

OGC GeoPackage Related Tables Extension Interoperability Experiment

Table of Contents

1. Summary	4
1.1. Requirements & Research Motivation	4
1.2. Prior Work	4
1.3. Summary of Experiments and Results	5
1.4. Future Recommendations	5
1.5. Document contributor contact points	5
1.6. Foreword	5
2. References	7
3. Terms and Definitions	8
3.1. Abbreviated terms	9
4. Overview	10
5. Design	11
5.1. Overview	11
5.2. Requirements Classes	12
5.2.1. Media	13
5.2.2. Simple Attributes	13
5.2.3. Features	14
5.3. Usage scenario	14
6. Technology Integration Experiments	19
6.1. Test Samples and Descriptions	19
6.2. ETS Results	20
6.3. TIE Results for RTE-Aware Software	23
6.4. TIE Results for non-RTE-Aware Software	25
7. Discussion of Design Decisions and Ramifications	27
7.1. Use Cases	27
7.2. Semantics	27
7.2.1. Relation Name	27
7.2.2. Mapping Table Name Use with Schema Extension	28
7.2.3. Metadata	28
7.3. Cardinality	28
7.4. External References	29
7.5. Handling of Unregistered MIME Types	29
7.6. Implications of gpkg_extensions Requirements for Non-Conforming Clients	29
7.7. User-Defined Mapping Table UUIDs	30
7.8. Preventing Data Duplication	30
Appendix A: Revision History	31

Publication Date: 2018-08-22

Approval Date: 2018-07-10

Posted Date: 2018-05-31

Reference number of this document: OGC 17-093r1

Reference URL for this document: <http://www.opengis.net/doc/PER/gpkg-rte-ie-er>

Category: Public Engineering Report

Editor: Jeff Yutzler, Ashley Antonides

Title: OGC GeoPackage Related Tables Extension Interoperability Experiment

OGC Engineering Report

COPYRIGHT

Copyright © 2018 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

This OGC Engineering Report describes the results of the OGC GeoPackage (GPKG) [Related Tables Extension Interoperability Experiment \(GPKG-RTE IE\)](#) [<http://www.opengeospatial.org/projects/initiatives/gpkg-rteie>]. This IE tested a proposed extension to the OGC GeoPackage Encoding Standard (12-128r14). The GPKG-RTE defines the rules and requirements for associating tables with existing feature or attribute tables in a GeoPackage data store. As part of this IE, the participants performed Technology Integration Experiments (TIEs) where they produced GeoPackages that used this extension, loaded them into GPKG-compliant software systems, and observed the results. As a result of this work, the IE participants agree that the extension is fit for use and consideration as a standard by OGC.

1.1. Requirements & Research Motivation

The purpose of the GPKG-RTE is to define relationships between feature tables and tables that hold related content, including multimedia, simple attributes, and other features. One use case for this extension is to associate features with related multimedia content such as:

- photographs;
- audio or video files; or
- PDFs.

For example, implementing this extension would provide the ability to associate pictures of a house with a specific parcel (land lot).

It is also possible to use the GPKG-RTE to associate features with simple attributes or features with other features. The GPKG-RTE supports many-to-many relationships, which allows a natural mapping from complex data models.

This extension, like all GeoPackage extensions, is intended to be transparent and to not interfere with GPKG-compliant, but non-supporting, software packages.

The goal of the IE was to verify that the extension was correctly designed to meet the design goals and to be transparent in this manner. This goal was achieved by building GeoPackages containing embedded multimedia content and sharing those GeoPackages with other software products, some compliant with the extension and others unaware of it.

1.2. Prior Work

Before this interoperability experiment, Compusult had produced a Related Tables Extension that it used internally. While Compusult was satisfied with the extension, they recognized that it would only achieve its full potential if it were standardized. Without standardization, there was the risk that other organizations would implement competing extensions and that there would not be interoperability between them.

1.3. Summary of Experiments and Results

Five RTE-aware and two non-RTE-aware software packages were tested using 13 samples from seven different providers. These integration experiments demonstrated that the Related Tables Extension works and is backward compatible (i.e., does not fail when loaded by non-RTE-aware software).

The experiments also highlighted some considerations for developers to be aware of and areas for future work. Producers of GeoPackages with the RTE need to register the extension (otherwise it might not be detected) and should test the files using the Executable Test Suite (ETS) to ensure conformance. The most common interoperability case for the RTE involved feature base tables and attributes as the related table; however, client software should be aware of other possible combinations (e.g., an attributes table as the base table or a features table as the related table). Finally, both producers and consumers should be aware of file sizes and complexity introduced by different media types, especially for mobile applications.

Now that this IE is complete, the participants are confident that the extension is ready to be standardized and adopted by OGC.

1.4. Future Recommendations

The GeoPackage SWG should finalize the [GeoPackage Related Tables Extension](http://www.geopackage.org/18-000.html) [http://www.geopackage.org/18-000.html] and submit it to OGC for consideration as an adopted GeoPackage Extension.

1.5. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors.

Table 1. Contacts

Name	Organization
Jeff Yutzler	Image Matters
Tracey Birch	SOFWERX
Jason MacDonald	Compusult
Ashley Antonides	Radiant Solutions
Brad Hards	—

1.6. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might

be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following normative documents are referenced in this document.

- OGC: [OGC 12-128r14 OGC® GeoPackage Encoding Standard v1.2.0, 2017 \(On-line\)](http://www.geopackage.org/spec120/index.html)
[<http://www.geopackage.org/spec120/index.html>]

Chapter 3. Terms and Definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- attributes data

Non-spatial tabular data that is designed to be joined with geospatial data for analysis. In a GeoPackage, attributes data is stored in attributes tables as per <http://www.geopackage.org/spec/#attributes>.

- base data

Data that is linked in some way to related data (in other words, the left side of the $A \rightarrow B$ relationship). In this extension, base data is stored in geospatial or attributes data tables.

- cardinality

The property of a relationship between two entities, specifying whether it is one-to-one, one-to-many, many-to-one, or many-to-many.

- geospatial data

Data containing location information and/or geometries. In a GeoPackage, geospatial data may be stored in features or tiles tables.

- relationship

For the purposes of this extension, a link between two entities 'A' and 'B'. 'A' refers to base data and 'B' refers to related data.

- one-to-many

A type of cardinality in which an element of A may be linked to zero or more elements of B, but an element of B is linked to one and only one element of A.

- many-to-many

A type of cardinality in which an element of A may be linked to zero or more elements of B, and an element of B may be linked to zero or more elements of A.

- many-to-one

A type of cardinality in which an element of A is linked to one and only one element of B, but an element of B may be linked to zero or more elements of A.

- related data

Data that is linked in some way to base data (in other words, the right side of the $A \rightarrow B$ relationship). In this extension, related data is stored in a user-defined attributes table.

- user-defined attributes table

In this extension, a user-defined attributes table is a table that contains data that is related to existing geospatial data.

- user-defined mapping table

In this extension, a user-defined mapping table is a join table that links geospatial data and related data.

- user-defined media table

In this extension, a user-defined media table is a user-defined attributes table that is specifically designed to contain multimedia content.

3.1. Abbreviated terms

- GPKG GeoPackage
- RTE Related Tables Extension

Chapter 4. Overview

The purpose of the GPKG-RTE-IE is to demonstrate that it is possible to associate features, tiles, and attributes with other content within the GeoPackage. The goal of this Engineering Report (ER) is to present the work performed as part of the IE.

[Section 5](#) presents a conceptual overview of the Related Tables Extension, as well as a detailed discussion of the requirements classes and use cases.

[Section 6](#) presents the results of the individual Technology Integration Experiments (TIEs) that were performed as part of this IE.

[Section 7](#) presents a recap of discussion topics that arose during the IE.

Chapter 5. Design

5.1. Overview

The core of the Related Tables Extension is a mapping between existing table types defined by GPKG 1.2 - [features](http://www.geopackage.org/spec/#features) [http://www.geopackage.org/spec/#features], [tiles](http://www.geopackage.org/spec/#tiles) [http://www.geopackage.org/spec/#tiles], and [attributes](http://www.geopackage.org/spec/#attributes) [http://www.geopackage.org/spec/#attributes]. The mapping is defined by a new kind of table defined by the Related Tables Extension. The mapping table links related rows in those tables of those types by reference to their primary keys. For example, to link a row in Table A to a row in Table B, the mapping table includes a row that has two values - the primary key of the row from Table A, and the primary key of the row from Table B.

The mapping table allows many-to-many relationships. For example, to relate another row in Table B to the same row in Table A, the mapping table would simply include another row with the appropriate primary keys. See [Figure 1](#).

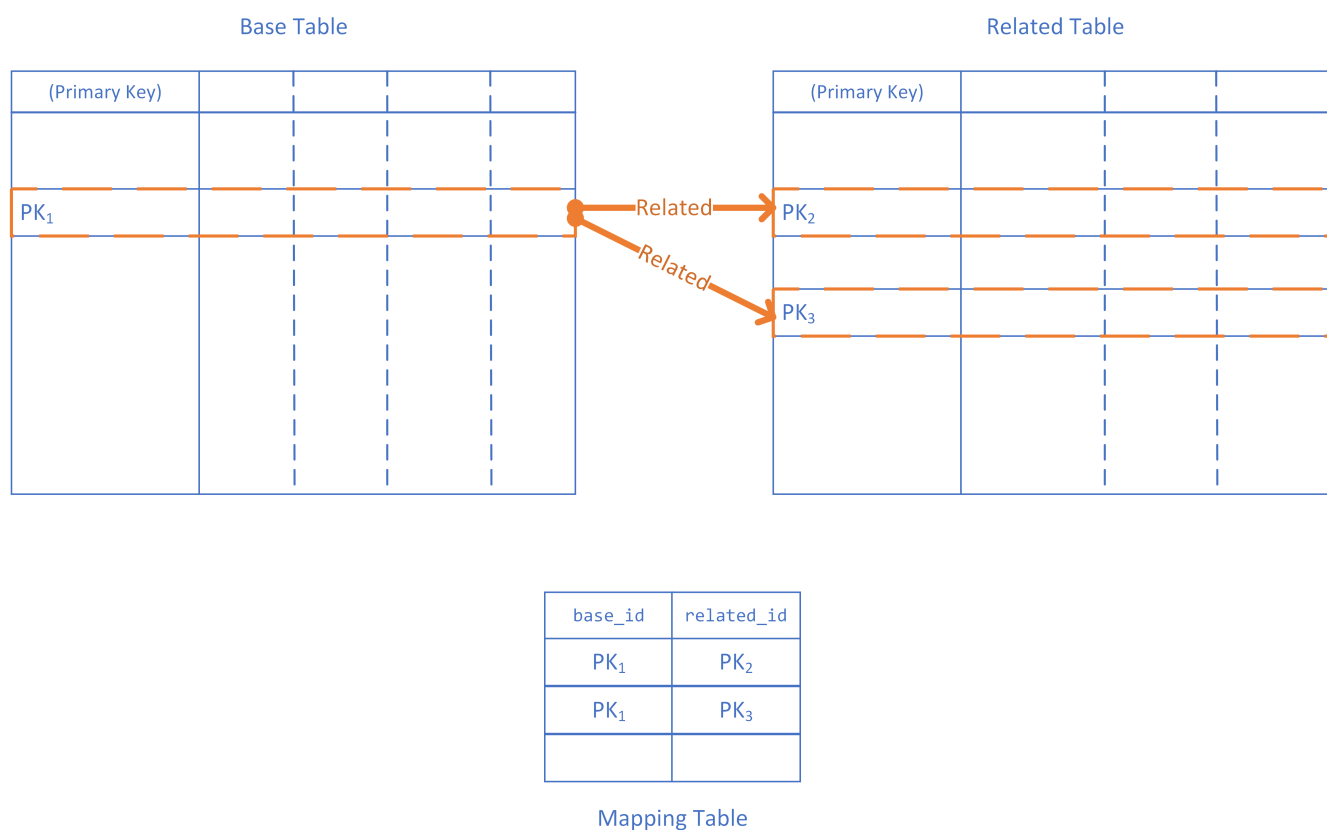


Figure 1. Related Tables concept

Mapping tables are unique to each pair of tables. The appropriate mapping table for each table pair (if any) is identified in a new table `gpkgext_relations`, which also specifies the name of the primary key column and the type of related data. This version of the Related Tables Extension supports three types of related data, which are separate conformance classes:

- media;
- simple attributes; and
- features.

The relationships can be considered directional in that they relate primary keys of two tables in terms of base (the "left" or "from" side of the mapping) and related (the "right" or "to" side of the mapping). Since the related tables are valid GPKG 1.2 table types (potentially with some additional constraints), they can form the base side of another mapping. This allows chaining (directed graph) of relationships as appropriate to represent the modelled data. See [Figure 2](#).

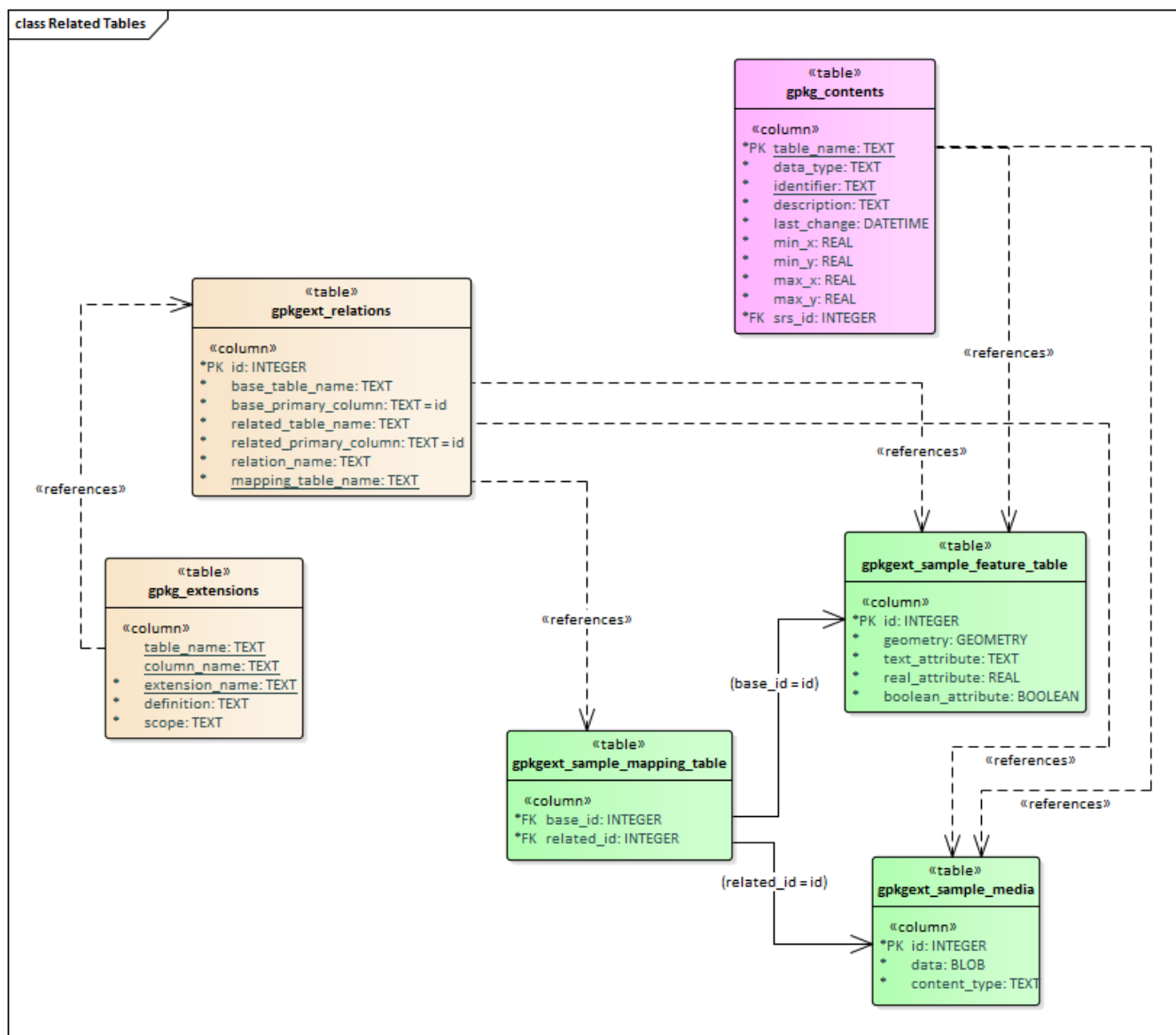


Figure 2. Related Tables UML Diagram

The Related Tables Extension makes no constraints on the base table; it can be any table type supported by GPKG 1.2 - tiles, attributes or features. The related ("right" / "to") table is constrained by defined values of `relation_name` which is a TEXT value in the `gpkgext_relations` table. The constraining of relationships serves two purposes - it allows clients to provide appropriate rendering of content and it communicates the intent of the relationship. Since the relationship is text, values other than those defined by the Related Tables Extension document can be used, however this will not be interoperable without some other coordination mechanism.

5.2. Requirements Classes

5.2.1. Media

The Media conformance class is used for related tables that provide multimedia content. The GPKG table type is [attributes](http://www.geopackage.org/spec/#attributes) [http://www.geopackage.org/spec/#attributes]. This was the original intent of the Related Tables Interoperability Experiment, and remains an important use. For example, using a `relation_name` of `media` provides the ability to link a set of photographs, line diagrams, documents, videos, and/or audio files to a specific location (typically a point or polygon feature; but the Related Tables Extension does not prohibit some other kind of feature, or a row in an attribute table, or a row in a tile table being used as the base side of the mapping to the media table). The minimum content of the user defined media table is a primary key, a BLOB containing the media content (conceptually a byte array in the GeoPackage), and the [IANA Media Type](https://www.iana.org/assignments/media-types/media-types.xhtml) [https://www.iana.org/assignments/media-types/media-types.xhtml] type for the media content (e.g., `image/jpeg` for a photograph).

An example of this is a land parcel (land lot) as the feature (base table), and photographs of the location (house, commercial property, etc.) as the related media.

Note that the related table does not need to include additional columns, although additional columns are permitted in the related table definition, so they can be added if desired. The Related Tables Extension does not constrain or codify what the additional columns can be. Specific communities of interest may wish to provide usage profiles of the Media conformance class to meet specific operational or business needs. Clients that intend to display GeoPackages that make use of the Media conformance class of the Related Tables Extension may wish to provide additional attribute display on a "best efforts" basis (e.g., view with the column names as labels for the text and numeric row values).

For example, additional column content might include:

- An indicator of the size of the media content (although this can be determined using the SQLite `length()` function);
- A title or description of the content of the media BLOB; or
- License information, usage restrictions, or security constraints.

5.2.2. Simple Attributes

The Simple Attributes conformance class is used for related tables that include only "simple attributes" - those SQLite values that are part of the TEXT, INTEGER and REAL [storage classes](https://www.sqlite.org/datatype3.html#storage_classes_and_datatypes) [https://www.sqlite.org/datatype3.html#storage_classes_and_datatypes]. The GPKG table type is [attributes](http://www.geopackage.org/spec/#attributes) [http://www.geopackage.org/spec/#attributes]. This is intended to support data that would typically be represented in Comma Separated Value (CSV) or spreadsheet formats, such as reference tables or observations. The simple attributes related table is not permitted to contain BLOB data (such as multimedia content or feature GEOMETRY - these are covered by other conformance classes).

Only two columns are required in the related attributes table - the primary key and one other column (which can be of TEXT, INTEGER, REAL, or a type that maps to one of those storage classes). As for Media, the Simple Attributes related table does not need to include additional columns, although additional columns are permitted in the related table definition, so they can be added if desired. The Related Tables Extension does not constrain or codify what the additional columns can

be. Specific communities of interest may wish to provide usage profiles of the Simple Attributes conformance class to meet specific operational or business needs. Clients that intend to display GeoPackages that make use of the Simple Attributes conformance class of the Related Tables Extension may wish to provide additional attribute display on a "best efforts" basis (e.g., view with the column names as labels for the text and numeric row values; or a spreadsheet-style table representation).

An example of this is a land parcel (land lot) as the feature (base table), and contact details for the managing agent as the related table. While this could be supported by embedding the contact details for each land parcel, this could be a lot of duplication and require update if a phone number or email address changes.

Note that the feature (base table) could link to many attribute table rows. An example of this would be for a set of valuations for the property, or records of property inspections or maintenance work conducted on the property.

5.2.3. Features

The Features conformance class is used for related tables that are [GPKG 1.2 vector feature tables](http://www.geopackage.org/spec/#feature_user_tables) [http://www.geopackage.org/spec/#feature_user_tables]. The GPKG table type is [features](http://www.geopackage.org/spec/#features) [http://www.geopackage.org/spec/#features]. This is intended to support defining relationships between feature types. No changes or constraints are made on the extant definition of the features tables.

An example of this is linking the location of a condominium (town house) or apartment with the locations of associated parking places or individual garden plots.

5.3. Usage scenario

A single GeoPackage could include each of these relationships. For example, an airport can be considered as a point location with some attributes, which would be represented in GeoPackage as a features table. Similarly, the runways may be considered as polygons with attributes, which would be represented in GeoPackage as a different features table. See [Figure 3](#). The mapping between those feature tables can be represented using the Related Tables Extension, so that a graphical user interface could identify and select the runways for a particular airport, including associated attributes and metadata.

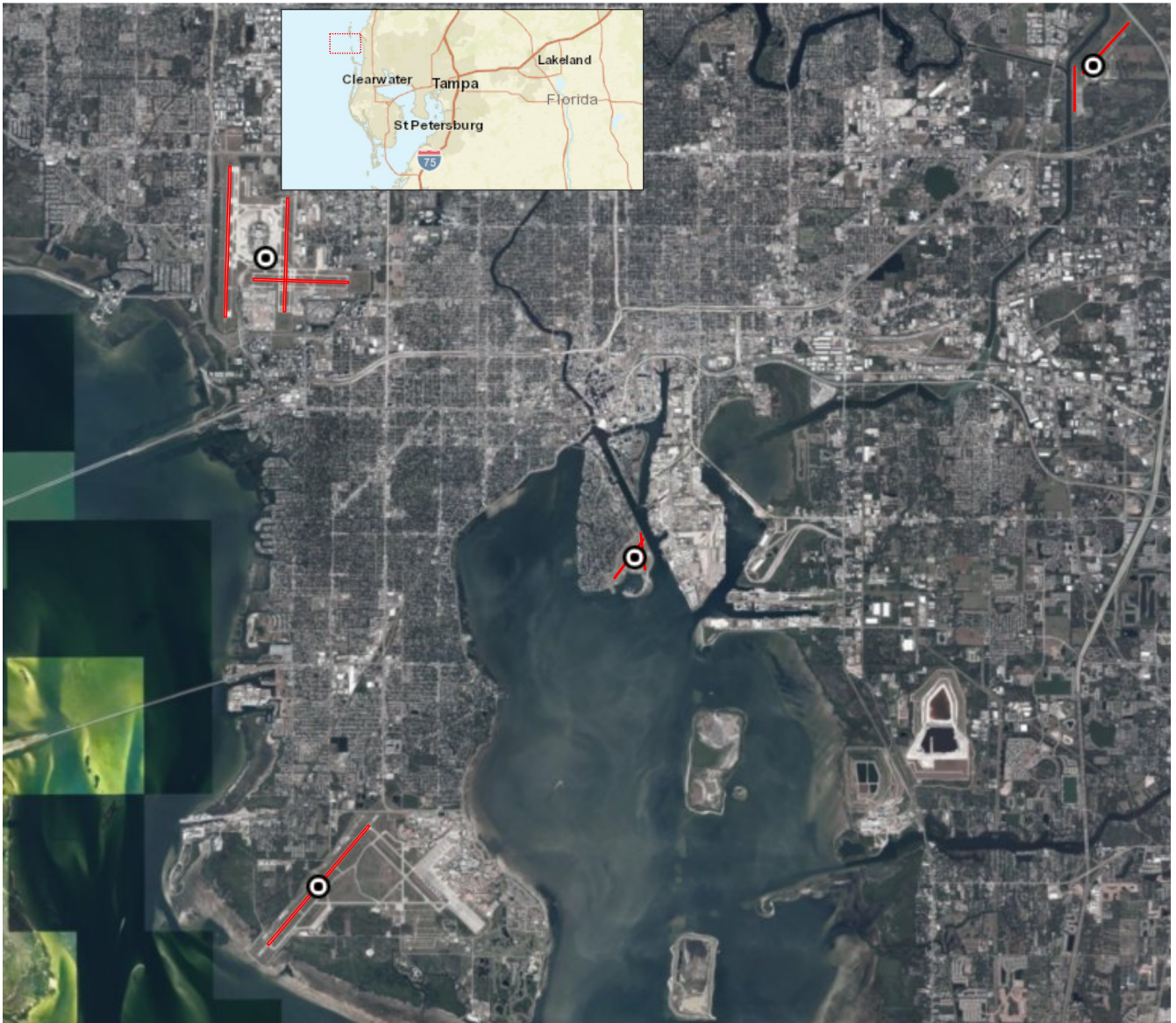
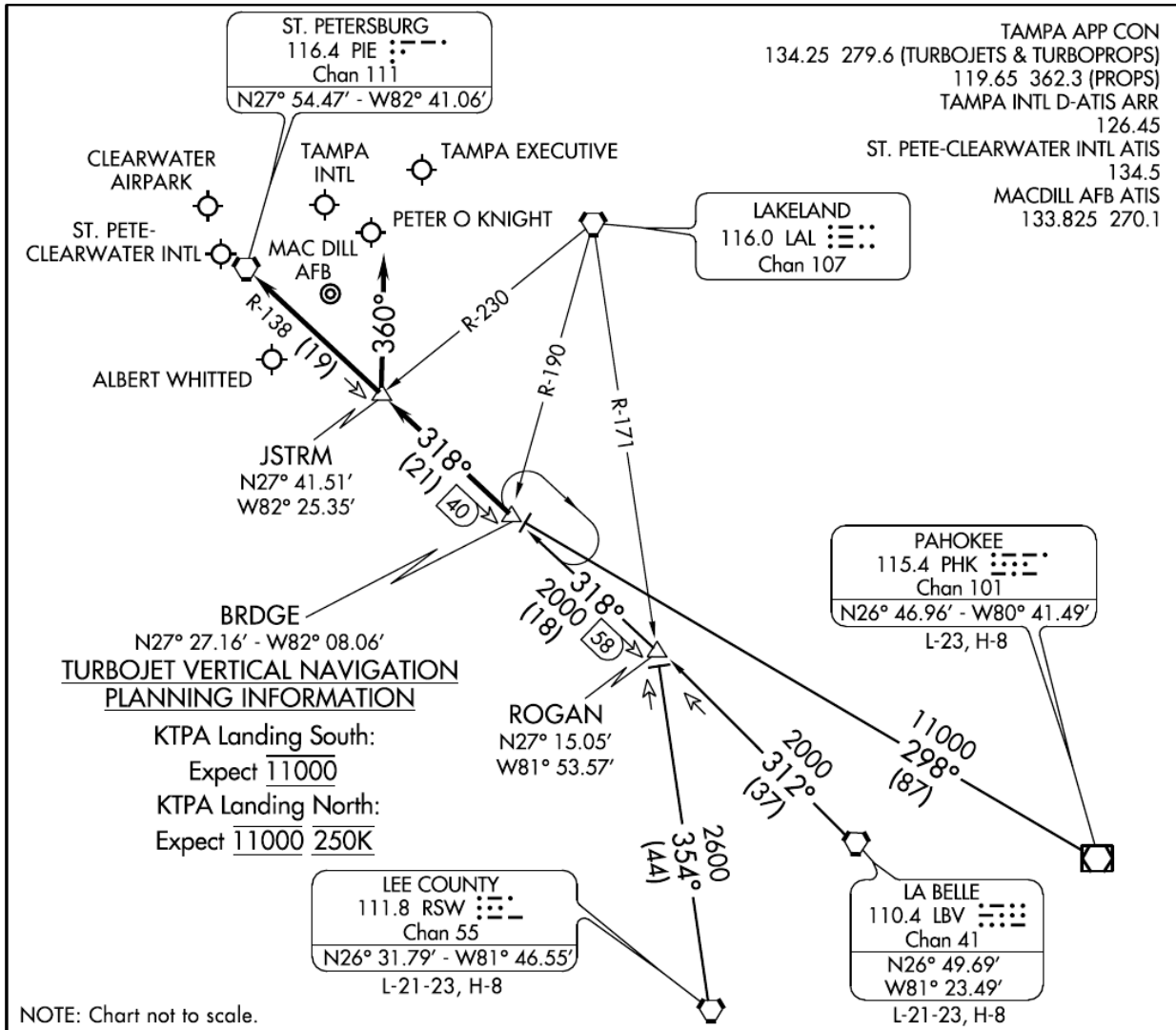


Figure 3. Airports and runways for Tampa area (from FAA data, AIRAC cycle 1802)

In addition to feature geometry, an airport may have associated documents, such as terminal procedures. These are typically provided as PDF documents containing a mix of text and diagrams, as shown in [Figure 4](#). These could be common to a range of airports in a close area (which is the case for that Arrival procedure), specific to a particular airport, or they could be specific to a particular runway, as shown in [Figure 5](#).

BRDGE EIGHT ARRIVAL

TAMPA, FLORIDA



ARRIVAL ROUTE DESCRIPTION

LA BELLE TRANSITION (LBV.BRDGE8): From over LBV VORTAC on LBV R-312 & PIE R-138 to BRDGE INT. Thence. . . .

LEE COUNTY TRANSITION (RSW.BRDGE8): From over RSW VORTAC on RSW R-354 to ROGAN INT, then on PIE R-138 to BRDGE INT. Thence. . . .

PAHOKEE TRANSITION (PHK.BRDGE8): From over PHK VOR/DME on PHK R-298 to BRDGE INT. Thence. . . .

KTPA:

. . . .RWY 19L/R: From over BRDGE INT on PIE R-138 to JSTRM INT. Depart JSTRM INT heading 360° for vector to final approach course.

. . . .RWY 01L/R: From over BRDGE INT on PIE R-138 to PIE VORTAC. Expect RADAR vector to final approach course after BRDGE INT.

KPIE, KCLW, KTPF, KSPG, KMCF, KVDF:

. . . .From over BRDGE INT on PIE R-138 to PIE VORTAC. Expect RADAR vector to final approach course/airport after BRDGE INT.

LOST COMMUNICATIONS: STANDARD.

BRDGE EIGHT ARRIVAL

TAMPA, FLORIDA

(BRDGE.BRDGE8) 27APR17

Figure 4. Bridge Eight Arrival (from FAA data, AIRAC cycle 1802)

WAAS CH 49100 W19A	APP CRS 187°	Rwy Idg 8300 TDZE 26 Apt Elev 26
--	------------------------	---

RNAV (GPS) Z RWY 19L

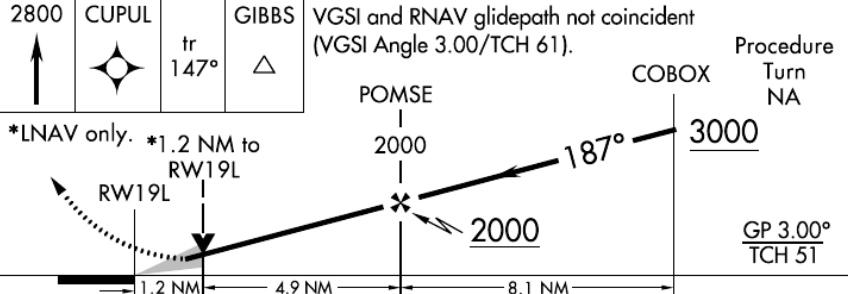
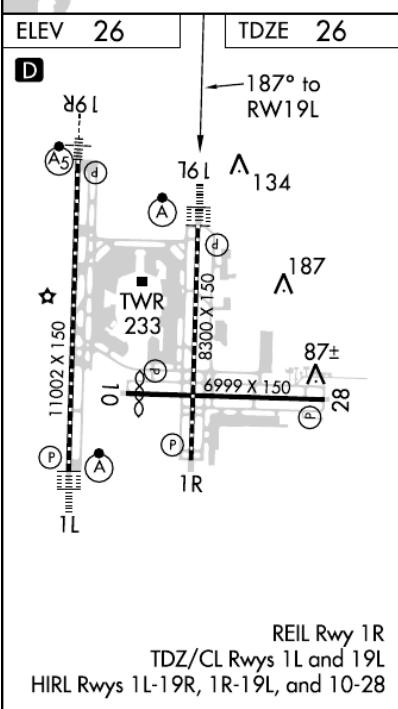
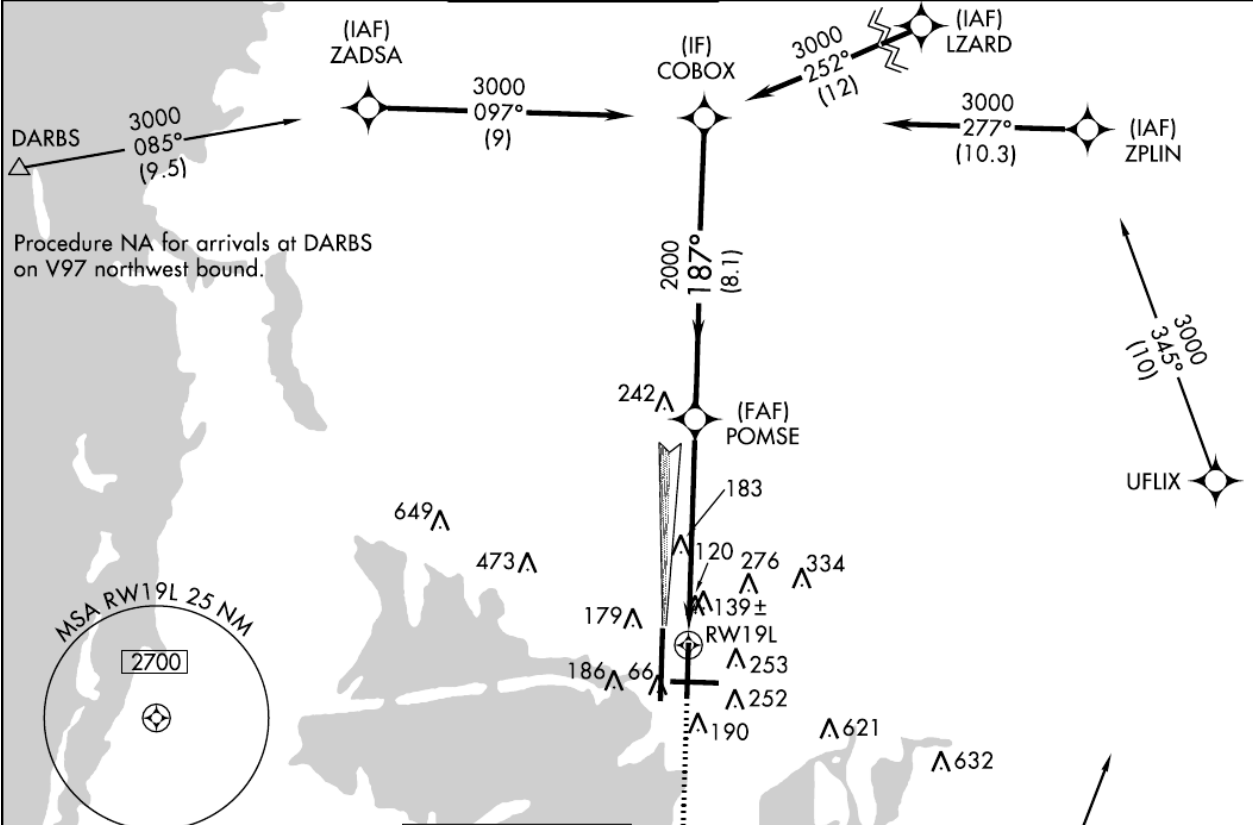
TAMPA INTL (TPA)

▼ For uncompensated Baro-VNAV systems, LNAV/VNAV NA below -15°C (5°F) or above 43°C (109°F). DME/DME RNP-0.3 NA. Simultaneous approach authorized with Rwy 19R. LNAV procedure NA during simultaneous operations. Use of FD or AP providing RNAV track guidance required during simultaneous operations. ** RVR 1800 authorized with use of FD or AP or HUD to DA.



MISSED APPROACH:
Climb to 2800 direct
CUPUL and on track
147° to GIBBS and hold.

ARR 126.45	D-ATIS 128.475	DEP 118.5 290.3	TAMPA APP CON 119.5 269.4	TAMPA TOWER 119.5 269.4	GND CON 121.7 269.4	CLNC DEL 133.6	CPDLC
----------------------	--------------------------	---------------------------	-------------------------------------	-----------------------------------	-------------------------------	--------------------------	-------



CATEGORY	A	B	C	D
LPV DA**		226/24	200 (200-1/2)	
LNAV/VNAV DA		440/45	414 (500-7/8)	
LNAV MDA	460/24	434 (500-1/2)	460/40	434 (500-3/4)
CIRCLING	560-1	534 (600-1)	560-1 1/2 534 (600-1 1/2)	580-2 554 (600-2)

Figure 5. Tampa International RWY 19L GPS Instrument Approach (from FAA data, AIRAC cycle 1802)

This information can be represented using the user defined media table, either by incorporating the original PDF as the content of the data BLOB, or by rendering it to an image format such as PNG and using that as the content of the data BLOB. Mapping tables can relate the feature tables (e.g., airport geometry as points and runways as polygons, as described above) to the media table.

Airports may also have associated attributes that are not geospatial or media, such as the communications frequencies that are required. There are often multiple frequencies and they are often common to multiple airports in an area. The frequency information can be represented as an attributes table, with the mapping from airport to communication frequency through a simple attributes mapping. There could well be additional attribute information, such as a mapping from the terminal procedures media table to currency (validity dates, last change) or to the responsible information provider and associated contact details.

Chapter 6. Technology Integration Experiments

This section discusses the Technology Integration Experiments (TIEs) conducted for the Related Tables Extension.

6.1. Test Samples and Descriptions

Thirteen samples were contributed by seven different providers, and covered a range of use cases and media types. All samples were assessed for alignment with the [Abstract Test Suite \(ATS\)](https://github.com/jyutzler/geopackage-related-tables/blob/master/standard/annex-ats.adoc) [https://github.com/jyutzler/geopackage-related-tables/blob/master/standard/annex-ats.adoc] and tested with an Executable Test Suite (ETS) to verify conformance with the draft specification.

Table 2. Test Samples used in final TIEs

Sample Number	Title	Description	Requirements Classes
1	Audio - simple example	Points in Nepal linked with one attributes table	Media (audio/wav)
2	Audio+Cats	Four point features with two attributes tables	Media (image/jpeg, audio/wav)
3	cats.gpkg	Four point features with one attributes table	Media (image/jpeg)
4	FAA Airport Features	FAA airports and runways linked with related table features type	Features
5	GDAL Sample v1.2 plus RTE	GDAL-generated v1.2 GeoPackage from geopackage.org + many-to-many RTE example	Media (image/jpeg, image/png)
6	Puerto Rico mixed sample	Mix of tiles, features and images	Media (application/pdf, image/jpeg, image/tiff)
7	Simple Attributes example	Minimal example	Simple Attributes
8	photos_rte	Mock example of plague sites with photo media	Media (not compliant)

Sample Number	Title	Description	Requirements Classes
9	USATAMPA.gpkg	Example with larger number of features and related media tables (structures, routes, photos, etc.)	—
10	birds	Points over Midway island with video, audio, and images. Video table includes two different MIME types	Media (audio/m4a, video/quicktime, application/mp4, image/jpeg)
11	hack-n-hunt plus photos	SOFWERX Hack-n-Hunt event map + photos	Media (image/jpeg)
12	zerozero	Polygons + photos, many-to-many relationships	Media (image/jpeg, image/png)
13	safehouse_plus	Tiles, vector features, RTE media and x-relation types	Media, also x-extension

6.2. ETS Results

The existing GeoPackage 1.2 Executable Test Suite was extended to validate the four Related Tables Extension conformance classes. The additional code is believed to cover all ATS requirements, except for checks for UNIQUE constraints on tables. Not having verification of UNIQUE constraints was undesirable, but did not significantly affect the value of ETS testing. In particular, early access to the ETS allowed correction of minor issues in some samples.

Table 3. Test Sample ETS results

Sample Number	Title	Summary (Total/Passed/Skipped/Failed)	Core	Features	Tiles	Attributes	Extension Mechanism	RTE Core	RTE Media	RTE Simple Attributes	RTE Features	Notes
1	Audio - simple example	179/57/122/0	Pass	Pass	N/A	Pass	Pass	Pass	Pass	N/A	N/A	-

Sam ple Num ber	Title	Sum mary (Tota l/Pas s/Ski pped/ Faile d)	Core	Feat ures	Tiles	Attri butes	Exte nsion Mech anis m	RTE Core	RTE Medi a	RTE Simp le Attri butes	RTE Feat ures	Note s
2	Audi o+Cat s	179/5 7/122/ 0	Pass	Pass	N/A	Pass	Pass	Pass	Pass	N/A	N/A	-
3	cats.g pkg	179/5 4/125/ 0	Pass	Pass	N/A	Pass	Pass	Pass	Pass	N/A	N/A	-
4	FAA Airpo rt Featu res	179/5 5/124/ 0	Pass	Pass	N/A	Pass	Pass	Pass	N/A	N/A	Pass	-
5	GDAL Samp le v1.2 plus RTE	179/0/ 179/0	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	All tests skipp ed
6	Puert o Rico mixe d samp le	179/8 1/98/0	Pass	Pass	Pass	Pass	Pass	Pass	Pass	N/A	N/A	-
7	Simpl e Attrib utes exam ple	179/5 7/122/ 0	Pass	Pass	N/A	Pass	Pass	Pass	N/A	Pass	N/A	-

Sam ple Num ber	Title	Sum mary (Tota l/Pas s/Ski pped/ Faile d)	Core	Feat ures	Tiles	Attri butes	Exte nsion Mech anis m	RTE Core	RTE Medi a	RTE Simp le Attri butes	RTE Feat ures	Note s
8	photo s_rte	179/5 7/121/ 1	Fail	Pass	N/A	N/A	Pass	N/A	N/A	N/A	N/A	Missi ng SRS entry for core. Does not inclu de gpkge xt_rel ation s or decla re RTE.
9	USAT AMP A.gpk g	179/6 9/108/ 2	Pass	Pass	N/A	N/A	Pass	N/A	N/A	N/A	N/A	Old exam ple, not updat ed. Two meta data test failur es for refer ence table s not in conte nts table.
10	birds	179/5 7/122/ 0	Pass	Pass	N/A	Pass	Pass	Pass	Pass	N/A	N/A	-

Sam- ple Num- ber	Title	Sum- mary (Total/ Pass/ Skipped/ Failed)	Core	Feat- ures	Tiles	Attri- butes	Exte- nsion Mech- anis- m	RTE Core	RTE Medi- a	RTE Simp- le Attri- butes	RTE Feat- ures	Note s
11	hack- n- hunt plus photo s	179/0/ 179/0	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	All tests skipp- ed
12	zeroz ero	179/0/ 179/0	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	Fault	All tests skipp- ed
13	safeh- ouse_ plus	179/8 6/91/2	Pass	Pass	Pass	Pass	Pass	Fail	Pass	N/A	N/A	No defau- lts provi- ded for gpkge xt_rel- ation- s. Also a RTree defini- tion was incor- rect.

Note that the ETS includes tests for extensions that were not part of this IE (e.g., schema extension).

The tests that faulted are likely a bug in the ETS or a support library. Those files can be opened in other GeoPackage readers, and in SQLite. The problem was reproduceable on the OGC TE2 (beta) test site, and is not directly related to the ETS extensions for RTE.

6.3. TIE Results for RTE-Aware Software

TIEs were conducted for five software packages that had implemented the RTE. Negative tests with non-compliant samples were also included, as it is useful to know whether particular non-compliant samples work or not.

Table 4. Experiment 1: TIE Pairings for RTE-Aware Software

Sample Number	Software 1	Software 2	Software 3	Software 4	Software 5
1	successful test	successful test		successful test	successful test [7]
2	successful test	successful test		successful test	successful test [7]
3	successful test	successful test		successful test	successful test [7]
4	successful test	successful test [3a]		failed test [1a]	successful test
5	successful test	failed test [1b]	successful test	failed test [1b]	failed test [1b]
6	successful test	successful test [3b]		successful test	failed test [8]
7	successful test [3a]	successful test [3a]		successful test	successful test
8	failed test [5]	failed test [1b]		failed test [2]	failed test [5]
9	successful test	failed test [4]		failed test [2]	successful test
10	successful test [6]	successful test [3a]		successful test	successful test [7]
11	successful test	failed test [5]	successful test	failed test [2]	failed test [2]
12	successful test [3a]	failed test [5]	successful test	successful test	failed test [9]
13	successful test	successful test [3a]	successful test	failed test [1b]	failed test [8]

[1a] Failed because software did not anticipate that related table could be a features table (was expecting an attributes table)

[1b] Failed because related data was an unknown type (aspatial)

[1c] Failed because software did not anticipate that base table could be a attributes table (was expecting a features table)

[2] Failed because software did not detect the extension - no gpkg_extensions entry

[3a] Success - Media not rendered, but referenced

[3b] Success - No errors, but software was incapacitated due to data processing

[4] Failed - Software crashed; file threw an exception due to its size

[5] Failed - Nothing to show

[6] Success - Related media was discovered but the file names were "null"

[7] Success - Related media was discovered but only image/jpeg content displayed

[8] Failed - Invalid WKT string reported

[9] Failed - Error during load: "attempt to write a readonly database"

6.4. TIE Results for non-RTE-Aware Software

In addition to TIEs for software that had implemented the RTE, two non-RTE-aware software packages were also tested to ensure that the extension is backward compatible. Negative tests with non-compliant samples were also included, as it is useful to know whether particular non-compliant samples work or not.

Table 5. Experiment 2: TIE Pairings for non-RTE-Aware Software

Sample Number	Software 5	Software 6
1	successful test	successful test
2	successful test	successful test
3	successful test	successful test
4	successful test	successful test
5	successful test	successful test
6	successful test	successful test
7	successful test	successful test
8	shows up / however contents empty	ERROR 1: bad result for PRAGMA foreign_key_check, got 18 rows, expected 0 ERROR 1: pragma foreign_key_check on 'Simple_Example.gpkg' failed. You can disable this check by setting the OGR_GPKG_FOREIGN_KEY_CHECK configuration option to NO
9	successful test	successful test
10	successful test	successful test
11	successful test	successful test

Sample Number	Software 5	Software 6
12	shows up / however contents empty	<p>ERROR 1: bad result for PRAGMA foreign_key_check, got 15 rows, expected 0</p> <p>ERROR 1: pragma foreign_key_check on 'zerozero (1).gpkg' failed. You can disable this check by setting the OGR_GPKG_FOREIGN_KEY_CHECK configuration option to NO</p> <p>gdalinfo failed - unable to open 'zerozero (1).gpkg'.</p>
13	successful test	successful test

Chapter 7. Discussion of Design Decisions and Ramifications

7.1. Use Cases

As part of the development of one software package, the developer defined a set of use cases. The API for the software package was designed to have a method for each use case.

1. Inspect whether the GeoPackage has the Related Tables Extension.
2. Add the Related Tables Extension (including all required tables and `gpkg_extensions` rows) to the GeoPackage.
3. Remove the Related Tables Extension (including all required tables, relationships, mappings, and `gpkg_extensions` rows) from the GeoPackage.
4. Get all registered relationships, according to the contents of `gpkgext_relations`.
5. Add a relationship based on the base table name, related table name, mapping table name, and relationship name (the base primary key name and related primary key name can be detected).
6. Remove a relationship based on the base table name, related table name, and relationship name.
7. Add a mapping (base ID and related ID) for a relationship (note that technically it is possible for a mapping table to have duplicate mappings).
8. Delete a mapping for a relationship based on its base ID and related ID (note that this would remove duplicate rows if they exist).
9. Given the ID of a base data row, get the IDs (primary keys) of all of the related data for the given relationship.
10. Given the ID of a related data row, get the IDs (primary keys) of all of the base data for the given relationship.

7.2. Semantics

One of the challenges the participants worked to address is the semantics of the related data. This can be handled in a few ways but the participants did not reach a consensus on a recommended approach.

7.2.1. Relation Name

The first way to define the semantics of a relationship is through the `relation_name` of `gpkgext_relations`. The main use of this column is to identify which requirements class is being used (media, features, etc.). This approach does not allow users to differentiate between multiple instances of a specific relation type. For example, `media` could refer to audio or video files and there could be separate relations for both.

7.2.2. Mapping Table Name Use with Schema Extension

The second way to define the semantics of a relationship is through the mapping table name. A relationship is uniquely identified by its mapping table name but this name should probably be terse for usability reasons. While this name often will not provide any useful semantics, the necessary semantics could be provided through the GeoPackage [Schema Extension](http://www.geopackage.org/spec120/#extension_schema) [http://www.geopackage.org/spec120/#extension_schema]. This extension is designed to describe the columns of tables in a GeoPackage with more detail than can be captured by the SQL table definition directly. One possible use of the Schema Extension is to describe a specific column value. If the Schema Extension is in use, it would allow a user to add a title and/or description for the `mapping_table_name` by adding a row to `gpkg_schema` as shown in [gpkg_schema column values for describing a relationship](#).

Table 6. `gpkg_schema` column values for describing a relationship

Column Name	Column Value
<code>table_name</code>	<code>gpkgext_relations</code>
<code>column_name</code>	<code>mapping_table_name</code>
<code>name</code>	<code>mapping_table_name</code> value
<code>title</code>	title (i.e., short description)
<code>description</code>	description (i.e., long description)
<i>other values</i>	<code>null</code>

This approach was not tested during this interoperability experiment.

7.2.3. Metadata

The third way to define the semantics of a relationship is through the Dublin Core Profile. Using [Dublin Core elements](http://dublincore.org/documents/dcmi-terms/) [http://dublincore.org/documents/dcmi-terms/] on either the mapping table or the related data table, it is possible to identify many aspects of a relationship. This approach is described in [Annex C](https://github.com/jyutzler/geopackage-related-tables/blob/master/standard/annex-profile.adoc) [https://github.com/jyutzler/geopackage-related-tables/blob/master/standard/annex-profile.adoc] of the draft standard. It was not tested during this interoperability experiment.

7.3. Cardinality

Sponsors have identified a need to handle all possible cardinalities between base data and related data – one-to-many, many-to-one, and many-to-many. The use of a mapping table allows any of these relationships.

Participants agreed to make the mapping table mandatory as part of this extension. This would create a single code path for the software to follow, joining the base table, the mapping table, and the related data table. While a one-to-many relationship could be handled without this table structure (a foreign key relationship would suffice), this would require a foreign key in the related data table. This would restrict the related data to a single related data relationship.

Participants recognized that there was a potential need to relate multiple sets of related data to a single base table. The participants determined that this could be handled naturally by not constraining the number of relations that are registered with a particular base data table. Similarly,

it would be possible to chain relationships together. For example, an airport feature could relate to multiple runways which could then relate to multiple media files.

One issue came up that was not resolved during the experiment. It would be useful to identify the type of relationship that is represented to clarify the expected output schema on export. However, the participants could not agree on a solution - each proposal seemed to cause more problems than it solved. For example, a `cardinality` column was proposed for the `gpkgext_relations` table but this was rejected because it would be too easy for a GeoPackage client to ignore the value and add relationships that were inconsistent with it. Adding constraints to compensate for this issue would have required extensive error handling which was determined to not be worth the effort.

7.4. External References

The participants agreed that there was value to storing references to related content, as opposed to embedding the content as a BLOB inside the database. Some media files (particularly high resolution full motion video) can get quite large and SQLite databases do not scale very well past a few GB. In the interest of time, the participants elected to make this out of scope for the IE. However, the decision to have separate requirements classes for different relation types provides a potential way ahead in this area. An "external media" requirements class could potentially have a URI column instead of a BLOB column.

It was noted that in an operational setting, it is useful to separate large, static GeoPackages from smaller, dynamic GeoPackages because smaller files are easier to manage. This is consistent with other prior discussions.

7.5. Handling of Unregistered MIME Types

Because the Media related attribute tables must contain a `content_type` that is not null, the participants discussed how to handle commonly-used media types that are not registered IANA MIME types (e.g., `audio/wav` is not a defined IANA media type, even though it is commonly used). The participants agreed to maintain the requirement that `content_type` cannot be null, as producers can fall back on the option to specify `application/octet-stream` for documents without a more specific, registered MIME type.

Future work could involve adjusting this requirement in a way that maintains testability and interoperability while enabling an improved user experience.

7.6. Implications of `gpkg_extensions` Requirements for Non-Conforming Clients

Because the extension requires adding rows for `gpkgext_relations` and the mapping table(s) to the `gpkg_extensions` table, the participants noted that could cause interoperability issues for non-conforming clients. For example, if the client deleted a row in the related table, it would result in a row in the mapping table where the target primary key did not exist.

This issue was not fully resolved; one suggested approach to address this would be to require that the rows for `gpkgext_relations` and the mapping table(s) have a scope of `write-only`. Alternately,

implementors could be required to deal with unexpected disappearance/alteration of base table rows.

7.7. User-Defined Mapping Table UUIDs

The columns (`base_id` and `related_id`) in a user-defined mapping table are required to be `INTEGER` values, which is consistent with primary key requirements for GeoPackage user tables. After running the TIEs, one participant provided feedback that this requirement may unnecessarily force additional logic to be implemented by applications that use text UUIDs as keys in the database, for example to support syncing GeoPackages between devices for mobile disconnected operations. They requested future consideration for changing the mapping table to require two `TEXT` fields instead of two `INTEGER` fields.

7.8. Preventing Data Duplication

One participant noted that future work could include the ability to prevent data duplication for cases where the same content is linked to different features. This could be done by implementing a content reuse feature where the user could browse and select currently attached content, or a field for a checksum for the content, to enable an application to more easily verify if content being added already exists in the GeoPackage.

Appendix A: Revision History

Table 7. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
February 12, 2018	J. Yutzler	.1	all	Initial version
May 8, 2018	A. Antonides	.8	all	Working draft for review; updated theme to OGC Testbed 14 version
May 16, 2018	A. Antonides	.9	Summary, TIEs, Discussion, Bibliography	Updated draft with final TIEs for review
May 17, 2018	A. Antonides, J. Yutzler	1.0	all	Inline links, final revisions
May 24, 2018	A. Antonides	1.1	Discussion, TIEs	Additional discussion and updated TIEs