DIGITAL LIBRARIES
University Libraries
UNT

UNT Libraries OAIS Information Package Specification

Date: October 2015

Version: 1.0

Contributors:

Mark Phillips Assistant Dean for Digital Libraries
Hannah Tarver Department Head, Digital Projects Unit
Ana Krahmer Supervisor, Digital Newspaper Unit
Daniel Alemneh Supervisor, Digital Curation Unit
Laura Waugh Repository Librarian for Scholarly Works

Introduction

The UNT Libraries operate a digital repository system built around the functional components designated by the Open Archival Information Systems (OAIS) Reference Model[1]. As such, this requires a formal definition of the specification of the Submission Information Package (SIP), Archival Information Package (AIP), and Dissemination Information Package (DIP) accepted and processed by the UNT Libraries Digital Library Infrastructure.

Because the SIP, AIP, and DIP are closely related in the UNT Libraries' Digital Collections infrastructure, this document is used to describe all three packages. When all three of the formats are discussed together they are referred to as Information Packages (IPs).

Historically the UNT Libraries have referred to the Dissemination Information Packages (DIPs) for the Digital Collections as Access Content Packages (ACP). In the document below there are references to ACPs that can and should be equated to DIPs when mapping these concepts to the OAIS Reference Model.

Components

There are a number of core components and technologies used by the UNT Libraries to package, transfer, verify, and track Information Packages throughout the system. This section enumerates these components and discusses how each is used for packaging and transferring Information Packages throughout the repository.

BagIt

The UNT Libraries' Information Package (IP) formats make use of the BagIt[2] bag container as a way of encapsulating content submitted and stored by the repository; the container functions as a mechanism for checking for the well-formedness and validity of a resource housed within the bag. Many tools have been built in most popular programming languages that allow for individuals to build, change, and validate BagIt bags. In addition, the BagIt format has a mechanism for storing basic metadata information about the BagIt bag itself such as Internal and External Identifiers, descriptions of the resource(s) encapsulated by the BagIt bag, and the person or body responsible for the BagIt bag.

The UNT Libraries' IP profile assumes bags are based on the 0.97 version of the BagIt Specification.

NaMaste File

The UNT Libraries use the NaMaste specification within the IP format to provide a quick human-readable indicator that a given bag is an Information Package, and to verify that it isn't

another kind of BagIt nag that may not be intended as a UNTL Information Package. The UNT Libraries also use the existence of this file as a "sanity check" for the ingest process of IPs into the repository system; i.e., if the NaMaste file is not present and of the correct format then it is quickly handled as "badly formed" IP and moved into a quarantine space with error information explaining why ingest failed for the resource.

The UNT Libraries' SIP profile requires the existence of a NaMaste file named "UNTL-SIP-1.0" in the root of the BagIt bag. Internal text inside this file is the following string "untl aip 1.0"

The UNT Libraries' AIP profile expects the existence of a NaMaste file called "UNTL-AIP-1.0" in the root of the BagIt bag. Internal text inside of this file consists of "what does it say inside."

Coda Directives

The UNT Libraries pass processing information to both the SIP-to-AIP workflow and the AIP-to-ACP workflow using a configuration file called "coda_directive.py". The Coda Directives include information related to how the SIP-to-AIP conversion should proceed. For example, there may be instructions about whether the processor should assume a standard file-naming convention locally referred to as "MagickNumbers," or if the processor should assume a logical grouping of files into a file bundle based on like filenames, once extensions have been removed. Each of these local bundles are referred to as a "fileSet." Directives can be set for each of the "manifestations" within the digital object folder. Further instructions regarding the conversion from AIP-to-ACP may be included, e.g., if a file should be tiled for a zoom interface.

An example coda-directives.py file is included in Appendix 1.

Bag-info.txt

While the bag-info.text file is a part of the BagIt specification, the UNT Libraries have specific assumptions in mind when creating these files. Information included in this file should be sufficient to identify what project or event was responsible for this object, and/or that the object is part of the general collection building process of the UNT Libraries. If it is collection- or project-based, the bag-info.txt should include basic information about the contributor and project that were responsible for creating the project. While not expected to be exhaustive, statements that can apply to all of the files held within the object (e.g., object consists of TIFF images and master OCR files processed with PrimeOCR) might be helpful in the future. The oxum field must be present with an oxum value that represents the contents in the /data/ directory.

An example bag-it.txt file is included in Appendix 2.

Organizational Concepts

The UNT Libraries' Information Packages and the core data model used by the Aubrey access system assume that digital objects are either created or packaged in a way that can be easily understood by other systems. This section describes the organizational components used in the Information Packages throughout the system.

File

A File is the smallest unit in the Information Package. A File can be of any type, though the SIP-to-AIP conversion expects and handles certain formats as preferred and others as non-preferred formats. File names can consist of letters, numbers, spaces, periods, hyphens and underscores. While there are no specific requirements for file-naming conventions, generally the UNT Libraries encourages content creators to use locally-meaningful file-naming conventions when naming their files, and to try to use standard ASCII character sets whenever possible. It is advisable that file names include a standard extension appropriate to the file format. For example: .tiff, .tif, .pdf, .jpeg, .jpg, etc.

fileSet

A fileSet is a bundle of files grouped logically based on the basename of a filename; these are typically multiple files that represent a single unit. For example, a scan of a book page may be saved as a TIFF file, however, for printed text, digital libraries will create an OCR file for that page and possibly a TXT file of the character information from the OCR file, without coordinate information, that can be used for indexing. In this case we may have three files to represent a single page of text in a book: 0001.tif, 0001.ocr, and 0001.txt. If one takes the basename 0001 and groups the files together using this basename, we have what the UNT Libraries refers to as a fileSet.

These fileSets are useful in the UNTL model for attaching information such as pagination data, annotation data, or labels for video and audio clips. A fileSet designates a primary File which forms the basis for other processing instructions in the ingest workflow. These primary Files are taken from the list of preferred formats[3]. The example given above would have "0001.tif" assigned as the primary File in the fileSet by the SIP-to-AIP conversion process.

The coda_directive.py has a directive that allows the creator of the SIP to identify if a resource was created using naming conventions that make use of the concept of fileSets. For other kinds of items (e.g., photographs, or handwritten manuscript pages) the SIP-to-AIP conversion step is directed to process each File it encounters as an individual fileSet.

Manifest

The UNT Libraries allow for multiple "representations" of a resource to be stored logically together within the local data model. For example: a single PDF document submitted as an Electronic Theses and Dissertation (ETD) to the repository would be normalized, for preservation purposes, into its constituent parts (or pages). The result is generally a series of image files, with one image per page of the original PDF. In this example there are two manifestations (PDF and image files) that are deposited into the repository in the single IP. Manifestations are designated in the UNT Libraries' SIP Specification by directories in the root of the /data/ directory in the BagIt bag with a notation expressing order and file types, e.g., 01_jpg, 01_tif, 01_pdf, or in the case of the example above, 01_jpg and 02_pdf. The numeric prefix helps designate the order of manifestation, with the lower number designating the priority for presentation to an end user in one of the Aubrey access systems; the text string denoting the file type is used to provide a cue as to the type of manifestation. This text string should be changed to reflect the needs of the manifestation, but users should strive for consistency over a large set of resources. Common text strings include jpg, tif, pdf, doc, data, raw, and jp2.

For each manifestation there is a section within the coda_directives.py file, which governs how it is processed during the SIP-to-AIP and AIP-to-ACP conversion steps. Additional information notes: if page numbers can be automatically extracted from the filenames because they use a common pattern, if the resource follows standard fileSet groupings based on the file-naming conventions, if the resource consists of preferred file formats, or if there are files in the resource that are not on the list of preferred formats for preservation. Finally, instructions are given for the AIP-to-ACP conversion, such as the need to create tiles for the resource, extra instructions as to the size or resolution of derivative images for the ACP, and finally if the manifestation should be included in the final ACP and exposed via one of the Aubrey access systems.

METS Record

A METS (Metadata Encoding and Transmission Standard) record conforming to the UNTL METS Profile[6] is included in both the Archival Information Package (AIP) and the Dissemination Information Package (DIP). This METS record contains metadata about all of the Files in the IP, the role of each File within a fileSet, the sequence of a fileSet within a manifestation, and information such as pagination labels, annotations, audio track names, or video clip names. The sequence of manifestations within the IP is also encoded in the METS record. For AIPs and DIPs there are additional preservation metadata files encoded within the METS record using the PREMIS Data Dictionary in the PREMIS Version 2.0 XML Scheme. The METS record also stores the output of the Unix *File* command for each File in the IP. Each File in the IP has an associated and referenced JHOVE output file that was created during the SIP-to-AIP conversion process and which is stored as part of the AIP and DIP format.

Requirements

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119[4]]

An implementation is not compliant if it fails to satisfy one or more of the MUST and REQUIRED level requirements for the protocols it implements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements for its protocols is said to be “unconditionally compliant”; one that satisfies all the MUST level requirements but not all the SHOULD level requirements for its protocols is said to be “conditionally compliant.”

UNTL-SIP Specification

A UNTL-SIP **MUST** be a fully valid BagIt bag that can be validated by standard BagIt software. The UNTL-SIP specification **MUST NOT** use the “holey bag” features as defined in the BagIt specification. All files that constitute the digital object **MUST** be stored in the /data/ folder. At the time of writing, the UNTL-SIP makes use of MD5 cryptographic hash functions for fixity; therefore a manifest-MD5.txt file **MUST** be used for the hash values for the files in the /data/ folder.

A valid NaMaste file called 0=untl_sip_1.0 **MUST** be in the root of the BagIt bag designating that the bag is a UNTL-SIP.

A valid coda_directives.py file is **REQUIRED** with appropriate processing instructions for the digital object.

A valid bag-info.txt file containing information about the digital object being deposited is **REQUIRED**. The oxum field **MUST** be present with an oxum value that represents the contents in the /data/ directory.

The contents of the /data/ directory **MUST** include Manifestation folders, and **SHOULD** include a metadata.xml file in the UNTL Metadata format, and an **OPTIONAL** note.txt file.

Care should be taken to remove hidden or system files (often invisible when processing on Windows, OSX, or Linux systems) as they will be considered part of the digital object if they are submitted with the SIP. If included, these files can cause the SIP-to-AIP conversion process to fail, or may be processed successfully into an AIP and later an ACP in an unintended manner.

Locally the UNT Libraries make use of a Python tool called SIPmaker.py, which helps to automate the creation and packaging of folders of content into valid UNTL-SIPs.

An example of the UNTL-SIP structure is in Appendix 3.

UNTL-AIP Specification

A UNTL-AIP **MUST** be a fully-valid BagIt bag that can be validated by standard BagIt software. The UNTL-AIP specification **MUST NOT** use the “holey bag” features as defined in the BagIt specification. All files that constitute the digital object **MUST** be stored in the /data/ folder. At the time of writing the UNTL-AIP makes use of MD5 cryptographic hash functions for fixity; therefore a manifest-MD5.txt file **MUST** be used for the hash values for the files in the /data/ folder.

A valid NaMaste file called 0=untl_aip_1.0 **MUST** be in the root of the BagIt bag designating that this bag is a UNTL-AIP.

A valid coda_directives.py file is **REQUIRED** with appropriate processing instructions for the digital object.

A valid bag-info.txt file containing information about the digital object being deposited is **REQUIRED**. The oxum field **MUST** be present with an oxum value that represents the contents in the /data/ directory.

Within the /data/ directory there **MUST** be only two directories: a /data/data/ directory containing the contents of the original UNTL-SIP as ingested into the system, and a /data/metadata/ directory containing metadata generated during the ingest process that is stored as part of the UNTL-AIP.

In addition to the /data/data/ and /data/metadata/ directories there is a **REQUIRED** METS metadata record with the filename of <NAME>.mets.xml where the value of <NAME> is the ARK Name assigned to the resource upon ingest into the UNT Libraries Digital Collections. This ARK Name, when appended to the UNT Libraries’ Name Assigning Authority M (NAAM), creates a full ARK identifier in the format of ark:/67531/<NAME>. An example of this is ark:/67531/metapth1234

A highly suggested but **OPTIONAL** metadata.xml file in the UNTL metadata format and an optional note.txt file can be stored in the /data/ directory.

No other files or folders than those mentioned above may be stored in the /data/ directory.

Locally the UNT Libraries make use of a Python tool called makeAIP.py, which automates the creation and packaging of a UNTL-SIP into a valid UNTL-AIP.

An example of the UNTL-AIP structure is in Appendix 4.

UNTL-DIP (ACP) Specification

Note: For historical reasons the UNT Libraries refers to its Dissemination Information Package (DIP) using the name Access Content Package (ACP). The specification of the UNTL-DIP is the same as that for the UNTL ACP and they are interchangeable for the purpose of mapping to the OAIS Reference Model.

A UNTL-DIP **MUST** be a fully-valid BagIt bag that can be validated by standard BagIt software. The UNTL-DIP specification **MUST NOT** use the "holey bag" features as defined in the BagIt specification. All files that constitute the digital object **MUST** be stored in the /data/ folder. At the time of writing the UNTL-DIP makes use of MD5 cryptographic hash functions for fixity; therefore a manifest-MD5.txt file **MUST** be used for the hash values for the files in the /data/ folder.

A valid NaMaste file called 0=untl_acp_1.0 **MUST** be in the root of the BagIt bag designating that this bag is a UNTL-DIP.

A valid coda_directives.py file is **REQUIRED** with appropriate processing instructions for the digital object.

A valid bag-info.txt file containing information about the digital object being deposited is **REQUIRED**. The oxum field **MUST** be present with an oxum value that represents the contents in the /data/ directory.

Within the /data/ directory there **MUST** be only one directory, a /data/web/ directory containing the dissemination or access versions of the originally deposited files that were generated from the UNTL-AIP.

In addition to the /data/data/ and /data/metadata/ directories there is a **REQUIRED** METS metadata record with the filename of <NAME>.mets.xml where the value of <NAME> is the ARK Name assigned to the resource upon ingest into the UNT Libraries' Digital Collections.

A **REQUIRED** UNTL metadata record in the UNTL metadata format (validates with the UNTL Metadata Schema[5]) with the filename of <NAME>.mets.xml where the value of <NAME> is the ARK Name assigned to the resource upon ingest into the Digital Collections.

No other files or folders than those mentioned above may be stored in the /data/ directory.

Locally the UNT Libraries make use of a Python tool called makeACP.py, which automates the creation and packaging of a UNTL-AIP into a valid UNTL-DIP (ACP).

An example of the UNTL-ACP structure is in Appendix 5.

References

1. http://www.iso.org/iso/catalogue_detail.htm?csnumber=57284
2. http://tools.ietf.org/html/draft-kunze-bagit
3. http://www.library.unt.edu/digital-projects-unit/digital-file-formats
4. https://www.ietf.org/rfc/rfc2119.txt
5. http://digital2.library.unt.edu/untl.xsd
6. http://www.loc.gov/standards/mets/profiles/00000045.xml

Appendix 1: Example coda_directives.py files

Example 1: coda_directives.py file for a single manifestation item.

```
"""
    these are directives on a per-manifestation basis
    they'll be indexed by pathname relative to the data
    directory
"""
#if certain filetypes need certain Jhove modules, put them here
#jhove_matcher = (
#                   (r".*\.log", "ASCII-hul"),
#                   (r".*\.cdx", "ASCII-hul"),
#                 )

manifestation_directives = \
{
    "01_tif":
    {
        #Does this manifestation use "magicknumbers" filenaming?
        #"use_magick_numbers" : True, # Default False

        #Does this manifestation follow the UNTL fileSet layout?
        #"untl_filesets" : False, # Default True

        #Does this manifestation need to be added to the
        #Access Content Package (ACP) or not added?
        #"add_to_acp" : False, # Default True

        #What label should be added for this manifestation?
        #"label" : "tif", #String used for manifestation label

        #What should the maximum size be for an image across the horizontal?
        #"max_width": 1200, #Default 1500

        #Do we want to create Zoomify tiles for this manifestation?
        #"make_tiles" : True, #Default False

        #Do we want to absolutely require thumbnails to be created for this manifestation?
        #"require_thumbnails" : False, #Default True
     }
}
```

Example 2: coda_directives.py file for a multi-manifestation item.

```
"""
    these are directives on a per-manifestation basis
    they'll be indexed by pathname relative to the data
    directory
"""
```

```
#if certain filetypes need certain Jhove modules, put them here
#jhove_matcher = (
#                   (r".*\.log", "ASCII-hul"),
#                   (r".*\.cdx", "ASCII-hul"),
#                 )

manifestation_directives = \
{
    "01_tif":
    {
        #Does this manifestation use "magicknumbers" filenaming?
        #"use_magick_numbers" : True, # Default False

        #Does this manifestation follow the UNTL fileSet layout?
        #"untl_filesets" : False, # Default True

        #Does this manifestation need to be added to the
        #Access Content Package (ACP) or not added?
        #"add_to_acp" : False, # Default True

        #What label should be added for this manifestation?
        #"label" : "tif", #String used for manifestation label

        #What should the maximum size be for an image across the horizontal?
        #"max_width": 1200, #Default 1500

        #Do we want to create Zoomify tiles for this manifestation?
        #"make_tiles" : True, #Default False

        #Do we want to absolutely require thumbnails to be created for this manifestation?
        #"require_thumbnails" : False, #Default True
     },
    "02_pdf":
    {
        #Does this manifestation use "magicknumbers" filenaming?
        #"use_magick_numbers" : True, # Default False

        #Does this manifestation follow the UNTL fileSet layout?
        #"untl_filesets" : False, # Default True

        #Does this manifestation need to be added to the
        #Access Content Package (ACP) or not added?
        #"add_to_acp" : False, # Default True

        #What label should be added for this manifestation?
        #"label" : "pdf", #String used for manifestation label

        #What should the maximum size be for an image across the horizontal?
        #"max_width": 1200, #Default 1500

        #Do we want to create Zoomify tiles for this manifestation?
        #"make_tiles" : True, #Default False

        #Do we want to absolutely require thumbnails to be created for this manifestation?
        #"require_thumbnails" : False, #Default True
     },
}
```

Appendix 2: Example bag-info.txt file

Example bag-info.txt file for an item.

```
Bag-Size: 156.09M
Bagging-Date: 2015-03-04
CODA-Ingest-Batch-Identifier: d246388c-c75a-465f-b2da-a44627ed9182
CODA-Ingest-Timestamp: 2015-03-04T21:06:16-0600
Contact-Email: mark.phillips@unt.edu
Contact-Name: Mark Phillips
Contact-Phone: 940-565-2415
External-Description: Newspaper issues from multiple titles funded through the
 support of grants awarded to individual partner libraries, historical
 societies, and genealogical societies. Content was digitized from microfilm
 using NDNP standards by iArchives and processed for inclusion in the Portal to
 Texas History. Master files are tiff images with accompanying OCR and bounding
 box files.
External-Identifier: ark:/67531/metapth592269
Internal-Sender-Identifier: 1947082201
Organization-Address: P. O. Box 305190, Denton, TX 76203-5190
Payload-Oxum: 163567298.34
Source-Organization: University of North Texas Libraries
```

Appendix 3: Example SIP structure.

Example SIP file structure for a single manifest item.

```
Giddings-Box1-Folder3-1871-02-ChappellHill-SM
├── 0=untl_sip_1.0
├── bag-info.txt
├── bagit.txt
├── coda_directives.py
├── data
│   ├── 01_tif
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.pro
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.pro.xml
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.tif
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.txt
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.pro
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.pro.xml
│   │   ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.tif
│   │   └── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.txt
│   └── metadata.xml
└── manifest-md5.txt
```

Appendix 4: Example AIP structure.

Example AIP/DIP file structure for a single manifest item.

```
metapth594985
├── 0=untl_aip_1.0
├── bag-info.txt
├── bagit.txt
├── coda_directives.py
├── data
│   ├── data
│   │   └── 01_tif
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.pro
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.pro.xml
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.tif
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.txt
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.pro
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.pro.xml
│   │       ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.tif
│   │       └── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.txt
│   ├── metadata
│   │   ├── 2e177905-cfda-48bf-9bdd-601e80639976.jhove.xml
│   │   ├── 392cb351-596e-4a4f-9f51-5f48a6254db4.jhove.xml
│   │   ├── 87c6b3ea-ae2f-45fb-9ef9-8f56922fbac5.jhove.xml
│   │   ├── 887d1234-9738-4694-9a2c-708c954d2d6a.jhove.xml
│   │   ├── 9011fc2f-d589-4a00-b83b-ad163315e578.jhove.xml
│   │   ├── 95f45549-43f0-462d-918e-7c20f27bc86e.jhove.xml
│   │   ├── a301686b-e8ec-41a2-a671-3eaa43a8b43f.jhove.xml
│   │   └── b6cdc593-0d9e-4c39-b41f-ad04024043ce.jhove.xml
│   ├── metadata.xml
│   └── metapth594985.aip.mets.xml
└── manifest-md5.txt
```

Appendix 5: Example DIP/ACP structure.

Example Dissemination Information Package (DIP) or Access Content Package (ACP) file structure for a single manifest item.

```
metapth594985/
├── 0=untl_acp_1.0
├── bag-info.txt
├── bagit.txt
├── coda_directives.py
├── data
│   ├── metapth594985.mets.xml
│   ├── metapth594985.untl.xml
│   └── web
│       └── 01_tif
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.med_res.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.pro.xml
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.square.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.thumbnail.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_01.txt
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.med_res.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.pro.xml
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.square.jpg
│           ├── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.thumbnail.jpg
│           └── Giddings-Box1-Folder3-1871-02-ChappellHill-SM_02.txt
└── manifest-md5.txt
```