Preservica

Information Package Structure Definition

v6.9.0

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark.

Table of Contents

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

References

Document	Ref	Date	Details & Issue
Preservica Guide to System Documentation	[DOC]	Sept 2022	git/doc/UG/SUG V6.6.0
Preservica Logical Data Model	[LDM]	See [DOC] for version information.	
Preservica Developer Guide	[DEV]		
Preservica XIP Schema Description	[SCHEMA]		
Preservica Standard Workflows	[SWF]		

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

Chapter 1. Introduction

1.1. Purpose of this Document

This document describes the structure of Submission Information Packages (SIPs) in the Preservica system.

The SIP Creator tool supplied with Preservica creates SIPs that conform to this structure. Preservica supports the ingest of SIPs in this format and a number of other package formats.

1.2. Context of this Issue

This document is consistent with Preservica 6.9.

1.3. Definition of Terms

Term	Definition
SIP	Submission Information Package. OAIS definition: “an Information Package that is delivered by the Producer to the OAIS for use in the construction of one or more AIPs”.
AIP	Archival Information Package. OAIS definition: “an Information Package consisting of the Content Information and the associated Preservation Description Information (PDI), which is preserved within an OAIS”.
UUID	Universally Unique Identifier. A 128-bit number, represented in string form as 32 hexadecimal digits, displayed in 5 groups separated by hyphens, in the form 8-4-4-4-12, for a total of 36 characters (32 digits and 4 hyphens).
XIP	The metadata schema used with the Preservica system.

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

Chapter 2. Submission Information Package Structure

2.1. Physical Structure

```
'/UUID'
        /content
                '/file hierarchy'
                ...
        metadata.xml
'UUID'.protocol
```

Every SIP has exactly one root or top-level directory. The root directory contains a subdirectory named content, and the associated metadata in a file called metadata.xml. No other files should be placed directly under the root level.

The content subdirectory contains all the physical files that make up the SIP; any arbitrary directory / file hierarchy is allowed within the content subdirectory.

The SIP Creator tool creates an additional file at the same level as the SIP root directory. This XML file has the same name as the root directory (typically a UUID), and the extension *.protocol*. The protocol file is written last by the SIP Creator to indicate that the SIP is now complete on disk.

The physical structure of an example SIP is as follows:

```
/0849651a-d5d0-460b-ba15-d0c784894d22
        /content
                /reports
                        report01.pdf
                        report02.pdf
                /images
                        image01.jpg
                        image02.jpg
        metadata.xml
0849651a-d5d0-460b-ba15-d0c784894d22.protocol
```

A SIP may also be a ZIP file containing the directory. In this case no protocol file is needed.

2.1.1. Protocol file format

The format of the protocol file written by the SIP Creator tool is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<protocol xmlns="http://www.tessella.com/xipcreateprotocol/v1">
    <dateCreated>SIP creation date</dateCreated>
    <size>Size of SIP in bytes</size>
    <files>Number of directories and files contained in SIP</files>
    <submissionName>Collection name from metadata</submissionName>
    <catalogueName>Collection code from metadata</catalogueName>
    <localAIP>UUID of SIP</localAIP>
    <globalAIP>UUID of AccessionRef from metadata</globalAIP>
    <createdBy>Creator of SIP</createdBy>
</protocol>
```


Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

The protocol file corresponding to the example physical structure in section 2.1 above is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<protocol xmlns="http://www.tessella.com/xipcreateprotocol/v1">
    <dateCreated>2009-12-15T10:10:11.099Z</dateCreated>
    <size>2594665</size>
    <files>6</files>
    <submissionName>Example Collection</submissionName>
    <catalogueName>EXC</catalogueName>
    <localAIP>0849651a-d5d0-460b-ba15-d0c784894d22</localAIP>
    <globalAIP>37a90437-798b-4cdd-8a1c-39687adc2ff3</globalAIP>
    <createdBy>Test User</createdBy>
</protocol>
```

2.2. SIP Metadata

The metadata for the SIP, contained in the single file metadata.xml, must validate against the XIP schema definition used within Preservica. The XIP v6 Schema, including an example XIP document for ingest, is described in [LDM]. (You can find documentation about the XIP v4 schema in documentation from earlier versions of Preservica.)

You can use either XIP v6 or XIP v4 in your SIP. If you use XIP v4, then the v4 objects (Collections, DeliverableUnits etc) will undergo a transformation into v6 entities (StructuralObject, InformationObject etc) similar to that for existing data described in the Migration to V6 document.

As well as conforming to the XIP schema definition, there are additional requirements for the SIP metadata, which will be detailed in this section.

As described in [SCHEMA], all of the top level entities in the XIP schema are optional. Whilst there may be situations where each of these could be omitted, these are not general cases and as such, for SIPs only some of these should be considered optional. The entities that must be included, and the properties that must be set on each, are listed below, grouped by their top level entity.

2.2.1. StructuralObject

If you want to include any structure in your package, you will need to include *StructuralObject* elements. If you don't specify a *Parent* subelement, then they will be added at the top level.

2.2.2. InformationObject

If you want any assets in your package, you will need to include *InformationObject* elements. The *Parent* subelement should be set to the ref of a structural object, either defined in the package or already present in the system.

2.2.3. Objects relating to Content

If you want to ingest any content files into Preservica, you will also need to specify the rest of the logical model:

- *Representation* elements. These should refer to an information object in the package in their *InformationObject* subelement, and have a list of *ContentObject* references to content objects in the package. (The COs will be later in the package documents.)
- *ContentObject* elements. The parent of a content object should be an information object in the package, and that information object should be the same one that the representation referring to this CO is related to.

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

- *Generation* elements. You don't need to provide format and property information; this will be filled in by characterisation when you ingest the package. Generations should include *Bitstreams* in a *Bitstream* container, referring to a *Bitstream* by its file path (*PhysicalLocation* and *Filename*). For example if a *Bitstream* in the XIP has a *PhysicalLocation* of / and a *Filename* of *myfile.jpg*, the *Bitstream* in the generation should be */myfile.jpg*. If the *PhysicalLocation* were *subdir*, the reference should be *subdir/myfile.jpg*.
- *Bitstream* elements. The *PhysicalLocation* should be the directory, relative to the package's content directory, where the file actually resides, and the *Filename* should be the actual name of the file on disk (or in the ZIP file). The *FileSize* should match the actual size of the file. Unless you configure your ingest workflow errors to ignore not having them (VLDTN_58), you also need to specify at least one *Fixity* value inside the *Fixities* container element.

2.2.4. Descriptive Metadata

If you want any descriptive metadata fragments attached to entities in the package, you need to add *Metadata* elements at the top level of the XIP.

2.2.5. Additional Constraints

Constraint	Description
Maximum SIP size	Preservica 6.9 ingest has been tested against SIPs up to 100Gb in size.
Maximum number of files in a single SIP	Preservica 6.9 ingest has been tested against SIPs containing up to 100,000 files.
Maximum individual file size	Preservica 6.9 ingest has been tested against SIPs containing individual files of up to 30GB.
Allowed character sets for directory and file names	Preservica 6.9 has been successfully tested ingesting SIPs that include directories and files containing extended characters (i.e. UTF-8 characters greater than code point 128) in their names. However, if it is likely that files will be transferred between different operating systems, to avoid potential false conversions of filenames it is recommended that directory and file names in a SIP be composed of a restricted set of characters. The recommended set of characters is a subset of US-ASCII; these are the same characters as found in the first 128 characters of ISO-8859 and Unicode (UTF-8) character sets. The recommended characters are: * Letters: A-Z, a-z * Digits: 0-9 * Other: ! # $ % ( ) + , - . = @ [ ] { } ~
Maximum path length for individual files	Preservica 4 ingest has been tested against SIPs containing files with path lengths of up to 2048 characters.

2.2.6. SIPs in Ingest Workflows

Preservica supports the ingest of SIPs in the XIP format and a number of other package formats. Where SIPs are supplied in XIP format Preservica will validate the the package conforms to the specification provided in this document and the referenced XIP Schema documentation [SCHEMA].

The *Import from Transfer Area* step detects the .protocol file associated with a SIP, and then copies the SIP root directory to the step's working area. (Note that the .protocol file itself is not copied.) The remaining standard ingest steps all follow the convention for parameters as fully described in [DEV]; the output metadata file of a workflow step becomes the input metadata file of the next step. Each step is also given as a parameter the absolute path of the content directory of the SIP, as copied to the working area.

Details of the other ingest package formats supported and any associated restrictions are detailed in the Preservica Standard Workflows document [SWF].

Copyright Preservica 2022. All rights reserved. Preservica is a registered trademark

www.preservica.com

Reference: git/doc/ADD/IPS | Issue: 6.9.0
Copyright Preservica 2022 | All rights reserved | Preservica is a registered trademark

Preservica, 32 The Quadrant, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YS, UK info@preservica.com - preservica.com