

**Government of Sultanate of Oman**

**National Records and Archives Authority**

**(NRAA)**

**Specifications**

**for**

**Oman Submission Information Package (OSIP)**

**An Interface Between EDRMS**

**and**

**Digital Preservation Systems**

**Electronic Document and Records Management System (EDRMS)**

# Document History and Revision Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. |  | Date | Description of Changes | Author |
| 1 |  | 20200823 | First released version | Yahya Yai Abdullah, Expert in Electronic Records Management & EDRMS Project Manager. |
| 2 |  | 20210208 | Second released version | Yahya Yai Abdullah, Expert in Electronic Records Management & EDRMS Project Manager. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[Document History and Revision Control 3](#_Toc63780593)

[1. Executive Summary 6](#_Toc63780594)

[2. Objectives 6](#_Toc63780595)

[3. Audience 6](#_Toc63780596)

[4. Acknowledgements 8](#_Toc63780597)

[5. Terms and Definitions 9](#_Toc63780598)

[6. Notation 9](#_Toc63780599)

[7. Overview 10](#_Toc63780600)

[7.1. What is a SIP? 10](#_Toc63780601)

[7.2. Digital Preservation Systems 10](#_Toc63780602)

[7.3. EDRMS 10](#_Toc63780603)

[8. Package 10](#_Toc63780604)

[8.1. Structure 10](#_Toc63780605)

[8.2. Metadata (Header) 10](#_Toc63780606)

[8.3. Primary Data (Content) 10](#_Toc63780607)

[8.4. Naming 11](#_Toc63780608)

[8.5. Container format and compression 11](#_Toc63780609)

[8.6. Size Limits 11](#_Toc63780610)

[8.7. Maximum Path Lengths 12](#_Toc63780611)

[8.8. Formats and Character Sets 12](#_Toc63780612)

[9. Primary Data 12](#_Toc63780613)

[9.1. Folders and Subfolders 12](#_Toc63780614)

[9.2. Computer Files (representing Digital Objects) 13](#_Toc63780615)

[10. Metadata 14](#_Toc63780616)

[10.1. Table of Contents (<toc>) 14](#_Toc63780617)

[10.1.1. Folder (<folder>) 14](#_Toc63780618)

[10.1.2. Computer Files (<digitalObject>) 15](#_Toc63780619)

[10.2. Submission (submission) 17](#_Toc63780620)

[10.2.1. Classification System 19](#_Toc63780621)

[10.2.2. Classification Level 20](#_Toc63780622)

[10.2.3. File 20](#_Toc63780623)

[10.2.4. File Volume 23](#_Toc63780624)

[10.2.5. Records 25](#_Toc63780625)

[11. Sample SIP 30](#_Toc63780626)

**Table of figures**

[Figure 1: Terms and Definitions 9](#_Toc63780627)

# Executive Summary

The Oman Submission Information Package (OSIP) specifies an open and neutral standard format that must be implemented by all systems that keep and manage official records in digital form in order to produce valid and complete digital information packages of those records that are to be transferred to the digital preservation system of NRAA for permanent preservation according to Omani law and regulations.

The OSIP specifications describe the structure and order of such packages, including the digital content and information that must be present in these packages for it to be considered valid and complete both in the technical and business sense.

OSIP itself is designed with reference to the Open Archival Information Systems (OAIS) reference model accepted and used worldwide by various national, state and city archives, libraries, and other corporate memory preservation institutions, including large multi-national corporations. OSIP is also based on the Swiss E-Government Standards eCH-0160 with modifications and translations.

However, OSIP also takes into account the specific characteristics and requirements of preservation of records that are inherent in the standards, practices and culture of Oman as well as the surrounding region of Arabic-speaking countries.

# Objectives

The objectives of NRAA in producing a localized specification for the Submission Information Package (SIP) is to specify a single open and standard format for packaging of digital records along with their associated information necessary for its transfer and ingestion into NRAA’s digital preservation system for permanent preservation, or into any other digital preservation system certified for use by NRAA in Oman.

This format is designed to be simple and easy enough for the various EDRMS vendors and solution providers to understand, in order that they may be able to develop and maintain a compliant implementation of it in their EDRMS products and solutions. As such this format is designed to be vendor and platform neutral.

It is also designed to be easily maintained and evolved over time by NRAA to meet any future requirements of digital preservation of digital records and archives.

It is the intention of NRAA that OSIP be promoted and accepted as the open standard for SIP in Oman, in the Gulf Cooperation Council (GCC) countries, the East African countries and in the Arabic-speaking world in general.

# Audience

This document is meant for the following parties:

1. Government agencies , government-owned corporations and public utilities who are required under Omani law and regulations to transfer their records to NRAA as permanent archives. This document provides a tool for them to facilitate communication with their in-house software development team or external software vendors to ensure compliance with OSIP specifications.

1. Implementors of digital preservation systems who wish to understand the structure of OSIP packages in order to validate it, transform it, process it, and ultimately ingest them into those systems.
2. Implementors of electronic records management systems and line-of-business systems that manages records, who will need produce valid OSIP packages from the records it manages, for transfer and ingestion into digital preservation systems. The ability to produce valid OSIP packages will henceforth be considered mandatory for compliance with NRAA’s national standards and guidelines for management of electronic document and records in the Oman government offices and in other entities where compliance is mandated by law.
3. Implementors of standalone SIP creation tools that will allow the manual creation of OSIP packages from records that are not managed using electronic records management systems or line-of-business systems, or they are managed using legacy systems that are no longer supported by their original vendors.

# Acknowledgements

NRAA is indebted to the authors and creators of the *eCH-0160 v1.0 Archival Submission Interface* standards of the Swiss Federal Archives (SFA) for their original work on the SIP standards and allowing NRAA to fork eCH-0160 and customize it to be the current OSIP standard.

The eCH-0160 v1.0’s original authors are:

Digital archiving expert group

Marguérite Bos, Swiss Federal Archives,  
[marguerite.bos@bar.admin.ch](mailto:marguerite.bos@bar.admin.ch)

Georg Büchler, KOST  
[georg.buechler@kost.admin.ch](mailto:georg.buechler@kost.admin.ch)

Anouk Dunant Gonzenbach, Geneva State Archives  
[anouk.dunant-gonzenbach@etat.ge.ch](mailto:anouk.dunant-gonzenbach@etat.ge.ch)

David Gubler, Fachlabor Gubler AG  
[d.gubler@swiss-mikrosave.ch](mailto:d.gubler@swiss-mikrosave.ch)

Martin Kaiser, KOST  
[martin.kaiser@kost.admin.ch](mailto:martin.kaiser@kost.admin.ch)

Markus Lischer, Lucerne State Archives  
[markus.lischer@lu.ch](mailto:markus.lischer@lu.ch)

Nicole Martini, Swiss Federal Archives  
[nicole.martini@bar.admin.ch](mailto:nicole.martini@bar.admin.ch)

Urs Meyer, Swiss Federal Archives  
[urs.meyer@bar.admin.ch](mailto:urs.meyer@bar.admin.ch)

Ursina Rodenkirch-Brändli, Graubünden State Archives  
[ursina.rodenkirch@sag.gr.ch](mailto:ursina.rodenkirch@sag.gr.ch)

Stefan Ryter, Bern State Archives  
[stefan.ryter@sta.be.ch](mailto:stefan.ryter@sta.be.ch)

We also appreciate the work of Mr. Felix Akeret in providing his consultancy to NRAA in identifying the most appropriate of the various international SIP standards to adopt and to customize, and for his efforts in translating the original eCH-0160 specifications from German into English, and providing the first draft of the translation of the eCH-0160 XML / XSD from German into English.

The authors and current maintainers of the OSIP v1.0 standard are:

1. Yahya Yai Abdullah

Expert in Electronic Records Management &

Project Manager of the National Wosool (EDRMS) Project,

National Records and Archives Authority, Sultanate of Oman

[yahya.yai@nraa.gov.om](mailto:yahya.yai@nraa.gov.om)

1. Hamood Suliman Ali Al-Fahdi

Assistant Director,

Information Technology Department,

National Records and Archives Authority, Sultanate of Oman

[hamood@nraa.gov.om](mailto:hamood@nraa.gov.om)

1. Marya Said Nasser Al-Siyabi

Software Analyst (Electronic Records Specialist),

National Records and Archives Authority, Sultanate of Oman

[marya@nraa.gov.om](mailto:marya@nraa.gov.om)

The authors of OSIP has reviewed, modified and finalized the English translation of the German language XML element and attribute names and enumerated values. They have also removed certain elements specific to the Swiss context (e.g. “GEVER”) and added new structures and elements specific to the Omani context e.g. support for file volumes, relationship links, additional point-of-capture metadata for files and records.

All these changes meant that OSIP is no longer compatible with the eCH-0160, and the intention is to maintain it as a separate standard moving forward.

# Terms and Definitions

The following terms, abbreviations and acronyms used in this document have the following definitions.

<TBD>

Figure 1: Terms and Definitions

Terms used for the XML elements, attributes and enumerated values inside the metadata of the SIP are described in the section on the Metadata below.

# Notation

All folder, subfolder and computer file names in this document will be formatted in a brown Courier font.

Numbering and naming patterns will be formatted in a bold purple Courier font and in non-italics like so:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **f** | **n** | **n** | **n** | **n** | **n** | **n** |

The following symbols are used in this document for the notation of folders and files:

|  |  |
| --- | --- |
| *Symbols* | *Description* |
| / | Root folder. This is the folder at the top of a hierarchy of folders in the SIP package |
| header/ | A folder with the name “header” |
| d000024.pdf | A computer file with the name “d000024” and the file extension “.pdf”. |
| … | A placeholder indicating that there are more items to follow which is not explicitly shown. |

# Overview

## What is a SIP?

## Digital Preservation Systems

## EDRMS

# Package

## Structure

An OSIP package consists of multiple subfolders and computer files (representing digital objects) contained within a single root (top-level) folder.

The root folder must contain only two subfolders - header and content – and no other folders or files besides these two.

## Metadata (Header)

The header subfolder within the root folder can only contain two computer files – metadata.xml and metadata.xsd. All the metadata relating to all the records in this package are contained within the metadata.xml file. The metadata.xsd is an XML Schema file used to validate the correctness of the metadata.xml file.

No other files are allowed to be inside the header subfolder.

*The* metadata.xml *file will be described in more detail below in section ???.*

## Primary Data (Content)

The content subfolder within the root folder can only contain folder, subfolders and the computer files of the digital objects for all the records referenced in the metadata.xml file.

Each folder here corresponds to a (records) File and every subfolder corresponds to a File Volume mentioned in the metadata.xml file. The digital objects for all records within a File and File Volume must therefore be found in the same folder and subfolder representing that File and File Volume.

For systems that support only File and do not support File Volumes, then the content subfolder will only contain one level of folders only containing computer files for digital objects.

<Snippet of xml and the corresponding folder structure in the content folder>

The specifications for the folders, subfolders and the computer files contained within them are further defined in section ??? below.

## Naming

The root folder must be named according to this naming scheme:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fixed identifier | Separator (underscore) | Submission Date | Separator (underscore) | Code of submitting agency | Separator (underscore) | Accession Number |
| SIP | \_ | YYYYMMDD | \_ | XXXXX | \_ | YYYY\_NNN |

For example, a submission from the Ministry of Finance (MOF) on the date 1st January 2023 with accession number 2022\_003 (provided by NRAA to the Ministry) will have a root folder named:

SIP\_20231122\_MOF\_2022\_003

In the case where the package is compressed and / or contained within a single container format e.g. ZIP, then the name of the container must be the name of the root folder with the file extension indicating the container format.

Example:

SIP\_20231122\_MOF\_2022\_003.zip

## Container format and compression

The OSIP package can be compressed and packed into a single container ZIP format for ease of handling and faster transfer especially over online channels. Container formats other than ZIP must be negotiated prior to the submission between the submitting agency and NRAA.

Although this ZIP format approach is highly recommended, there may be situations where the OSIP package should not be compressed into a single artifact e.g. when the package contains audio and video files that are too large to be supported by compression formats like ZIP.

In such cases, the OSIP package – the root folder and all its contents – can be copied as is without compression or using any container format into a transfer media such as an external disk or portable flash-based storage.

## Size Limits

The size for an OSIP package must not exceed 8 GB.

For faster handling, transfer and ingest, it is highly recommended that the size of the OSIP package not exceed 2 GB. This can be achieved by distributing the digital records to be transferred by the submitting agency over multiple packages through proper prior planning and prior agreement with NRAA.

OSIP packages over 8 GB in size can be only submitted in very exceptional cases. The submitting agency must contact NRAA and provide cogent reasons for doing so before creating such large OSIP packages.

A OSIP package must not contain more than 1,000,000 computer files.

A single folder in OSIP must not contain more than 5,000 computer files. If a single folder does contain more than that number of computer files, it is required to use subfolders to reduce the number of computer files per folder.

<Example>

## Maximum Path Lengths

All paths to each folder and computer file in the package must not exceed 250 characters in length (including the / file separator symbol ), and the paths include the root folder’s name as well.

<example>

## Formats and Character Sets

The XML and XSD files in the SIP must be in “UTF-8” encoding to cater for the use of Arabic names and descriptions.

The formats of the digital objects must be the original formats stored in the EDRMS system from which the records are extracted in order to create the SIP.

# Primary Data

Primary data are all the folders and subfolders, and computer files that are found under the /content folder of OSIP package. They represent the actual digital content to be transferred to NRAA as archives.

## Folders and Subfolders

Every folder representing a (records) File or File Volume must be named according to this scheme below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **f** | **n** | **n** | **n** | **n** | **n** | **n** |

**f**: is a fixed constant character representing a (f)older.

**n**: represents a single digit numeric value from 0 – 9.

Example: f000035

The entire folder name must be unique in the entire package. The number following the f constant character will be assigned automatically and serially starting with f000001 followed by f000002, and then f000003 and so on until the maximum value of f999999. Of course then this means that an OSIP package can only contain up to 999,999 folders and subfolders.

This name will be used as a reference id in the metadata.xml file whenever Files or File Volumes are referenced. Example:

<example>

Only File objects and File Volume objects should represented as folders and subfolders under the content folder. Classification system and classification levels should not be represented in this way, and should only be specified in the metadata.xml file.

## Computer Files (representing Digital Objects)

Every computer file[[1]](#footnote-1) in the OSIP package representing digital object must be named according to this scheme below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **d** | **n** | **n** | **n** | **n** | **n** | **n** | **.** | **extension** |

**d**: is a fixed constant character representing a (d)igital object.

**n**: represents a single digit numeric value from 0 – 9.

**extension**: is the computer file’s original file extension e.g. docx, jpg, pdf.

Example: d000982.pdf

The entire computer file name for digital objects must be unique in the entire package. The number following the d constant character will be assigned automatically and serially starting with d000001 followed by d000002, and then d000003 and so on until the maximum value of d999999. Of course then this means that an OSIP package can only contain up to 999,999 computer files representing digital objects.

This name will be used as a reference id in the metadata.xml file whenever digital objects are referenced. The original name for the digital objects are preserved as metadata in the metadata.xml file.

|  |
| --- |
| ***Note***  The reason why the digital objects in OSIP must be renamed according to this scheme and the need to preserve the original filename in metadata is to avoid problems such as:   * duplicate computer file names for what are essentially different digital content in the same folder or subfolder * Differences in the support for certain characters in different technology platforms for computer file names * Lack of support for certain character sets on different platforms * Differences in technology platforms for the maximum length of computer file names |

Example:

<example>

The computer files representing the digital objects must not be encrypted or password-protected in any way. If these computer files were stored in encrypted form in their source systems, then they must be decrypted first before being placed into the package.

If the package requires enhanced protection during its transfer to NRAA, then the entire package as a whole or the media in which it is transported must be encrypted using a single encryption key which must be sent to NRAA through a different channel than that which is used to transfer the package. However, this must be agreed beforehand between the submitting agency and NRAA.

# Metadata

The metadata are all the information about the digital content and the OSIP package itself. The OSIP package is self-describing – the information in the metadata.xml file describes exactly what folders and computer files can be found inside the OSIP package. All this textual information inside the metadata.xml must be encoded in Unicode.

The root element in the metadata.xml is the <package> element. This element has an attribute schemaVersion that specifies the version of the OSIP schema which is 1.0 at the time of this document. There may be enhanced OSIP schemas in the future where the version number will be higher. The <package> element also has 3 child elements: <packageType>, <toc>, and <submission>. <packageType> element value will always be “SIP” for the current specifications. In the future, OSIP may support other package types. <toc> and <submission> will be described in detail in the following sub-sections.

## Table of Contents (<toc>)

The <toc> element represents the table of contents for the package. This is where every folder, sub-folder and computer file inside the OSIP package is described in the metadata.xml file. The computer files are represented by the <digitalObject> elements with the <folder> elements. A hierarchy of folders and sub-folders within the package are represented by nesting <folder> elements within other <folder> elements.

Every <folder> element and <digitalObject> element has the <name> and <originalName> child elements. This is to allow for automated renaming of folders and computer files by the software that generates the OSIP package[[2]](#footnote-2) where the original name of the folder or computer file is preserved in metadata.

## Folder (<folder>)

The <folder> element represents an actual folder in the package. This element can nest within itself in order to represent a tree structure of folders and sub-folders.

Each <folder> will have a child <name> element and <originalName> element. For <folder> elements that correspond to the <file> and <fileVolume> elements described under the <submission> section of the SIP, the <folder>’s <name> value must correspond to the id attribute of the <file> and <fileVolume> elements.

|  |  |
| --- | --- |
| <name> | Mandatory. Text value.  This value is either header or content for the two fixed folders that must be at the root of the SIP package, or it must be the auto-generated name for the sub-folders under the root content folder.  Example of fixed root folder name: header or content  Example of auto-generated sub-folder name: f000002 |
| <originalName> | Mandatory. Text value.  The is the original computer folder name for the digital object.  It must be the same value as for the <name> element  Example of fixed root folder name: header or content  Example of auto-generated sub-folder name: f000002 |
| <digitalObject> | A <folder> element may contain either child <folder> elements (to mimic the hierarchy of <file> and <fileVolume> tree), or it may contain <digitalObject> elements.  <digitalObject> elements are described in the next section below. |
| <folder> | A <folder> element may contain either child <folder> elements (to mimic the hierarchy of <file> and <fileVolume> tree), or it may contain <digitalObject> elements.  <digitalObject> elements are described in the next section below. |

## Computer Files (<digitalObject>)

The <digitalObject> element represents an actual computer file within the package.

<example>

Each <digitalObject> will have the following attributes:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Mandatory. Text value.  An identifier that is unique for each digital object in this SIP. The identifier need only be unique within the SIP it is contained within. This identifier should be auto-generated according to the following numbering pattern:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **d** | **n** | **n** | **n** | **n** | **n** | **n** |   **d**: is a fixed constant string representing a digital object.  **n**: represents a single digit numeric value from 0 – 9.  Example: d000016    This identifier must match the auto-generated filename of the computer files for the corresponding digital object (see section 9.2 above). |

Each <digitalObject> will have the following child elements:

|  |  |
| --- | --- |
| <name> | Mandatory. Text value.  This is the auto-generated name for the computer file that this element represents. See section 9.2 above for how the computer file name should be auto-generated.  This value must match the <digitalObject>’s id attribute with the difference being that this value will include the actual extension of the computer file.  For example, if the id is d000001, then the auto-generated name value must be d000001.xml (if the extension of the computer file is .xml).  Example: d000001.xml |
| <originalName> | Mandatory. Text value.  The is the original computer filename for the digital object.  Example: acceptance-invite.xml |
| <checksumAlgorithm> | Mandatory. Text value.  The value is the name of the checksumAlgorithm used in computing the checksum.  Example: MD5  Example: SHA1 |
| <checksum> | Mandatory. Text value.  The <checksum> element’s value is the computed checksum of the computer file using the algorithm specified in the <checksumAlgorithm> element.  Example: 627f1e1e819df185cc9d9788852cdacb |

## Submission (submission)

The <submission> element contains information that describes the files and records that are transferred in this package include the business classification levels that they each belong under.

The <submission> element has the following child elements:

|  |  |
| --- | --- |
| <submissionType> | Mandatory. Text value.  In this current version of the OSIP specifications, the value of this element should always be “EDRMS”. |
| <submittingOrganization> | Mandatory. Text value.  The full name of the agency that is submitting this package to NRAA. |
| <submissionNumber> | Mandatory. Text value.  The accession number given by NRAA to the agency in the approval letter NRAA will issue to the agency in response to the Form <form number> submitted by the agency containing the list of files that is to be transferred in this package.  Every submission must have a unique submission number that will checked against the accession numbers previously issued by NRAA. An accession number cannot be re-used for more than one unique submission.  <Example value> |
| <transferApprovalReference> | Mandatory. Text value.  The reference number of NRAA’s approval letter authorizing the application to transfer previously submitted by the agency. |
| <creationTimePeriod> | Mandatory.  This element contains two mandatory child elements whose values are dates with the format yyyy-mm-dd (example: a value representing 1st of January 2016 will be represented as 2016-01-01):   |  |  | | --- | --- | | <from> | Mandatory. Date value.  The registered date of the earliest record of any file in this package. | | <until> | Mandatory. Date value.  The registered date of the latest record of any file in this package. | |
| <protectionPeriodCategory> | Optional. Text value.  Reference to an article of law or regulation that imposes a period of time (“closure period”) before which the transferred records should not be made accessible to the public by NRAA. This would be one of either Article 27 or Article 28 or Article 29 of the Records and Archives Law (promulgated by the Royal Decree No. 60/2007)  <Example value> |
| <protectionPeriod> | Optional only if <protectionPeriodCategory> has no value. Integer value.  The number of years before which the transferred records should not be made accessible to the public by NRAA. |
| <protectionPeriodArguments> | Optional. Text value.  A description of what records and why they require the protection period. |
| <provenance> | Mandatory.  This element must contain the following child elements:   |  |  | | --- | --- | | <creatorName> | Mandatory. Text value.  The full name of the agency who created and is submitting this SIP. | | <systemName> | Mandatory if the SIP is generated by an EDRMS system or business system instead of by hand. Text value.  The full name and version number of the EDRMS system or business system that generated this SIP. | | <systemDescription> | Mandatory if the SIP is generated by an EDRMS system or business system instead of by hand. Text value. A short description of the EDRMS system or business system that generated this SIP. | | <systemRelated> | Optional. Text value.  Systems that have exchanged data with the system described and are therefore subsystems, parallel systems or superordinate systems. The designations of the systems and the nature of the relationship are entered here. | |
| <classificationSystem> | Mandatory.  This will be described in more detail in the next sub-section. |

## Classification System

The <classificationSystem> element represents the multi-level, hierarchical structure of a business classification scheme, or file plan under which files are classified. Files and their contained records that comes from an EDRMS compliant system will already be classified under such a system, and such information would already be available for that system to generate the SIP package. For records that come from non-EDRMS systems lacking such a structured classification plan (e.g. records kept in shared network drives), then the creator of the SIP package will have to create a simple classification system under which those folders and records will be classified.

It contains the following direct child elements:

|  |  |
| --- | --- |
| <name> | Mandatory. Text value.  This is the name of the classification system.  <example> |
| <classificationSystemVersion> | Mandatory. Text value.  Classification systems may be updated or changed over time, and therefore should be tracked using a version number. This version number for the classification system under which the files in the package belong, must be provided here. |
| <classificationLevel> | Mandatory.  A <classificationSystem> element may contain multiple <classificationLevel> child elements.  Further described in detail in the sub-section below. |

## Classification Level

A <classificationLevel> element represents a single class or level within a business classification scheme.

Each <classificationLevel> element may contain these following **attributes**:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Mandatory. Text value.  An identifier that is unique for each <classificationLevel> in this SIP. The identifier need only be unique within the SIP it is contained within. This identifier should be auto-generated according to the following numbering pattern:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **c** | **n** | **n** | **n** | **n** | **n** | **n** |   **c**: is a fixed constant string representing a classification level.  **n**: represents a single digit numeric value from 0 – 9.  Example: c000012 |
| levelNumber | Mandatory. Text value.  An identifier that is unique for each <classificationLevel>. It is the business identifier given to each classification level in the business classification scheme to which the files in the SIP belong.  Example: 1100  Example: 1233/5 |

Each <classificationLevel> element may contain these following child **elements**:

|  |  |
| --- | --- |
| <title> | Mandatory. Text value.  The full title of the classification level |
| <classificationLevel> | Optional.  Any <classificationLevel> may or may not have child <classificationLevel> elements. This is dependent on the structure of the business classification scheme. |
| <file> | Mandatory for <classificationLevel> elements that do not contain other <classificationLevel> elements.  Further described in detail in the sub-section below. |

## File

A <file> element represents a logical File (i.e. aggregation of records like a dossier or folder, not a single computer file) that represents a unique business activity for which the contained records represent the individual transactions under that business activity.

Each <file> has the following attributes:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Mandatory. Text value.  An identifier that is unique across all <file>s and <fileVolume>s in this SIP (<fileVolume> will be described further below). The identifier need only be unique within the SIP it is contained within. This identifier should be auto-generated according to the following numbering pattern:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **f** | **n** | **n** | **n** | **n** | **n** | **n** |   **f**: is a fixed constant string representing a file or file volume.  **n**: represents a single digit numeric value from 0 – 9.  Example: f000012 |
| fileNumber | Mandatory. Text value.  An identifier that is unique for each <file>. It is a business identifier given to each file.  Example: 1234/2016-16 |

Each <file> has the following child elements:

|  |  |
| --- | --- |
| <title> | Mandatory. Text value.  The full title of the file. |
| <creationTimePeriod> | Mandatory.  This element contains two mandatory child elements whose values are dates with the format yyyy-mm-dd (example: a value representing 1st of January 2016 will be represented as 2016-01-01):   |  |  | | --- | --- | | <from> | Mandatory. Date value.  The registered date of the earliest record within any file volumes in this particular file. | | <until> | Mandatory. Date value.  The registered date of the latest record within any file volumes in this particular file. | |
| <securityLevel> | Mandatory. Text value.  A single character code representing the security level of the file. The value for this code must be one of the following:   |  |  | | --- | --- | | U | Unclassified | | C | Confidential | | R | Restricted | | S | Secret | | T | Top Secret |   Note: a file may contain records of different security levels, but the security level of each child record cannot be greater than the security level of the file itself. |
| <organisationUnitResponsible> | Mandatory. Text value.  The organization unit that owns or is responsible for maintaining this file. |
| <description> | Optional. Text value.  A description of the unique business activity that is represented by the file. |
| <retentionSeries> | Mandatory.  This element represents the specific rule (series) in the retention policy that is applied to this file.  This element has the following child elements:   |  |  | | --- | --- | | <retentionSeriesNumber> | Mandatory. Text value.  The number of the retention series that is applicable to this file. | | <retentionSeriesTitle> | Mandatory. Text value.  The title of the retention series that is applicable to this file. | |
| <formOfAppearance> | Mandatory. Text value.  This specified whether the file contains records that are purely digital or non-digital or a mix of both.  The value must be one of the following:   |  |  | | --- | --- | | unspecified | Not specified | | digital | The file is purely digital | | non-digital | The file is purely non-digital | | mixed | The file contains a mix of records that are digital and non-digital i.e. hybrid file. | |  |  | |
| <fileVolume> | A <file> element must contain at least one <fileVolume> child element.  The <fileVolume> element is described in more detail in the following section below. |
|  |  |

## File Volume

A file may consists of multiple volumes (especially if the file is originally a non-digital or hybrid file that has been subsequently digitized).

The <fileVolume> element represents the information for each of such file volumes.

Each <fileVolume> element will contain the following attributes:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Mandatory. Text value.  An identifier that is unique across all <file>s and <fileVolume>s in this SIP. The identifier need only be unique within the SIP it is contained within. This identifier should be auto-generated according to the following numbering pattern:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **f** | **n** | **n** | **n** | **n** | **n** | **n** |   **f**: is a fixed constant string representing a file or file volume.  **n**: represents a single digit numeric value from 0 – 9.  Example: f000012 | |
| fileNumber | Mandatory. Text value.  An identifier that is unique for each <fileVolume>. It is a business identifier given to each file volume. It must be unique across all <file> and <fileVolume> elements within this SIP.  Example: 1234/2016-16V1 | |
| volumeNumber | | Mandatory. Text value.  The volume number that indicates the order in which the file volumes were created. Will usually be a numeric value.  Note: Although the fileNumber attribute should contain digits that represent the volume number, this is not guaranteed for legacy files or file volumes received from agencies. Also, the standard pattern for fileNumber used by any particular agency may be different or changed over time; so having a separate attribute that unambiguously represent the volume number is required.  Example: 1 |
|  | |  |

Each <fileVolume> element will contain the following elements:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| <creationTimePeriod> | Mandatory.  This element contains two mandatory child elements whose values are dates with the format yyyy-mm-dd (example: a value representing 1st of January 2016 will be represented as 2016-01-01):   |  |  | | --- | --- | | <from> | Mandatory. Date value.  The registered date of the earliest record within any file volumes in this particular file. | | <until> | Mandatory. Date value.  The registered date of the latest record within any file volumes in this particular file. | | |
| <dateClosed> | Mandatory. Date value.  The date on which the file volume has been closed. | |
| <creator> | | Mandatory. Text value.  The name of the person who created this file volume. |
| <record> | | Mandatory.  Every file volume in the SIP must contain at least one record.  Therefore every <fileVolume> element must contain at least one <record> element.  The <record> element is described in further detail in the next sub-section. |

## Records

The <record> element represents a single record in the SIP.

Please note that a record may be comprised of multiple digital objects (i.e. compound record). Each digital object will be different computer files.

Each <record> element contains the following attributes:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Mandatory. Text value.  An identifier that is unique across all records in the SIP. This identifier should be auto-generated according to the following numbering pattern:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **r** | **n** | **n** | **n** | **n** | **n** | **n** |   **r**: is a fixed constant string representing a record.  **n**: represents a single digit numeric value from 0 – 9.  Example: r000007 |

Each <record> element contains the following child elements:

|  |  |
| --- | --- |
| <title> | Mandatory. Text value.  The full title of the record |
| <recordNumber> | Mandatory. Text value.  This is business identifier given to every record in the file.  Example: 1234/2016-16V1.1 |
| <recordType> | Mandatory. Text value.  This is the name of the type of the record.  Example: Correspondence  Example: Minutes of Meeting  Example: Invoice |
| <dateRegistered> | Mandatory. Date value.  This is the date on which the record is registered into an EDRMS system or a registry book, when it is created, or received from an external party.  This date value must be in the format yyyy-mm-dd (example: a value representing 1st of January 2016 will be represented as 2016-01-01). |
| <author> | Optional. Text value.  The name of the author of the record. |
| <creator> | Mandatory. Text value.  The name of the person who entered or registered the record into the EDRMS system or a registry book.  The author and the creator could be, but isn’t always necessarily the same person. |
| <securityLevel> | Mandatory. Text value.  A single character code representing the security level of the record.  The value for this code must be one of the following:   |  |  | | --- | --- | | U | Unclassified | | C | Confidential | | R | Restricted | | S | Secret | | T | Top Secret | |  |  | |
| <formOfAppearance> | Mandatory. Text value.  This specified whether the record is a digital or non-digital record.  The value must be one of the following:   |  |  | | --- | --- | | unspecified | Not specified | | digital | The record is digital | | non-digital | The record is non-digital | | mixed | The record may comprise of components that are a mixed of digital computer files and non-digital artifacts like paper. | |
| <digitalObjectRef> | A record may be comprised of multiple digital objects (i.e. compound record). Each digital object will be different computer files.  The digital objects will be described under the <toc> element of the SIP.    Over here under the <submission> element of the SIP, the record will reference those digital objects through the <digitalObjectRef> element. The value of the <digitalObjectRef> element will be the value of the id attribute of the referenced digital object.  Each <record> element must have at least one <digitalObjectRef> child element. |
| <additionalInfo> | Optional.  The EDRMS system from which the SIP is generated may hold other business metadata regarding the record that adds important contextual information to the record.  Such metadata is described under the <additionalInfo> element.  This element has the following child elements:   |  |  | | --- | --- | | <attribute> | Each separate piece of business metadata will be encapsulated within an <attribute> element. This element will have its own XML attribute called name, which is the name of the metadata, and the element’s value will be the corresponding value of the business metadata.  An <additionalInfo> element must have at least one <attribute> child element.  Example:  <additionalInfo>  <attribute name="sender\_organization">Ministry of Finance</attribute>  <attribute name="date\_sent">2016-01-01</attribute> </additionalInfo> | |  |  | |
| <relationships> | Optional.  The EDRMS system from which the SIP is generated may hold information regarding the relationships between records in the same SIP that adds important contextual information to the record.  Such information is described under the <relationships> element.  This element has the following child elements:   |  |  | | --- | --- | | <relationship> | Each separate relationship will be encapsulated within an <relationship> element. This element will have its own XML attribute called type, which is the type of relationship, and ref will be the value of the corresponding id of the linked record existing in the same SIP.  An <relationships> element must have at least one <relationship> child element.  Example:  <relationships>  <relationship type=”RELATES TO” ref=”r000005”/>  </relationships> | |  |  | |
| <digitalObjectReference> | A record may or may not have digital objects depending on the record’s <formOfAppearance> value.  If a record’s <formOfAppearance> is digital or mixed, then there must at least be one <digitalObjectReference> element within the <record> element if not more.  If a record’s <formOfAppearance> is non-digital, then there must not be any <digitalObjectReference> element within the <record> element.  If a record’s <formOfAppearance> is unspecified, then there may or may not be any <digitalObjectReference> elements within the <record> element.  The value of the <digitalObjectReference> element must correspond to the value of the attribute named id of one of the <digitalObject> elements under the <toc> section of the SIP.  Example: d000007 |

# Sample SIP

A sample of a correct SIP XML and XSD is included with this specification to illustrate how the entire SIP package will look like.

1. Exceptions are made for the metadata.xml and metadata.xsd file under the header folder which should not be renamed. [↑](#footnote-ref-1)
2. See the note in section 9.2 on why renaming of folders and computer files (i.e. ‘digitalObjects’) is sometimes required. [↑](#footnote-ref-2)