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emm%5C%3Abase%0D%0AAND+m.received.month%3A2011-02%0D%0A&version=2.2&start=0&rows=0&indent=on&facet=true&facet.field=m.body&facet.mincount=1&facet.limit=1000

24. Determine frequency of file attachment mimetypes

    r.isderivationof.pid:uk-ac-man-emh\:base

    Facet specific URL modification:

    &facet=true&facet.field=f.file.mimetype

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase&version=2.2&start=0&indent=on&rows=0&facet=true&facet.field=f.file.mimetype

25. Determine frequency of file attachment file sizes, 0-20Kb, 20-50Kb, 50-100Kb, 100-200Kb, 200-500Kb, 500Kb-1Mb, 1-2Mb, 2-5Mb, 5-10Mb, 10-50Mb, 50-100Mb

    r.isderivationof.pid:uk-ac-man-emh\:base

    Facet specific URL modification:

    &facet=true&facet.query=f.file.size:[0 TO 20000] &facet.query=f.file.size:[20001 TO 50000]&facet.query=f.file.size:[50001 TO 100000]&facet.query=f.file.size:[100001 TO 200000]&facet.query=f.file.size:[200001 TO 500000]&facet=true&facet.query=f.file.size:[500000 TO 1000000]&facet.query=f.file.size:[1000001 TO 2000000]&facet.query=f.file.size:[2000001 TO 5000000]&facet.query=f.file.size:[5000001 TO 10000000]&facet.query=f.file.size:[10000001 TO 50000000]&facet.query=f.file.size:[50000001 TO 100000000]

    Post modification URL:

    http://<URL for local SOLR index>/carcanet/select/?q=r.isderivationof.pid%3Auk-ac-man-emh%5C%3Abase&version=2.2&start=0&rows=0&indent=on&omitHeader=true&facet=true&facet.query=f.file.size:[0%20TO%2020000]%20&facet.query=f.file.size:[20001%20TO%2050000]&facet.query=f.file.size:[50001%20TO%20100000]&facet.query=f.file.size:[100001%20TO%20200000]&facet.query=f.file.size:[200001%20TO%20500000]&facet=true&facet.query=f.file.size:[500000%20TO%201000000]&facet.query=f.file.size:[1000001%20TO%202000000]&facet.query=f.file.size:[2000001%20TO%205000000]&facet.query=f.file.size:[5000001%20TO%2010000000]&facet.query=f.file.size:[10000001%20TO%2050000000]&facet.query=f.file.size:[50000001%20TO%20100000000]

2.9.2 Index schema

All email digital objects are indexed in Apache SOLR which enables searching and analysis of the collection. Indexed and searchable fields are derived from the different datastreams that make up the digital objects and are tailored to meet a generalised and rich set of searching requirements. These include simple searches, advanced searches, search facets, display options and sorting of records. Section 2.9.1 gives example searches which may be combined in a range of ways. The below is a full list of the indexed fields, names and a brief description for each, available for searching and display of the collection's content.

Table 2.9.2.1 Tokenized fields suitable for phrase, word and wildcard searches

Field name	Field description
ID	Record identifier, used to uniquely identify the SOLR record, this can be the same as the PID but is different for attachment file records
PID	Digital object persistent identifier, use this to identify any individual digital object
allfields	Concatenated metadata, used for doing 'google-like' all fields search
f.isfileattached	Yes or No, indicates whether one or more email file attachments exist, one entry for each attached file, specific to email message records
f.fileattached.id	Datastream identifier(s) of attached file, one entry for each attached file, specific to email message records
f.fileattached.name	Name(s) of attached file, one entry for each attached file, specific to email message records
f.fileattached.mimetype	Mimetype(s) of attached files, one entry for each attached file, specific to email message records
f.fileattached.source	Concatenated file metadata, used to relate the f.fileattached.* fields for display purposes, specific to email message records
f.file.id	File datastream identifier, specific to file attachment records
f.file.name	File datastream name, specific to file attachment records
f.file.mimetype	File datastream mimetype, specific to file attachment records
f.file.created	File datastream created date time, specific to file attachment records
f.file.created.day	File datastream created day yyyy-mm-dd, specific to file attachment records
f.file.created.month	File datastream created month yyyy-mm, specific to file attachment records
f.file.created.year	File datastream created year yyyy, specific to file attachment records
f.file.lastmodified	File datastream lastmodified date time, specific to file attachment records
f.file.lastmodified.day	File datastream lastmodified day yyyy-mm-dd, specific to file attachment records
f.file.lastmodified.month	File datastream lastmodified month yyyy-mm, specific to file attachment records
f.file.lastmodified.year	File datastream lastmodified year yyyy, specific to file attachment records
f.file.size	File datastream size in bytes, specific to file attachment records
f.file.preingest.md5	File datastream pre-ingest md5 checksum, specific to file attachment records
f.file.postingest.md5	File datastream post-ingest md5 checksum, specific to file attachment records
f.file.email.pid	PID of digital object file datastream is attached to, specific to file attachment records
f.file.source	Concatenated file metadata, used to relate the f.file.* fields for display purposes, specific to file attachment records
m.subject	Email message subject
m.body	Email message body, plain-text content

m.size	Email message size
m.received	Email message date-time received
m.received.day	Email message day received, yyyy-mm-dd
m.received.month	Email message month received, yyyy-mm
m.received.year	Email message year received, yyyy
m.sent.utc	Email message sent date-time in UTC form
m.recieved.utc	Email message received date-time in UTC form
m.sent.local	Email message sent date-time in local time-zone form
m.recieved.local	Email message received date-time in local time-zone form
m.sender.name	Email message sender name
m.sender.address	Email message sender address
m.from.name	Email message from name
m.from.address	Email message from address
m.to.name	Email message to name(s)
m.to.address	Email message to address(es)
m.cc.name	Email message cc name(s)
m.cc.address	Email message cc address(es)
m.bcc.name	Email message bcc name(s)
m.bcc.address	Email message bcc address(es)
m.reply_to.name	Email message reply to name(s)
m.reply_to.address	Email message reply to address(es)
m.resent_from.name	Email message resent from name(s)
m.resent_from.address	Email message resent from address(es)
m.resent_cc.name	Email message resent cc name(s)
m.resent_cc.address	Email message resent cc address(es)
m.folder	Full path name of email folder containing the email message
m.folder.path	Individual folder paths, one for each folder in the folder hierarchy e.g. the full folder path Inbox\subfolder1\subfolder2 would have the entries Inbox; Inbox\subfolder1; Inbox\subfolder1\subfolder2, used to generate a hierarchical tree structure
r.iscreatedby.az	The first letter of the family name of the individual(s) who have created the object, used to provide an A to Z list of individual(s) names
r.iscreatedby.pid	PID of person object that describes the individual that first created the object, this is a currently a constant and set to uk-ac-man-per:48079 (Phil Butler) although this should change in ingests of future accessions.
r.iscreatedby.source	Concatenated details of the individual(s) who created the object, formatted as: Family name, Givenname||PID||Spot ID||Givenname Familyname
r.islastmodifiedby.az	The first letter of the surname of the individual(s) who last modified the object, used to provide an A to Z list of individual(s) names
r.islastmodifiedby.pid	PID of person object that describes the individual who last modified the object, this is currently a constant and set to uk-ac-man-per:48079 (Phil Butler) but would change when objects are edited in the future
r.islastmodifiedby.source	Concatenated details of the individual(s) who last modified the object, formatted as: Family name, Givenname||PID||Spot ID||Givenname Familyname

r.isderivationof.pid	PID of the base object that describes the class of digital object (see section 2.8.3.2: Identify ingested and indexed objects for PID values)
r.ischildof.pid	PID of folder email digital object the email message is a child of e.g. uk-ac-man-emf:2012010001f1
r.ispartof.pid	PID of email sequence digital object email message is part of e.g. uk-ac-man-ems:2012010001
r.ismemberof.pid	PID of collection object, uk-ac-man-col:cpa
x.createddate	Date-time digital object was ingested
x.createddate.day	Day digital object was ingested, yyyy-mm-dd
x.createddate.hour	Hour digital object was ingested
x.lastmodifieddate	Date-time digital object was last modified
x.state	Active or Inactive, all email digital objects have been set to Inactive to keep them private
x.label	Label for type of digital object
x.ownerid	PID of individual who originally created the digital object
timestamp	Date-time record indexed

In addition to the search and display fields listed in Table 2.9.2.1, Table 2.9.2.2 lists a set of fields that have been added to the SOLR index to facilitate sorting, filtering and faceted searches across different classes of objects. These fields are untokenized which means they are NOT suitable for doing single word or partial word searches. Instead they form vocabularies of words or phrases that can be used to form a defined set of terms in advanced browse and filter functions.

Table 2.9.2.2 Untokenized fields available for facet and browse searches

Field name	Field description
from	Common field name for display and sorting purposes, untokenized version of m.from.name for message objects, m.folder from folder objects; will also be added for sequence, accession and collection objects
to	Common field name for display and sorting purposes, untokenized version of m.to.name for message objects (NB: not available for folder, sequence, accession and collection objects)
subject	Common field name for display and sorting purposes, untokenized version of m.subject for message and folder objects (NB: not available for sequence, accession and collection objects)
received	Common field name for display and sorting purposes, untokenized version of m.received.date for message and folder objects (NB: not available for sequence, accession and collection objects)
size	Common field name for display and sorting purposes, untokenized version of m.size for message objects; will also be added for folder, sequence, accession and collection objects
folder	Common field name for display and sorting purposes, untokenized version of m.folder for message and folder objects (NB: not available for sequence, accession and collection objects)
filename	Not available but may want to consider adding this to index of file attachments, would be equivalent to f.file.name
filemimetype	Not available but may want to consider adding this to index of file attachments, would be equivalent to f.file.mimetype
filesize	Not available but may want to consider adding this to index of

	file attachments, would be equivalent to f.file.size
filecreated	Not available but may want to consider adding this to index of file attachments, would be equivalent to f.file.created
filelastmodified	Not available but may want to consider adding this to index of file attachments, would be equivalent to f.file.lastmodified
m.from.name.facet	Untokenized version of m.from.name
m.to.name.facet	Untokenized version of m.to.name
m.cc.name.facet	Untokenized version of m.cc.name
m.bcc.name.facet	Untokenized version of m.bcc.name
f.fileattached.source.facet	Untokenized version of f.fileattached.source
r.iscreatedby.source.facet	Untokenized version of r.iscreatedby.source
r.islastmodifiedby.source.facet	Untokenized version of r.islastmodifiedby.source

---

Step 2.10: Access (secure)

Currently, the Carcanet Press Email Archive is embargoed to researchers for data protection, IPR and technical reasons. Access to the archive is limited to the curator and a small number of system administrators, and requires authentication by username and password. Read access to metadata is available using Manchester eScholar's SOLR index. This is comprehensive, and includes the full text of every email as well as extensive metadata.

Access for the purpose of editing or deleting any of the digital objects and/or their component datastreams is via the Fedora Administrative Client or the Fedora Application Programming Interface, and is restricted to a small number of system administrators.

Currently, a curatorial tool is under development, due for completion in August/September 2014: this will provide a user-friendly mechanism for the curator to access, interrogate and manage the archive.

Step 2.11: Secure deletion

Inevitably some email objects and attachments will not ingest successfully. If there are objects you wish to retain for further investigation rather than zipping them and ingesting them, make sure these are saved somewhere safe on the secure network drive.

Once all other objects have been successfully ingested and verified, it is necessary to securely delete ALL storage spaces where pre-ingest archival material has been saved, including the relevant partitions of the Workbench PC, and (where relevant) the removable hard drive and Quarantine PC. Secure deletion is carried out using Active@ KillDisk software. For instructions on how to do this, see Steps 1.28-30.

Appendix: transfer list pro-forma

Transfer list for digital archive accessions made by removable media

Donor/Depositor details:	
Name	
Address	
Telephone	
Email	
Details of UML curator and technical staff:	
Name	
Position	
Name	
Position	
Creator, if different from principal donor/depositor:	
Name	
Position	
Schedule of transferred material:	
Collection name, and reference code if accrual to existing collection	
Content description *[Brief summary including record types, subject coverage, arrangement]*	

Covering dates *[Approximate covering dates for the material]*	
Extent of the material *[In MB/GB]*	
Technical description *[Include information on: details of PC from which material was taken, and location on PC; file formats; passwords/encryption applied by creator]*	
Restrictions *[Information about copyright, personal data, and confidential material included in the accession]*	
Items for disposal *[Identify any specific folders/items which are scheduled for secure deletion before processing begins – highlighting on appended schedule if possible]*	
Signatures:	
Signature of donor/depositor or authorised representative	
Name of signatory	

Date	
Signature of curator	
Name of signatory	
Date	