

Identifier Management (CPP-005)

| | |
|----------------------------------|--------------------------------|
| CPP-Identifier | CPP-005 |
| CPP-Label | Identifier Management |
| Author | Mikko Laukkanen, Juha Lehtonen |
| Contributors | Bertrand Caron, Johan Kylander |
| Evaluators | Felix Burger, Maria Benauer |
| Date of edition completed | 29.08.2025 |
| Change history | Comments |
| Version 1.0 - 29.08.2025 | Milestone version |

1. Description of the CPP

Identifiers are assigned to *Objects*, *Information packages* and/or *Metadata*, and managed along to their life cycle.

Inputs and outputs

| Input(s) | |
|--------------------------|---|
| Data | <i>Information package</i> |
| | <i>Object</i> |
| Metadata | <i>Technical metadata</i> |
| | <i>Descriptive metadata</i> |
| Documentation / guidance | Identifier creation and management policy |
| Output(s) | |
| Metadata | Identifier-enriched <i>Information package</i> , <i>Object(s)</i> or <i>Metadata</i> |
| | <i>Provenance metadata</i> |

Definition and scope

Identifier management is the process of creating and updating identifiers and assigning them to *Objects*, *Metadata* or *Information packages*. Identifiers are essential components of digital preservation systems, serving as stable, long-term references to *Digital Objects* that can remain valid even when the *Objects* themselves are moved, renamed, or migrated to new systems.

Identifiers must be managed throughout the entire life cycle, taking into account any changes to their associated *Objects*, *Metadata* or *Information packages*. It is important to consider that not all types of identifiers are globally unique, some are unique only within their own identifier system.

A Persistent Identifier (PID) system can be used to generate unequivocal¹ identifiers to ensure that *Objects* can be precisely identified worldwide. PIDs are machine-readable strings of characters that conform to a defined scheme. Through providing and updating the reference link in the *Metadata*, these identifiers prevent the fundamental problem of "link rot" and ensure reliable access to preserved *Digital Objects* over time. However, this requires continuous maintenance of the identifiers to keep the *Metadata* up-to-date. Depending on the use case, it may be useful to assign multiple identifiers from different systems to an entity. To be able to provide user-facing PIDs, a TDA must manage local identifiers which provide the minimal baseline for providing persistent access and control to the data.

¹ This term is preferred over the "unique" adjective applied traditionally to identifiers. Indeed, it suggests that an identifier must reference one and only one thing, while "unique" might suggest that the thing must be referenced by only one identifier.

Common examples of PIDs are Digital Object Identifier (DOI), Uniform Resource Name (URN), handles and Archival Resource Key (ARK). One advantage of using PIDs is that their *Metadata* can be used to not only provide information about the *Object* itself, but also about its status, access conditions, and storage location. Even *Objects* which are not publicly accessible or have been disposed, can be identified and described by a PID. PIDs can also be moved from one organisation's administration to another.

All types of identifiers can be assigned to multiple levels of entities, creating a hierarchical identification structure that reflects the complex nature of digital collections and their preservation requirements. Identifiers are usually assigned on the level of a) *Objects*, b) collections and aggregations, and c) *Information Packages* (*AIPs* and *DIPs*). In addition, identifiers can be assigned to *Metadata* entities, collections of other related entities, and even institutions or persons.

Identifiers and their *Metadata* should be updated according to the entity's life cycle. In particular, when an entity may be deleted, merged, split or become partially unavailable, its identifier should be preserved. Moreover, its *Provenance metadata* should be updated in order to provide proper detail of information to the end users about its initial entity as well as the relationships to potential new entities that were created from the initial one. When using PIDs, some changes (e.g. the creation of identical parallel copies of the data that create new internal identifiers for each copy) can be documented in the PID version *Metadata* without creating a new PID.

Identifiers in a TDA are created at specific strategic points throughout the preservation life cycle, with timing and methods varying based on institutional policies and system architectures. Identifiers are typically assigned during *Ingest* as part of the packaging process, ensuring that every preserved *Object* has a persistent reference from the moment it enters the system. However, some institutions create identifiers earlier in the workflow (e.g. during acquisition planning or transfer preparation). This is especially useful when using PIDs, since it allows for early referencing and tracking of *Objects* before they undergo preservation processing. Identifiers can also be created after the initial preservation processing is complete, particularly when the final preserved format and structure have been determined. Identifiers can be also assigned to services or *Objects* which are not stored in the TDA but only generated on the fly based on user requests.

Identifiers may reveal the hierarchical relationships in the identifier string (e.g. by using qualifiers²) or might hide them by creating a whole new string for components³. This CPP does not choose between these approaches. Similarly, it does not make any assumptions on the organisation in charge of managing identifiers and whether identifiers are managed by the TDA directly. Since PID management is relatively resource-intensive and can also be performed outside the scope of digital long-term preservation, no assumptions are made here about the structure or organisation of this work area; instead, reference is made only to the entity "the PID management service".

² For example, the identifier <id:c8b> will be assigned to a Representation, and <id:c8b/001> to its first component or file.

³ In the previous example, the identifier <id:t5g> could then be assigned to the first component or file.

Process description

Trigger event(s)

| Trigger event | CPP-identifier |
|---|---|
| Pre-ingest transfer preparation | CPP-029 (Ingest) |
| Ingestion workflow | CPP-029 (Ingest) |
| Creation of new <i>Files</i> or <i>Representations</i> | CPP-028 (Creation of derivatives) |
| Replacement of corrupted <i>Files</i> | CPP-004 (Data Corruption Management) |
| Data export | CPP-006 (AIP Batch Export) |
| Data replication | CPP-011 (Replication) |
| Data migration | CPP-014 (File Migration) |
| Data normalisation | CPP-026 (File Normalisation) |
| Metadata ingest and creation | CPP-016 (Metadata Ingest and Management) |
| Occasional | |
| Data version update | CPP-021 (AIP Versioning) - if the new version gets a new PID. |
| Broken <i>File</i> needs a new identifier | CPP-027 (File Repair) |
| <i>Information package</i> , <i>File</i> or <i>Metadata</i> is removed from the <i>TDA</i> holdings | CPP-017 (Disposal) |

Step-by-step description

| No | Supplier | Input | Steps | Output | Customer |
|----|----------|--|--|-----------------|--|
| 1a | | A producer or a TDA has a need to reserve an identifier, (e.g. a PID, prior to the entity being added) | Reservation of identifier prior to new entity assignment (step 2) | | |
| 1b | | <i>Object</i> | New entity added or a need to assign an identifier to an existing entity (step 2) | | |
| | | <i>Information package</i> | | | |
| | | <i>Metadata</i> | | | |
| 1c | | <i>Object</i> | Entity with an identifier has changed (step 4) | | |
| | | <i>Information package</i> | | | |
| | | <i>Metadata</i> | | | |
| 1d | | <i>Object</i> | Entity is disposed (step 7) | | |
| | | <i>Information package</i> | | | |
| | | <i>Metadata</i> | | | |
| 2 | | Identifier creation and management policy | Create a new identifier according to the TDAs policy for identifier management | Identifier | |
| 3 | | (new) Identifier | Assign the new identifier to the entity and add it as a part of the entity's <i>Metadata</i> | <i>Metadata</i> | Many CPPs, e.g. CPP-004 (Data Corruption Management) |
| | | <i>Object</i> | | | |
| | | <i>Information package</i> | | | |

| | | | | | |
|---|--|---|---|---|--|
| | | <i>Metadata</i> | | | CPP-011 (Replication) CPP-014 (File Migration) CPP-016 (Metadata Ingest and Management) CPP-021 (AIP Versioning) CPP-025 (Enabling Access) CPP-027 (File Repair) CPP-028 (Creation of Derivatives) CPP-029 (Ingest) |
| 4 | | Identifier creation and management policy | If the changed entity has a PID assigned to it: evaluate if a new PID is required | Need for new PID identified (go back to step 2) | |
| | | | | No need for new PID identified (step 5) | |
| 5 | | | Update or add identifier relationships (e.g. hierarchical relations, sequential relations etc.) for the assigned entity | <i>Metadata</i> | |
| 6 | | | Update <i>Provenance metadata</i> for the entity so that identifiers have a history | <i>Provenance metadata</i> | |

| | | | | | |
|---|-----------------------|-----------------|---|--|--|
| 7 | CPP-017 (Disposal) | Disposed entity | If the entity is disposed: retain minimum metadata and the identifier for the disposed entity | | |
|---|-----------------------|-----------------|---|--|--|

Rationale(s)⁴ and worst case(s)

| Rationale | Impact of inaction or failure of the process |
|---|--|
| The rationale for implementing PIDs in TDAs stems from fundamental challenges in maintaining long-term access to digital objects and the core mission of preservation itself. | Link rot as well as Problems and challenges in <ul style="list-style-type: none"> • Internal data management problems • System migrations • Format migrations • Activity tracking • Interoperability |

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-016 | Metadata Ingest and Management | While ingesting into a TDA, the <i>Metadata</i> should be assigned an identifier. Also, the management functions of the <i>Metadata</i> may require replacing and/or updating identifiers. |
| Required by | CPP-017 | Disposal | When the life cycle of the <i>Digital Object</i> or <i>File</i> ends, the identifier should be updated to “retired” status. |
| Required by | CPP-021 | AIP Versioning | When an <i>AIP</i> gets a new version, the new <i>AIP</i> version must also be assigned a new identifier. |
| Required by | CPP-024 | Enabling Discovery | Enabling Discovery should make use of identifiers. |

⁴ Term derived from PREMIS.

| | | | |
|--------------------|---------|-----------------------------|--|
| Required by | CPP-025 | Enabling Access | Accessing <i>Digital Object</i> , <i>File(s)</i> or <i>Metadata</i> should be based on identifiers as parameters. |
| Required by | CPP-029 | Ingest | The ingestion workflow is responsible for assigning identifiers to various entities in TDA, such as <i>Files</i> and <i>Metadata</i> . |
| May be required by | CPP-004 | Data Corruption Management | If a <i>File</i> is corrupted, it may need to be repaired or replaced. During this process, a new identifier may be created. |
| May be required by | CPP-011 | Replication | When a <i>Digital Object</i> or <i>File</i> is replicated, the replicant may be assigned a new identifier. |
| May be required by | CPP-013 | Object Management Reporting | The management and reporting should require that the data is identified with identifiers. |
| May be required by | CPP-014 | File migration | During format migration, the migrated <i>File</i> format may be assigned a new identifier. |
| May be required by | CPP-019 | Data Quality Assessment | The data quality assessment may include validating the identifiers and their linked resources. |
| May be required by | CPP-026 | File normalisation | A normalised <i>File</i> format may be assigned with a new identifier. |
| May be required by | CPP-027 | File Repair | A repaired <i>File</i> may get a new identifier. |
| May be required by | CPP-028 | Creation of Derivatives | A derivative of a <i>File</i> may get its own identifier. |

3. Links to frameworks

Certification

| Certification framework | Term used in framework to refer to the CPP | Section |
|-------------------------------------|--|---|
| CTS Link | Persistent Identifiers | R09 Preservation Plan R12 Discovery and Identification |
| Nestor Seal Link | Persistent Identifiers | C27 Identification |

| | | |
|-----------------------------------|------------------------|-----------------------------|
| ISO 16363 Link | Persistent Identifiers | 4.2.4 4.2.5.4 4.2.6.3 |
|-----------------------------------|------------------------|-----------------------------|

Other frameworks and reference documents

| Reference Document | Term used in framework to refer to the process | Section |
|--------------------------------|--|---------------------------------------|
| OAIS Link | Persistent Identifiers | 6.2.4 |
| PREMIS Link | Persistent Identifiers | Data dictionary, 1.1 objectIdentifier |

4. Reference implementations

Example use case(s)

DOI given for a research dataset by TDA

| Institutional Background | |
|--------------------------|---|
| Institution | CSC, Finland (Digital Preservation Service for Research Data) |
| Hyperlink | https://wiki.eduuni.fi/x/9ZRYH Example of a DOI for a dataset |
| Description | |
| Trigger event | Submitting research data to the TDA |
| Problem statement | Research dataset does not have a DOI |
| Proposed solution | Before submitting a dataset to TDA (DPS in Finland), the user describes the dataset via a description tool or via a metadata API. When a dataset has been submitted, TDA automatically creates a DataCite description including a new DOI, and eventually it creates a corresponding publicly available website for the dataset <i>Metadata</i> . |

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/93608951/Metadata#Metadata-Identifyingmetadata |
| | 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 2.4.1.) |
| Archivematica | Digital preservation system | English | https://www.archivematica.org/en/docs/archivematica-1.17/user-manual/transfer/transfer/#transfer-tab-microservices |