perm.pub/dsi:0123456789abcdefghijklmnopq/1

Additional formats and editions available online.

Author date: 2024-07-15

Document Succession Identifiers

E. Castedo Ellerman (castedo@castedo.com)

Citation:
E. Castedo Ellerman (2024) "Document Succession Identifiers" perm.pub https://perm.pub/dsi: 0123456789abcdefghijklmnopq/1

Abstract

DOCUMENT TYPE: Living Technical Specification

A Document Succession Identifier (DSI) is designed for use within bibliographic references for the long-term retrieval of a document succession, a type of document that is correctable yet redistributable across multiple websites. This DSI specification works in conjunction with the Document Succession Git Layout (DSGL) specification, which is a storage format for document successions. A DSI can reference either an entire document succession or, when an edition number is included, specific document snapshots. This feature lets readers quickly access both the latest edition and earlier editions. This DSI specification does not define any specific format for document snapshots; however, an example is Baseprint document snapshots.

Background

Websites like https://perm.pub use free open-source software, such as the Python package Epijats, to process the formats of Document Succession Identifiers (DSI), Document Succession Git Layout (DSGL), and Baseprint document snapshots. For the motivation behind these technologies, refer to Why Publish Baseprint Document Successions. Tutorials and introductory materials are also available at https://try.perm.pub/.

Scope

This document is a specification of DSI for interoperability with the following free open-source software reference implementations:

- the Python package Hidos version 1.3 [1] and
- the Document Succession Highly Manual Toolkit [2].

This specification does not define potential DSI features that are not implemented in any software. The online forum at https://baseprints.singlesource.pub is available for communication about this living specification, its reference implementations, and other specifications related to Baseprint document successions.

Informal Description

Base DSI

The textual representation of a Document Succession Identifier (DSI) consists of a *base DSI*, which may be optionally followed by a slash and an edition number.

Example: Base DSI of this specification document.

> `1wFGhvmv8XZfPx0O5Hya2e9AyXo`

A base DSI is a 27-character string in base64url format (RFC 4648)[3] representing a 20-byte binary hash that identifies a document succession. This DSI specification does not define a storage format for document successions. However, the reference implementations for DSI also support the storage format defined by Document Succession Git Layout (DSGL). In DSGL, the base DSI is calculated from the initial Git [4] commit of a document succession. A Git commit corresponds to a *core Software Hash Identifier (SWHID) for revisions*.

Document Snapshots

A document succession contains document snapshots, which are static and digitally encoded. In DSGL, document snapshots are either Git blobs or Git trees, which are identifiable by Software Hash Identifiers (SWHIDs) and can be archived in the Software Heritage Archive [5].

Example: SWHID of a document snapshot from 2023 of this specification.

> `swh:1:dir:eb9dfc65c22cde7b558ca2070ed4b2950074ed2f`

Example: Permalink to the Software Heritage Archive.

> *https://archive.softwareheritage.org/swh:1:dir:eb9dfc65c22cde7b558ca2070ed4b2950074ed2f*

Snapshot Edition Numbers

Edition numbers identify document snapshots within a document succession. An edition number is composed of positive integers separated by periods (possibly just a single positive integer).

Example: DSI of the first edition.

> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1`

An edition number is *multilevel* if it is composed of more than one integer (separated by periods).

Example: DSI of the fourth subedition of the first edition.

> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.4`

Every document snapshot in a document succession has an edition number assigned to it.

Example: Edition 1.4 is assigned to a document snapshot.

> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.4`

> ↓
>
> `swh:1:dir:eb9dfc65c22cde7b558ca2070ed4b2950074ed2f`

A *snapshot edition number* is an edition number that is assigned to a document snapshot.

Coarse Edition Numbers

An edition number is *coarser* than another edition number if it drops one or more of the final integers of the other. Conversely, an edition number is *finer* than another if the latter is *coarser* than it. For example, edition number `1.3` is coarser than edition number `1.3.2`, which, in turn, is *finer* than `1.3`.

A *coarse edition number* refers to an edition number that is coarser than a snapshot edition number. A coarse edition number is not assigned to a document snapshot. Instead, it implicitly identifies a dynamic sequence of editions. This sequence consists of all the finer edition numbers assigned to document snapshots.

Example:

> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1`
>
> ↓
>
> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.1`
>
> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.2`
>
> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.3`
>
> `1wFGhvmv8XZfPx0O5Hya2e9AyXo/1.4`

Formal Definitions

The following grammar is expressed in ISO/IEC 14977 Extended Backus—Naur Form (EBNF) further extended to allow an ellipsis (…) to denote a range of ASCII characters.

Textual Representation of a DSI

```
dsi = [ prefix ], base_dsi, [ "/" , [ edition_number ] ] ;
base_dsi = 26 * b64u_digit, b64u_digit27 ;
edition_number = pos_int, 3 * [ ".", pos_int ] ;
pos_int = pos_dec_digit, 3 * [ dec_digit ] ;
pos_dec_digit = "1"…"9" ;
dec_digit = "0" | pos_dec_digit ;
b64u_digit = "A"…"Z" | "a"…"z" | dec_digit | "-" | "_" ;
b64u_digit27 = "A" | "E" | "I" | "M" | "Q" | "U" | "Y" | "c" |
               "g" | "k" | "o" | "s" | "w" | "0" | "4" | "8" ;
```

The optional `prefix` is not defined in this specification but is described in the Discussion section.

Criteria for a Document Succession

The base DSI is a base64url representation of a 20-byte hash that identifies a data structure. However, this DSI specification does not define the format of the data structure. The Document Succession Git Layout (DSGL) specification, on the other hand, does define it. Different data structure formats are compatible if they record document successions that satisfy the following criteria.

Criterion: The abstract data model recorded by a document succession is a mapping from edition numbers to document snapshots. A *snapshot edition number* refers to an edition number thus mapped.

Criterion: The document snapshots contained in a document succession are static and digitally encoded.

Criterion: Edition numbers are non-empty tuples of positive integers.

Criterion: Coarse edition numbers are not assigned to document snapshots. A *coarse edition number* is an edition number that is coarser than a snapshot edition number. An edition number is coarser than another edition number if it is an initial sub-tuple. In other words, it drops integers from the end of the tuple.

Empty Edition Number

In some formal contexts, such as in software, it may be convenient to define an *empty edition number*, which corresponds to an empty tuple containing no integers. Unless otherwise noted, the unqualified term *edition number* means a non-empty edition number.

The empty edition number is not mapped to a document snapshot in document successions.

Unlisted Edition Numbers

An edition number with any integer component equal to zero is an **unlisted** edition number. Some storage formats, such as Document Succession Git Layout (DSGL), support **unlisted** edition numbers. Renderings of a document succession may safely omit and ignore unlisted editions. Some implementations may provide a mechanism to access unlisted editions, keeping in mind that the default presentation of a document succession does not include unlisted editions.

Discussion

Public Archives

Due to the content policy of the Software Heritage Archive, once a document succession is publicly archived with a snapshot edition number, reassigning an edition number is particularly difficult and unlikely to be achieved. Updating a DSGL document succession in the Software Heritage Archive by adding a new edition number is effectively the only practical way to update a document succession.

Optional DSI Prefix

Users may choose to use a DSI with or without a prefix, depending on the application context. An intuitive choice for a prefix is `dsi:`, which mirrors the acronym "DSI". For added convenience, some websites provide a URL that serves as a DSI prefix.

As of 2023, the Hidos tool supports DSIs both with and without the `dsi:` prefix in its `find` sub-command. For example:

```
$ hidos find dsi:1wFGhvmv8XZfPx0O5Hya2e9AyXo
gh-703611066 https://github.com/digital-successions/
1wFGhvmv8XZfPx0O5Hya2e9AyXo.git
```

As of 2023, the website `perm.pub` supports a URL-based prefix `https://perm.pub/`, as demonstrated in the following example:

```
$ firefox https://perm.pub/1wFGhvmv8XZfPx0O5Hya2e9AyXo
```

Future Extensions

Future implementations may use different hashing mechanisms for the base DSI, as long as the risk of identifier collision remains acceptably low.

To accommodate future enhancements, there are three methods to extend the textual representation of a DSI:

- Use a character that is neither a slash (/) nor one of the 64 base64url characters.
- Vary the number of characters from 27.
- Make the 27th character one of the 48 base64url characters that never appear as the last character in a base64url encoding of 40 bytes (that is, any base64url character other than `A`, `E`, `I`, `M`, `Q`, `U`, `Y`, `c`, `g`, `k`, `o`, `s`, `w`, `0`, `4`, or `8`).

Use of Base64url Over Hexadecimal

The textual identifier uses “base64url” (Base 64 with a URL and filename safe alphabet) as specified in RFC 4648 [3]. Both base64url and hexadecimal have their advantages and disadvantages.

The main downside to base64url is its susceptibility to copy errors when the copying process relies on human sight. Certain fonts make a poor distinction between some characters. For example, some popular sans serif fonts make no visual distinction between capital ‘I’ and lowercase ‘l’.

However, creators of a new document succession are not obligated to use a specific DSI. If a DSI is deemed unsuitable, generating a new one is straightforward.

The main advantage of base64url is its brevity, requiring only 27 characters compared to 40 in hexadecimal. Since DSIs are used in contexts similar to DOIs, a 27-character identifier is more likely to be acceptable, as it is comparable to the length of a long DOI. Additionally, a shorter ID is more suitable for display on mobile devices, reducing the likelihood of truncation, horizontal scrolling, or the need for very small fonts.

The choice of base64url is partly made on the belief that the following technology trends mitigate the copy-by-human-sight issue:

1. Use of hyperlinks, copy-and-paste, and QR codes.
2. Tools that generate websites and PDFs with customizable fonts.
3. Human-to-computer interfaces incorporating features like autocomplete and typo correction to mitigate input errors.

Acknowledgments

Thank you to Valentin Lorentz for raising questions about design choices and pointing out an important shortcoming in how GPG digital signatures were used in the initial implementation of the Hidos library (version 0.3) [6].

This document has been copyedited with AI using https://copyaid.it.

Further Reading

- This specification is heavily influenced by the concept of *intrinsic identifier* and related concepts discussed in [5] [7].

- For a discussion on various concepts and proposed terminology regarding persistent identifiers, see [8]. According to this proposed terminology, a DSI is a persistent identifier (PID) that is "frozen" and "waxing" with "intraversioned" and "extraversioned" PIDs depending on the edition number.

Changes

From Edition 2.2 to 2.3

- Restrict integers of an edition number to have no more than 4 digits.
- Restrict edition numbers to have no more than 4 integers (separated by periods).
- Change unqualified edition number to not include integers of zero.
- Add section about unlisted edition numbers.

From Edition 2.1 to 2.2

- Moved Git storage details into the new DSGL specification.
- Expanded material into formal definition and informal description sections.

From Edition 1 to 2

- The term "digital succession" has been updated to "document succession."

From Edition 1.2

- References to SSH signing keys have replaced mentions of GPG/PGP signing keys.

References

1. Hidos 1.3. 2024. Available: https://archive.softwareheritage.org/swh:1:rev:0c997b13a255be2a83f150371c9364a1217fa91a;origin=https://gitlab.com/perm.pub/hidos
2. Document succession highly manual toolkit manual. 2024. Available: https://archive.softwareheritage.org/swh:1:rev:f6da04dce5f53d88c4c324c1d2546110a8d42d8a;origin=https://gitlab.com/perm.pub/dshmtm
3. Josefsson S. The Base16, Base32, and Base64 data encodings. RFC Editor; Internet Requests for Comments; RFC Editor; 2006 Oct. doi:10.17487/RFC4648
4. Git — Wikipedia, the free encyclopedia. 2023. Available: https://en.wikipedia.org/w/index.php?title=Git&oldid=1177307938
5. Cosmo RD, Gruenpeter M, Zacchiroli S. Referencing Source Code Artifacts: A Separate Concern in Software Citation. Computing in Science & Engineering. 2020;22: 33–43. doi:10.1109/MCSE.2019.2963148
6. Hidos 0.3. 2022. Available: https://archive.softwareheritage.org/swh:1:rev:b963e5d2366724df6e8c34d864a7984ce4a2e1be;origin=https://gitlab.com/perm.pub/hidos
7. Di Cosmo R, Gruenpeter M, Zacchiroli S. Identifiers for Digital Objects: the Case of Software Source Code Preservation. iPRES 2018 - 15th International Conference on Digital Preservation. Boston, United States; 2018. pp. 1–9. Available: https://hal.archives-ouvertes.fr/hal-01865790
8. Kunze J, Calvert S, DeBarry JD, Hanlon M, Janée G, Sweat S. Persistence Statements: Describing Digital Stickiness. Data Science Journal. 2017;16: 39–. doi:10.5334/dsj-2017-039