3D Tiles Package Specification

Table of Contents

Abstract

This document describes the specification for the 3D Tiles Package format. *It defines a standard way to package a [3D Tiles] tileset and associated resources in a single file.*

References

- [SQLite], SQLite, 2004 https://sqlite.org/
- [SQLite Database File Format], SQLite, 2004 https://sqlite.org/fileformat2.html
- [3D Tiles], Cesium GS Inc. https://github.com/CesiumGS/3d-tiles
- [JSON Format], IETF RFC8259: The JavaScript Object Notation (JSON) Data Interchange Format, 2017, https://datatracker.ietf.org/doc/html/rfc8259
- [RFC 3986] Berners-Lee, T., Fielding, R., and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, https://www.rfc-editor.org/info/rfc3986
- [RFC 3987] Duerst, M. and M. Suignard, Internationalized Resource Identifiers (IRIs), RFC 3987, DOI 10.17487/RFC3987, January 2005, https://www.rfc-editor.org/info/rfc3987

1. 3D Tiles Package Format Specification

A 3D Tiles Package is distributed as a single file in the [SQLite Database File Format]. The [SQLite] library provides a means for creating, reading and modifying these files, independent of the

computing platform or programming language.

1.1. Database File Format

A 3D Tiles Package SHALL be a file in the [SQLite Database File Format]. The file extension of such a file SHALL be `.3dtiles`.

The `user_version` of the SQLite file, as defined in Section 1.3.14 of [SQLite Database File Format], SHALL contain a version number of the Package file. This version number is encoded in a 4-byte big-endian integer as `<major>*10000 + <minor>*100 + <patch>`. For example, a `user_version` of `10000` indicates a 3D Tiles Package version `1.0.0`. A `user_version` of `123456` indicates a 3D Tiles Package version `12.34.56`.

1.2. Database File Content - Version 1.0.0

1.2.1. Database Schema

A 3D Tiles Package with version 1.0.0 SHALL contain a single table called `media`. This table consists of two columns:

- `key` with the type `TEXT`
- `content` with the type `BLOB`

1.2.2. Key Column

The `key` column contains the path of a certain resource in the package. This path may be a `path-noscheme` as defined by [RFC 3986], Section 4.2 or or a `ipath-noscheme` as defined by [RFC 3987], Section 2.2. The keys SHALL be unique after normalization, as described in [RFC 3986], Section 5.3.2.3.

Reserved characters in URI and IRI must always be percent-encoded, according to [RFC 3986] and [RFC 3987]. This does not necessarily apply to characters that are neither reserved nor unreserved characters. Specifically, the space character `" "` may or may not be percent-encoded as `%20` inside a URI or IRI. Unreserved characters may or may not be percent-encoded.

NOTE

Creators of 3D Tiles Packages should be aware of path components that may limit the portability of the packages. This mainly refers to the case that such a package is extracted into a file system, and the keys are used to define the directory structure in this file system. For example: It is technically possible and legal to define a URI or IRI path with a backslash character `"\"` when it is percent-encoded as `%5C`. But when this key is supposed to represent a file in a file system, this may cause undefined behavior due to the special meaning that this character has for certain operating systems. Refer to [RFC 3987], Section 7.3. for further examples of cases where special care has to be taken during the interpretation of URI- or IRI paths.

1.2.3. Content Column

The `content` column contains the resource data as a BLOB. The type of the content is application-dependent, and SHALL be determined by clients. Clients SHALL NOT rely on information from the `key` column for detecting the content type. This means that clients SHALL NOT treat parts of the `key` as a file extension and use this for determining the data type.

The data in the `content` BLOB may be compressed. Clients SHALL examine the BLOB, and detect whether any compression has been applied from the first bytes of the BLOB.

NOTE	For example, when the first bytes of a `content` BLOB are $1f_{16}$ and $8b_{16}$, then this indicates that the content data is compressed with GZIP. Clients may then uncompress this data, or use the compressed form of the data directly, depending on the application context.

1.2.4. Database Content

The `media` table SHALL contain at least one row, where the `key` is `tileset.json`, and the `content` is a BLOB that contains the binary data of a file in [JSON Format] that contains a 3D Tiles tileset, as defined in the section "Tileset JSON" of the 3D Tiles specification.

1.2.5. Resolving References

The structure of the elements in a 3D Tiles Package closely resembles the structure of a 3D Tiles tileset in a file system: The `key` column of the table describes the path to a certain resource. When such a resource refers to other resources, then these references SHALL be given as *relative* paths. These paths are resolved against the path that results from resolving the path of the containing resource against the base path of the package. Relative paths SHALL NOT refer to resources that are *outside* of the package itself.

NOTE

Example

An example of the contents of a 3D Tiles Package. The main `tileset.json` file refers to a tile content with the *relative* path `tiles/tile0.gltf`. The tile content refers to an image with the *relative* path `../images/image0.png`. This path is resolved against the path of the tile content, resulting in the path `images/image0.png`.

Path (`key`)	Description	Comments
`tileset.json`	The main tileset file	Contains a tile with `content.uri = "tiles/tile0.gltf`
`tiles/tile0.gltf`	The content of one tile of the tileset	Contains an image with `image.uri = "../images/image0.png"`
`images/image0.png`	An image that is used as a texture	

If the tile content refered to an image with a path like `../../external.png`, then this package would be considered to be invalid, because the path can not be resolved within the package.

A 3D Tiles Package SHALL NOT refer to external elements with absolute URIs or IRIs.

Appendix A: Revision history

Date	Release	Editor	Primary clauses modified	Description
2022-??-??	1.0.0	Sean Lilley	all	This is the first normative version of this document.