Request for proposals
Gloucestershire Archives
{heather.forbes, claire.collins}@gloucestershire.gov.uk
23 March 2021

Introduction

Gloucestershire Archives (the Archive) is the local government archive service of Gloucestershire County Council and South Gloucestershire Council. The Archive has been active in the field of "digital preservation" for over a decade and has identified features and functions that are believed to be not currently addressed by the digital preservation sector.[1]

In order to further develop the practice of digital curation by the Archive we now issue this request for proposals which we describe as two work packages

a) the digital curation "packager", and
b) the digital curation "storage (fixity) manager".

The purpose of this request for proposals is to identify a development partner that will implement their proposal to a solution that allows the Archive to test and subsequently use operationally.

To date the Archive has fulfilled a leadership role within the local government digital preservation community and anticipates that an appropriate solution could be adopted for wider use with the sector. A key feature given especially its open source nature is that there be some form of additional commercial support available.

The descriptions below summarise our expectations. Given the short timescale and the limited budget it is expected that proposals will address a minimum viable product. However it will be an advantage to include "hooks" for future feature developments.

The proposal must also include a proposed support strategy. Ongoing support cost should be indicated.

In the event that you are unable to propose a response in respect of the work packages, it would be most helpful if you could (i) let Gloucestershire Archives know, and (ii) suggest any other organisation(s) that you think may be interested in responding.

The digital curation "packager"

The packager is considered to comprise several functional components which in principle might be thought of as enhanced versions of BagIt. It will be an advantage if the packager is similarly a Python application/module. It will also be an advantage if the overall architecture is modular. The packager is intended to replace an existing Perl based application used by archivists at the Archive. This application is run locally on both GNU/Linux and corporate Windows 7/10 platforms.

i. GUI front end to support package payload identification and selection, adjunct document identification and selection, and package archival description. A screen

1 https://www.gloucestershire.gov.uk/media/2094490/digital-preservation-for-local-authorities.pdf

shot of the existing description GUI is shown in the appendix. This illustrates the ISAD(G) elements used. An adjunct document is a document (file) included in the bag structure but not within the bag payload.

Use of the GUI should result in a valid BagIt structure containing the selected payload. The manifest must include at least two cryptographic hashes of each payload file, preferably SHA. The structure must be identified by a UUID.

The structure must also contain a valid METS file named “gaip.xml” co-located with the “bagit.txt” file. It is expected that this will express metadata information obtained from two sources. The first is the package payload itself. It is expected that the metadata extracted will be compatible with that revealed using ExifTool and PRONOM, for example via “opf-fido”. The second source of metadata is the archivist’s description of the payload. Please see the appendix for a non-normative example of a gaip.xml file.

The GUI must also facilitate “reading” a serialised (and possibly compressed) BagIt bag and displaying the archival description given in the gaip.xml file.

ii. There must be a serialise without compression and save function which takes the UUID identified BagIt structure created as described above and saves it as a file named to match the UUID in a selected directory path. It is expected that the file name of the file generated will be just the UUID, no file extension. Serialisation must use the “zip” algorithm and must be able to support large (> 4GB) files. It is expected that this functionality will be accessible via the GUI and possibly via the command line also.

iii. There must be a “double bagging” with optional encryption function. This takes as input the file that is the serialised bag described above. This must be encrypted. The encryption must be GnuPG compatible and make use of a secret pass phrase for simple file encryption, not PKI. A protocol for maintaining the secrecy of the pass phrase should be proposed. For example, the pass phrase could be retained in a configuration file that has restricted access but is available programatically. Prior to encryption the gaip.xml file content must be noted since a plain text version of the METS file is needed for double-bagging.

Double bagging refers to making a BagIt bag, the “outer bag”, that contains an “inner bag” as payload. The payload of the inner bag is thus “double bagged”. The motivation here is that when the inner bag is encrypted the Archive is assured that information remains confidential even if exfiltrated. Authentic transmission is assured by the manifest in the outer bag. A copy of the METS file, gaip.xml, must be included in the outer bag co-located with the outer bag bagit.txt file.

The outer bag must be serialised without compression using the zip algorithm (as above). It is expected that the file name of the file generated will be just the UUID, no file extension.

A schematic diagram is given in the appendix.

It is expected that this functionality will be accessible via the GUI and possibly via the command line also.

The anticipated deliverable in respect of the digital curation “packager” work package is “running code” that can be tested by archivists at the Archive prior to an operational deployment together with an ongoing support strategy including indicative cost.

The digital curation “storage (fixity) manager”

The storage (fixity) manager is expected to provide the principal means for the archivist to interact with the Archive’s digital repository(ies). The digital repository can be assumed to be cloud-based (both on and off premises). However it will be an advantage to also support upload and download to/from LAN attached storage.

It can be assumed that the archivist will have separate access to an archival discovery system that maintains a catalogue of AIP UUIDs.

The storage (fixity) manager must have a GUI that can be accessed from both GNU/Linux and Windows10 platforms.

i. Upload

   The storage (fixity) manager must allow the archivist to securely upload (deposit) selected files (both one or more) to selected repositories (both one or more). It is assumed that the files here are serialised BagIt bags (AIPs). It will be an advantage if this aspect of the storage (fixity) manager is compatible with Sword3.

   The storage (fixity) manager must provide fixity management. That is, it will maintain synchronised copies of a non-proprietary database in all the available repositories (the fixity database). Whenever an AIP is deposited a fixity record must be created. Each fixity record will contain at least two SHA based cryptographic hash values for the deposited AIP and be keyed by the AIP name (UUID). Please note that the focus of interest here is the AIP not its payload.

   Work by the British Library <https://github.com/britishlibrary/mpt> might be relevant.

ii. Fixity monitoring

   The storage (fixity) manager must allow the archivist to initiate an in-repository fixity verification of both individual deposited files (AIPs) and bulk verification, for example by deposit date range. This verification function must re-compute the cryptographic hashes for the selected files and compare these values with those in the fixity database. Please note this involves accessing all copies of the database.

   The result (verified/not verified) must be reported.

   It is expected that the storage (fixity) manager will be compatible with AWS “serverless fixity for digital preservation compliance” although other cloud storage providers must be included.

iii. Download

   The storage (fixity) manager must allow the archivist to securely download (retrieve) selected files (both one or more) from a selected repository. When retrieved each

file must be presented as a serialised BagIt bag (AIP) that is bit-stream identical to its corresponding deposit. It will be an advantage if this aspect of the storage (fixity) manager is compatible with Sword3.

When a file is retrieved its cryptographic hashes must be re-computed and compared with the fixity database as described above. The results (verified/not verified) must be reported. Please note that this requirement applies to bulk downloads also.

The anticipated deliverable in respect of the storage (fixity) manager work package is a demonstration and the opportunity for the Archive's archivists to test using their user data prior to an operational deployment together with an ongoing support strategy including indicative cost.

Constraints

i. Financial

Respondents are encouraged to submit proposals for either or both of the work packages.

Gloucestershire Archives has a strictly limited budget and timescale. Each of the two work packages must be priced at no more than £##### (plus VAT). Payment in respect of one of the work packages must be invoiced before 30 June 2021, the other by 16 December 2021.

It is therefore essential to specify a firm completion date in respect of the proposal for each work package.

ii. Open source

All substantive components/libraries must be open source. Please specify intended licence. User interfaces, for example, need not be open source but unencumbered use should be permitted.

Timetable

The Archive has to work to a compressed timetable. Funding restrictions are explained above.

It is therefore essential that proposals are received as soon as possible and no later than by 6pm 8 April 2021.

Contact

The Archive is happy to provide any extra clarification, explanation and guidance requested and will use best endeavours to respond as soon as possible.

Package bag structures

Bagit bag including one gaip.xml METS file and zero or more adjunct files

```
<uuid>/
  |
  | --- bag-info.txt
  | --- bagit.txt
  | --- manifest-sha256.txt
  | --- manifest-sha512.txt
  | --- tagmanifest-sha256.txt
  | --- tagmanifest-sha512.txt
  |
  | --- gaip.xml
  |
  | --- <adjunct files>
  |
  | --- data/
        |
        | --- <payload>
```

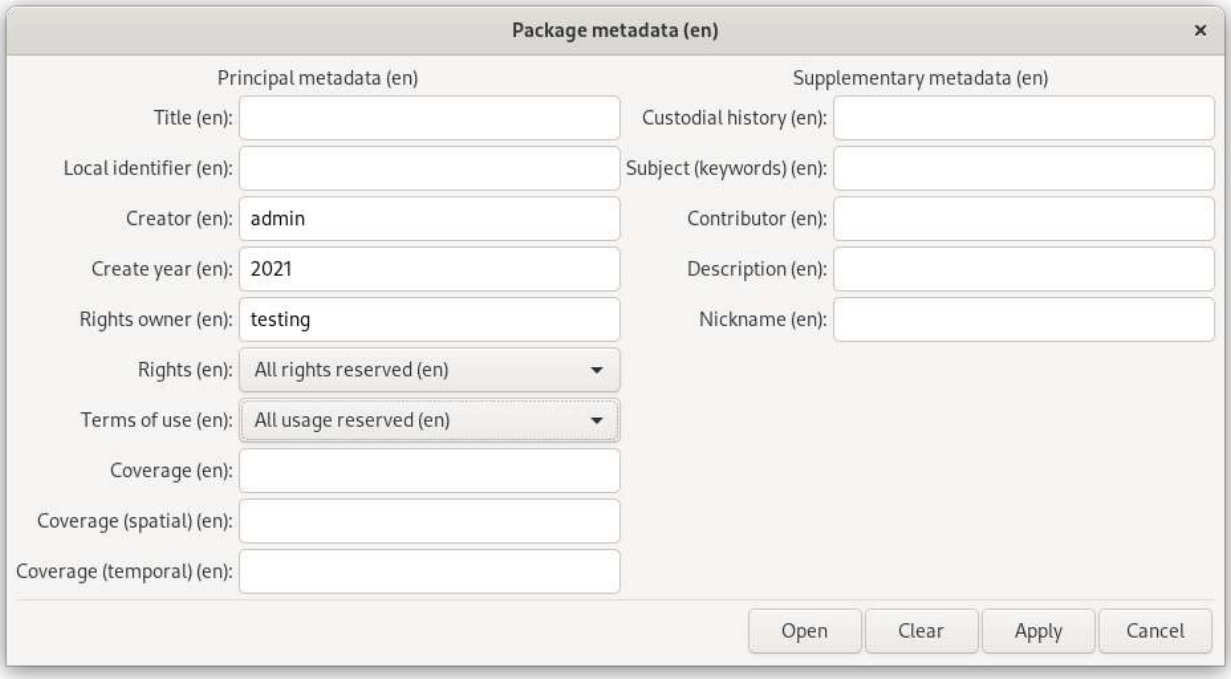
When serialized the bag is represented here by the file "`uuid`"

The encrypted version of this file is also called "`uuid`"; here called `uuid~`.

The double bag is

```
<uuid>/
  |
  | --- bag-info.txt
  | --- bagit.txt
  | --- manifest-sha256.txt
  | --- manifest-sha512.txt
  | --- tagmanifest-sha256.txt
  | --- tagmanifest-sha512.txt
  |
  | --- gaip.xml
  |
  | --- data/
        |
        | --- uuid~
```

Package archival description



The screen shot shows the GUI currently used by archivists to describe the AIP. Information is mapped to DC and XMP as follows.

title	=>	dc:title
local identifier	=>	xmp: Identifier
creator	=>	dc:creator
create year	=>	dc:date
rights owner	=>	xmpRights:Owner
rights	=>	dc:rights
terms of use	=>	xmpRights:UsageTerms
coverage	=>	dc:coverage
coverage (spatial)	=>	dcterms:spatial
coverage (temporal)	=>	dcterms:temporal
custodial history	=>	dcterms:provenance
subject (keywords)	=>	dc:subject
contributor	=>	dc:contributor
description	=>	dc:description
nickname	=>	xmp:Nickname

This is illustrated by the gaip.xml file shown below.

The packager must provide for the archivist having described the package using the labels shown (left column). This is in order to display the descriptions in legacy packages.

However it is expected that the packager will make use of ISAD(G) / EAD3 and that appropriate changes to the gaip.xml METS file will be implemented. The table below

shows the ISAD(G) terms that are considered relevant. (The numeric references are as used in the ISAD(G) specification.)

3.1.1 reference code(s)
3.1.2 title
3.1.3 date(s)
3.1.4 level of description
3.1.5 extent of the unit of description
3.2.1 name of creator(s)
3.2.2 administrative/biographical history
3.2.3 archival history
3.3.1 scope and content
3.4.1 conditions governing access
3.4.2 conditions governing reproduction
3.4.3 language of material
3.6.1 note

A relationship between ISAD(G) and EAD3 is given in <https://www.loc.gov/ead/EAD3taglib/tl_ead3.pdf>.

gaip.xml

The xml document described below, gaip.xml, is believed to be a minimal METS document. Rather than encoding information so that it can be reliably relocated spatially, METS is here used to support the temporal relocation of metadata. In this case the object of interest is the AIP or package.

```
<!-- METS package metadata file generated by 'SCAT' -->
<!-- File created 2017-08-04T13:03:20 -->

<mets xsi:schemaLocation="http://www.loc.gov/METS/
http://www.loc.gov
/standards/mets/version111/mets.xsd">
 <metsHdr CREATEDATE="2017-08-04T13:03:20">
  <agent ROLE="CREATOR" TYPE="ORGANIZATION">
   <name>GA</name>
  </agent>
  <agent ROLE="CREATOR" TYPE="INDIVIDUAL">
   <name>scat</name>
  </agent>
 </metsHdr>
 <dmdSec ID="description_0">
  <mdWrap LABEL="RDF-XMP-DC" MDTYPE="OTHER">
   <xmlData>
    <rdf:RDF>
     <rdf:Description rdf:about="">
      <dc:coverage/>
      <dc:creator>
       <rdf:Seq>
        <rdf:li>scat</rdf:li>
```

```
       </rdf:Seq>
      </dc:creator>
      <dc:date>
       <rdf:Seq>
        <rdf:li>2017</rdf:li>
       </rdf:Seq>
      </dc:date>
      <dc:description>
       <rdf:Alt>
        <rdf:li xml:lang="x-default"/>
       </rdf:Alt>
      </dc:description>
      <dc:format>application/zip</dc:format>
      <dc:rights>
       <rdf:Alt>
        <rdf:li xml:lang="x-default">All_rights_reserved</rdf:li>
       </rdf:Alt>
      </dc:rights>
      <dc:subject>
       <rdf:Bag>
        <rdf:li>tags</rdf:li>
        <rdf:li>are</rdf:li>
        <rdf:li>good</rdf:li>
       </rdf:Bag></dc:subject>
      <dc:title>
       <rdf:Alt>
        <rdf:li xml:lang="x-default">metadata list demo</rdf:li>
       </rdf:Alt>
      </dc:title>
      <dc:type>
       <rdf:Bag>
        <rdf:li>information package</rdf:li>
       </rdf:Bag>
      </dc:type>
     </rdf:Description>
     <rdf:Description rdf:about="">
      <dcterms:provenance/>
      <dcterms:spatial/>
      <dcterms:temporal/>
     </rdf:Description>
     <rdf:Description rdf:about="">
      <xmp:CreateDate>2017-08-04T13:03:20</xmp:CreateDate>
      <xmp:CreatorTool>'SCAT' is Curation And
Trust</xmp:CreatorTool>
      <xmp:Identifier>
       <rdf:Bag>
        <rdf:li>xxxxxxxx</rdf:li>
       </rdf:Bag>
      </xmp:Identifier>
      <xmp:MetadataDate>2017-08-04T13:03:20</xmp:MetadataDate>
      <xmp:Nickname/>
     </rdf:Description>
     <rdf:Description rdf:about="">
```

```
      <xmpRights:Owner>
       <rdf:Bag>
        <rdf:li>GA</rdf:li>
       </rdf:Bag>
      </xmpRights:Owner>
      <xmpRights:UsageTerms>
       <rdf:Alt>
        <rdf:li xml:lang="x-default">All_usage_reserved</rdf:li>
       </rdf:Alt>
      </xmpRights:UsageTerms>
     </rdf:Description>
    </rdf:RDF>
   </xmlData>
  </mdWrap>
 </dmdSec>
 <dmdSec ID="description_1">
  <mdWrap LABEL="RDF-XMP-DC" MDTYPE="OTHER">
   <xmlData>
    <rdf:RDF>
     <rdf:Description rdf:about="">
      <dc:format>application/pdf</dc:format>
      <dc:identifier>fmt/19</dc:identifier>
     </rdf:Description>
     <rdf:Description rdf:about="">
      <dcterms:extent>2954519</dcterms:extent>
     </rdf:Description>
    </rdf:RDF>
   </xmlData>
  </mdWrap>
 </dmdSec>
 <fileSec>
  <fileGrp>
   <file DMDID="description_1" ID="file_1">
    <FLocat LOCTYPE="OTHER"
xlink:href="data/Camm_Groot_1994.pdf"/>
   </file>
  </fileGrp>
 </fileSec>
 <structMap>
  <div TYPE="information_package">
   <fptr FILEID="file_1"/>
  </div>
 </structMap>
</mets>
```