

File Format Identification (CPP-008)

CPP-Identifier	File Format Identification
CPP-Label	CPP-008
Author	Kris Dekeyser
Contributors	Bertrand Caron
Evaluators	Matthew Addis, Maria Benauer, Laura Molloy
Date of edition completed	29.08.2025
Change history	Comments
Version 1.0 - 29.08.2025	Milestone version

1. Description of the CPP

The TDA identifies file formats to the appropriate level of precision, based on an existing registry (IANA MIME types, PRONOM, etc.).

Inputs and outputs

Input(s)	
Data	<i>File</i>
Documentation / guidance	File format policy - Identification (identification tool(s), handling of format identification issues, container extraction)
Output(s)	
Metadata	<i>Technical metadata</i> (format identifier(s), accompanied by a registry identifier)
	<i>Provenance metadata</i> (date; outcome (i.e. success or failure); tool, version and output)

Definition and scope

According to PREMIS, a format is defined as “*a specific, pre-established structure for the organization of a digital file or bitstream*” and “*has to correspond to some formal or informal specification*”¹. In other words, a format defines how information is stored and structured within a digital *File*. The format of a *File* is essential evidence to assess its risks of obsolescence or incompatibility with future systems.

Moreover, format identification is relevant to many CPPs because several digital preservation policies and actions depend on file format information, for the following reasons:

- Format migration is driven by the source format information and requires format identification to validate the migration process outcome;
- Metadata extraction needs the format identifier to select the tools that extract the *Metadata*;
- Risk identification relies on the format information and *Technical metadata* of the *File*.
- The choice between rendering or emulation can be driven by a format identifier.

There exists a vast number of file formats and format versions and many of them are registered in *File Format Registries*. These registries collect knowledge about known formats such as name, version, reference documentation, signature, etc. The format identification process depends on one or more registries for detecting possible file formats and gathering detailed information about a format.

The maintenance of format registries is generally a community effort and external to the TDA. Some organisations have opted to maintain an in-house format registry, which requires specialised resources and a significant amount of digital preservation experience. In any

¹ <https://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf> - p. 249 “Format information”

case, recording the file format identifiers from the format registries is recommended. Among external registries, the following are particularly notable:

- IANA (Media types)²;
- The National Archives (PRONOM)³;
- Wikidata⁴;
- Library of Congress (Format Descriptions)⁵.

Format identification is typically performed by the detection of format signatures (i.e. one or more byte sequences at relative positions at the beginning or the end of a *File*). This method prevents scanning the entire *File* and ensures reasonable performance even for large *Files*.

The format identification may result in 0, 1 or multiple candidate file formats. Hence, the outcome of the format identification process may not be singular, or decisive (e.g. as in the case of generic file formats like text files, XML documents or ZIP archives). In such cases, additional tools (e.g. other format identification tools, schema validators, content scanners, etc.) may be employed to enrich the results and facilitate the selection of the most probable format. The application of identification tools can therefore be a simple procedure as well as a more complex workflow involving a cascade or parallel execution of multiple tools and a decision tree. It is up to the TDA to decide the level of complexity that is appropriate in the implementation of this step (driven by their context and requirements) in order to achieve the most accurate and detailed format identification result.

All details about the File Format Identification process as well as its output(s) must be recorded (e.g. tools and tool versions used, identification method used, results, etc.). This audit trail supplements the file format information, which captures information about the format of a *File* (e.g. format identifier, format name, preferred file extensions, etc.) and will be stored in the *Provenance metadata*.

The TDA should have a procedure in place to deal with inconclusive format identification results (e.g. multiple possible format identifications possible; no format could be identified⁶; format identified, but file extension does not match the format, etc.). Measures for these 'format identification exceptions' could include instructions, such as:

- Keep the results as-is, but add annotations on the issue in the *Technical* and *Provenance metadata* so that they can be revised at a later stage.
- Queue problematic *File(s)* for review. The review process can be a manual inspection and/or running a set of format validators. If the review reveals a gap in the format registry, the TDA can also opt to contribute a new format definition or an update to it.
- Reject the *File(s)* or entire *Information package(s)*.

It is important to note that certain formats can serve as a container for *Files*. Examples of such container formats are archives (zip, tar, rar, etc.) or emails with attachments. A policy should define the conditions and requirements for the extraction of such *Files* and whether they should be exported as separate *Files*. If extracted, these *Files* should be considered as new *Files*, which means that they need to undergo the entire ingest stack of processes (e.g. virus scanning, checksum creation, format identification, format validation, etc.). The process of File Format Identification can therefore become a recursive action.

This CPP does not include identification of the embedded *Bitstream(s)* of *Files*. These *Bitstreams* cannot be extracted as standalone *Files* (e.g. multimedia *Files* with embedded audio, video and subtitle streams; multiple images in a TIFF *File*). However, their characterisation is important and typically performed by CPP-009 **Metadata extraction**.

² <https://www.iana.org/assignments/media-types/media-types.xhtml>

³ <https://www.nationalarchives.gov.uk/PRONOM/Default.aspx>

⁴ https://www.wikidata.org/wiki/Wikidata:WikiProject_Informatics/Structures/File_formats

⁵ <https://www.loc.gov/preservation/digital/formats/fdd/descriptions.shtml>

⁶ Results can also be unsatisfying (e.g. unrecognised *Files* being categorised as "application/octet-stream") or not precise enough (e.g. EPUB *Files* being categorised as ZIPs).

Process description

Trigger event(s)

Trigger event	CPP-identifier
A new <i>SIP</i> is submitted and processed	CPP-029 (Ingest)
<i>File</i> update or replacement due to a <i>Preservation Action</i> (e.g. migration)	CPP-014 (File Migration)
Any other action that results in a new or updated <i>File</i> to be added to the system	

Step-by-step description

No	Supplier	Input	Steps	Output	Customer
1		<i>File</i>	Apply the format identification tool(s) on the <i>File</i>	File format identification information	
	CPP-018 (Community Watch)	File format policy - Identification			
		Tool: format identification tool			
2		File format identification information	Parse and evaluate its output	File format identification successful (step 3)	
				File format could not be identified (step 2a)	

2a		File format identification information	Analyse the format identification information and decide on next steps	A gap in the format registry is identified (step 2a- optional)	
				Need for configuration update identified (step 2a- optional)	
				Manual file format identification possible (step 3)	
		File format policy - Identification		Mark the <i>File</i> for later review and add the reason for review to the event information (step 5)	
				Reject the <i>File</i> (step 5)	
2a-optional		<i>File</i>	Contribute to format registries by providing sample <i>File</i> to the registry administrators	<i>File</i> samples	
				Registry update request and/or signature submission	
		Format Identification Tool	Update system configuration	New configuration settings	
3		File format information	Check if the file format is a container format and the policy requires its content to be extracted	Extraction required (e.g. as part of the ingest processing): extract <i>File</i> , ingest it and start identification process	

				anew (step 1). This loop can be repeated as many times as needed.	
		File format policy - Identification		No extraction required (step 4)	
4		File format information	Store the file format information in the <i>File's Technical metadata</i>	Format information in <i>Technical metadata</i>	CPP-009 (Metadata Extraction)
					CPP-010 (Format Validation)
					CPP-014 (File Migration)
5			Register a format identification event containing the tool(s) name and version and the(ir) outcome(s)	<i>Provenance metadata</i>	

Rationale(s)⁷ and worst case(s)

Rationale	Impact of inaction or failure of the process
Identify the <i>File</i> 's format and store it with the <i>File</i> 's <i>Technical metadata</i> during the <i>Object</i> 's life cycle in the TDA	<i>Files</i> becoming obsolete; impossibility to extract <i>Technical metadata</i> ; <i>File</i> being overlooked when identifying <i>File(s)</i> at risk
Find <i>Files</i> where the file format cannot be determined or where the identified file format is too generic to be of practical use (e.g. plain text or a binary octet stream).	The specific file format and contents of the <i>File</i> are unknown which prevents further preservation actions being applied (e.g. file format validation or migration) or the <i>File</i> being usable for the designated community (suitable tools for opening, rendering or using the <i>File</i> cannot be identified).

2. Dependencies and relationships with other CPPs

Dependencies

CPP-ID	CPP-Title	Relationship description
/	/	/

Other relations

Relation	CPP-ID	CPP-Title	Relationship description
Required by	CPP-009	Metadata Extraction	Metadata Extraction (CPP-009) depends on the file format to aid selection of the appropriate extraction tool.
Required by	CPP-010	File Format Validation	Format Validation needs to know the format to aid selection of the appropriate validation tool.
Required by	CPP-013	Object Management Reporting	File format identification reports are required for a TDA to enable it to manage its content.

⁷ Term derived from PREMIS.

Required by	CPP-014	File Migration	Format Migration needs to know the source format to aid application of the appropriate migration plan.
Required by	CPP-016	Metadata Ingest and Management	Format information needs to be stored with the <i>File's Technical metadata</i> .
Required by	CPP-023	Risk Definition and Extraction	Risks can be related to a file format generic properties and concern all instances of it.
Required by	CPP-026	File Normalisation	The format information of the <i>File</i> in question is one of the deciding input parameters when considering File Normalisation.
Required by	CPP-029	Ingest	File formation identification is one of the core processes that must be performed during ingest.
Not to be confused with	CPP-010	File Format Validation	CPP-008 is only about identifying the format while CPP-010 describes full scanning of the <i>File</i> to ensure it complies with the format standard.

3. Links to frameworks

Certification

Certification framework	Term used in framework to refer to the CPP	Section
CTS Link	/	CoreTrustSeal doesn't explicitly focus on format identification as a separate requirement; it is an integral part of a TDA's data management practices. Identification is implied by references to format migration and is explicitly mentioned under R0(5) Levels of Curation.
Nestor Seal Link	/	Nestor Seal does not explicitly mention Format Identification but this process is in scope, and sets the ground for C30 <i>Technical metadata</i> .
ISO 16363 Link	"identify the file type"	4.2.5.1 "The repository shall have tools or methods to identify the file type of all submitted Data Objects"

Other frameworks and reference documents

Reference document	Term used in framework to refer to the process	Section
OAIS Link	/	/
PREMIS Link	"Identification of the format of a file or bitstream"	PREMIS provides a framework for recording and managing the format information in its <i>format</i> semantic unit, p. 65.

4. Reference implementations

Example use case(s)

PDF flavour identification

Institutional background	
Institution	TIB – Leibniz Information Centre for Science and Technology and University Library, Germany
Hyperlink	/
Description	
Trigger event	Ingest of PDF <i>Files</i> from academic publishers.
Problem statement	The format identification tool, DROID, returns multiple format identifiers for PDF <i>Files</i> , as PDF <i>Files</i> can declare conformance to both PDF/X and PDF/A versions.
Proposed solution	The organisation chooses one among the two results (in this case, PDF/A) returned by DROID and records the relevant PUID along with the format registry used (PRONOM).

Publicly available documentation

Institution	Organisation type	Language	Hyperlink
TIB – Leibniz Information Centre for Science and Technology and University Library, Germany	National library	English	https://wiki.tib.eu/confluence/spaces/lza/pages/93608618/Ingest
	Non-commercial digital preservation service		
	Research infrastructure		
	Research performing organisation		
CSC – IT Center for Science Ltd., Finland	Non-commercial digital preservation service	English	https://urn.fi/urn:nbn:fi-fe2020100578094 (section 2.4.4.1)
Archivematica	Digital preservation system	English	https://www.archivematica.org/en/docs/archivematica-1.17/user-manual/preservation/preservation-planning/#identification