

# Data format of a back-country ski touring network

## Technical Specification

Günter Schmudlach

12. September 2025

(V1.0.3)

### Abstract

The document specifies the **required data format** of a **back-country ski touring network**.

### Content

1	Introduction.....	2
2	Data description.....	2
3	Route Composition.....	2
3.1	Identifier (id).....	3
3.2	Segments (segments).....	3
3.3	Start name (start).....	3
3.4	Stop name (stop).....	4
3.5	Route Name (name).....	4
3.6	Start name in a particular language (start_xx).....	4
3.7	Stop in a particular language (stop_xx).....	5
3.8	Route name in a particular language (name_xx).....	5
3.9	Manual Difficulty (mdiff).....	6
3.10	Manual Ski Depot Distance (msdd).....	6
3.11	Type (type).....	7
3.12	Manual Direction Type (mdir).....	7
3.13	Target (target).....	8
3.14	Skitour (ski).....	8
3.15	Snowshoe (snowshoe).....	9
3.16	Relevance (relevance).....	9
3.17	Comment (comment).....	9
3.18	Region (region).....	10
4	Segments.....	10
4.1	Identifier (id).....	10
4.2	Segment name (name).....	11
4.3	Segment name in a particular language (name_xx).....	11
4.4	Skitour (ski).....	11
4.5	Snowshoe (snowshoe).....	12
4.6	Foot section (foot).....	12
4.7	Extra caution section (caution).....	12
4.8	DIN norm (din).....	13
4.9	Manual Direction Type (mdir).....	13
4.10	Target (target).....	14
4.11	Comment (comment).....	14
4.12	Owner (owner).....	15

# 1 Introduction

In order to work with data related to back-country ski touring a data format is required, that must fulfill two criteria:

- The data format must be able to represent a network of interconnected **segments**.
- The data format must be able to represent a **collection of routes**. A route is set of several connected segments.

## 2 Data description

Data are stored in any [OGR-compliant vector format](#). As the format should be able to handle two layers (segments and routes) priority should be given to formats, that can handle multiple layers. To be mentioned three formats with different purposes:

- Geopackage ([GPKG](#)): For local usage.
- Post-GIS ([PostgreSQL](#)): To be used in a client-server architecture.
- GeoJson ([GeoJSON](#)): A text-based format to be used on a Git-Repository. Two files will be needed. On a Git repository this is the factorized format. Make sure coordinates are in a meter CRS and have only two decimal digits.

The data format must hold two layers:

- Segments: A layer of segment features. A segment feature has a series of attributes (see chapter 4 ) and a line string geometry. It is not allowed to add a "MultiLineGeometry" to the feature. The geometry line string has two ends. Each end can only end at another "segment end" or in the "nowhere". A "segment end" can never end in the middle of another segment.
- RouteComposition: A layer of "route composition" features. Each feature represents a route. A route is defined by a sequence of connected segments. A composition feature has a series of attributes (see chapter 3 ) but no geometry. The Attribute "segments" contains a list of segment identifiers.

If segments and compositions are merged to a collection of routes, other, so called automatic attributes can be added.

## 3 Route Composition

A layer of "route composition" features. Each feature represents a route. A route is defined by a sequence of connected segments. A segment feature has a series of attributes (see chapter 3 ) but no geometry. The Attribute "segments" contains a list of segment identifiers.

The following chapters describe attributes that can be entered manually. Attributes, that can be calculated automatically are not listed here.

### 3.1 Identifier (id)

Name	id	Identifier
Description	A unique number that identifies the route	
Class	Mandatory	
Comment	<ul style="list-style-type: none"><li>• In order to avoid broken links, don't change the id of a published route.</li><li>• An id of a deleted route must remain reserved for that particular route. Don't recycle id's.</li><li>• The number range (1...9999) should not be exceeded.</li><li>• The id must be unique. Can be tested with qGis field editor: <code>count(1,"id") &gt; 1</code></li><li>• There is no need that the number is identical to the format-owned attribute OGC_FID.</li></ul>	
Type	Integer	
Values	1..9999	

### 3.2 Segments (segments)

Name	segments	Identifier
Description	A list of segments identifiers	
Class	Mandatory	
Comment	<ul style="list-style-type: none"><li>• Each segment identifier points to a feature of chapter 4</li><li>• The segments must be inter-connectable in the order of the list.</li><li>• The resulting route should be digitized in an upwards-direction: That means the start of the first segment should have a lower elevation then the end of the last segment. This rule is not mandatory.</li></ul>	
Type	Coma-seperated list of integers, forming a string.	
Values		
Example	67,785,432	

### 3.3 Start name (start)

Name	start	Start name
Description	The common name of the start place of the route	
Class	Mandatory	
Comment	<ul style="list-style-type: none"><li>• Usually a <b>village</b> name.</li><li>• In case the place has names in many languages, apply the most common name.</li></ul>	
Type	String	
Values		
Example	Rippje	

### 3.4 Stop name (stop)

Name	stop	Stop name
Description	The common name of the stop place of the route	
Class	Mandatory	
Comment	<ul style="list-style-type: none"><li>Usually a <b>summit</b> or <b>col</b> name.</li><li>In case the place has names in many languages, apply the most common name.</li></ul>	
Type	String	
Values		
Example	Ofenhorn	

### 3.5 Route Name (name)

Name	name	Route name
Description	The common name of the route	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>Leave empty if the route doesn't has a name. Most routes don't have names.</li><li>In case the route has names in many languages, apply the most common name.</li></ul>	
Type	String	
Values		
Example	Normalroute	

### 3.6 Start name in a particular language (start\_xx)

Name	start_xx	Start name in a particular language
Description	The common name of the start place of the route in the language xx	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>Usually a <b>village</b> name.</li><li>Use the language tags according to <a href="#">ISO 639</a>.</li><li>Don't copy the default name</li></ul>	
Type	String	
Values		
Example	"Rippje" for stop_de	

### 3.7 Stop in a particular language (stop\_xx)

Name	stop_xx	Stop name in a particular language
Description	The common name of the stop place of the route in the language xx	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>• Usually a <b>summit</b> or <b>col</b> name.</li><li>• Use the language tags according to <a href="#">ISO 639</a>.</li><li>• Don't copy the default name</li></ul>	
Type	String	
Values		
Example	"Pizzo d'Arbola" for stop_it	

### 3.8 Route name in a particular language (name\_xx)

Name	name_xx	Route name in a particular language
Description	The common name of the route in the language xx	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>• Leave empty if the route doesn't has a name. Most routes don't have names.</li><li>• Use the language tags according to <a href="#">ISO 639</a>.</li><li>• Don't copy the default name</li></ul>	
Type	String	
Values		
Example	"La Rampa" for route_es	

### 3.9 Manual Difficulty (mdiff)

Name	mdiff	Manual Difficulty
Description	Difficulty degree of the route expressed by the SAC difficulty scale ( <a href="#">Schwierigkeitsgrade</a> ).	
Class	Optional	
Comment	<ul style="list-style-type: none"> <li>Difficulty degree assigned by an expert (<b>manual</b> assignment).</li> </ul>	
Type	Integer	
Values	0 ( <b>default</b> ): Unknown 1: F 2: F+ 3: PD- 4: PD 5: PD+ 6: AD- 7: AD 8: AD+ 9: D- 10: D 11: D+ 12: TD- 13: TD 14: TD+ 15: ED- 16: ED 17: ED+ 18: EX	

### 3.10 Manual Ski Depot Distance (msdd)

Name	msdd	Manual Ski Depot distance
Description	Projected, horizontal distance in meters of the <b>ski depot</b> from the stop location.	
Class	Optional	
Comment		
Type	Real	
Values	0 ( <b>default</b> ): No ski depot was assigned. -1: There is no ski depot, respectively its possible to reach the summit by skis. >0: A ski depot was assigned.	

### 3.11 Type (type)

Name	type	Type
Description	The route type	
Class	Optional	
Comment		
Type	Integer	
Values	0 ( <b>default</b> ): A normal route that starts from a place accessible by car or public transport or cable car station. 1: A route that stops at a place not accessible by car or public transport or cable car station (usually a mountain hut). 2: A route that starts from a place not accessible by car or public transport or cable car station (usually a mountain hut).	

### 3.12 Manual Direction Type (mdir)

Name	mdir	Manual Direction
Description	Preferable travel direction, recorded manually.	
Class	Optional	
Comment	Its recommended to digitize always in the major direction. Usually that's from bottom (low elevation) to top (high elevation).	
Type	Integer	
Values	0 ( <b>default</b> ): Both directions. 1: Upwards 2: Downwards	

### 3.13 Target (target)

Name	target	Target
Description	The target of the route. A target is a platform, where the route can be published. Each platform is defined by a letter. If the letter is uppercase, the route can be published. If the letter is lowercase, its not allowed to publish the route.	
Class	Optional	
Comment	<p>Allowed platform tags:</p> <p>S: SkitourenGuru  P: Panico Books  O: OutdoorActive  M: Alpenvereins-Karten  I: IGN France  T Swisstopo  X: No platform</p> <p>Special cases:</p> <ul style="list-style-type: none"> <li>If the field is empty or null, the route can be published on all platforms (<b>default</b>)</li> <li>Use the characters "X" or "x", if no platform is allowed to publish the route.</li> </ul> <p>Remarks</p> <ul style="list-style-type: none"> <li>Its not allowed to mix uppercase and lowercase letters</li> </ul>	
Type	String	
Values		
Examples	m: Publish on all platforms except on Alpenverein-Karten PS: Publish only on Panico and SkitourenGuru iP: Combination is not allowed: null: Publish on all platforms empty string: Publish on all platforms x: Not to be published at all X: Not to be published at all	

### 3.14 Skitour (ski)

Name	ski	Skitour
Description	Defines whether the route is suitable as a <b>ski</b> tour (1) or not (0).	
Class	Optional	
Comment		
Type	Integer	
Values	0: Not suitable as a ski tour 1( <b>default</b> ): Suitable as a ski tour	



### 3.15 Snowshoe (snowshoe)

Name	snowshoe	Snowshoe
Description	Defines whether the route is suitable as a <b>snowshoe</b> tour (1) or not (0).	
Class	Optional	
Comment		
Type	Integer	
Values	0 ( <b>default</b> ): Not suitable as a snowshoe tour 1: Suitable as a snowshoe tour	

### 3.16 Relevance (relevance)

Name	relevance	Relevance
Description	Defines the degree of relevance of the route. Expressed in number of stars (0..5).	
Class	Optional	
Comment	Obviously there are no objective criteria for the number of stars to be given to a route.	
Type	Integer	
Values	0: A route with hooks and eyes 1: 2: 3: 4: 5: A very beautiful route.	

### 3.17 Comment (comment)

Name	comment	Comment
Description	Any comment in any language	
Class	Optional	
Comment		
Type	String	
Values		

### 3.18 Region (region)

Name	relevance	Relevance
Description	Defines the geographic region of the routes	
Class	Optional	
Comment	This value can be used to calculate unique id's. In order to produce consistent unique id's over all platforms we suggest to update the id's as follows: $id = id * 10 + \text{region}$	
Type	Integer	
Values	1: Switzerland (CH) 2: Austria (AT) 3: France (FR) 6: Italy (IT) 7: Germany (DE) 8: Slovenia (SI)	

## 4 Segments

A layer of segment features. A segment feature has a series of attributes (see chapter 4 ) and a line string geometry. Its not allowed to add a "MultiLineGeometry" to the feature. The geometry line string has two ends. Each end can only end at an other "segment end" or in the "nowhere". A "segment end" can never end in the middle of an other segment. A segment can't cross an international border. However it can follow a border.

The following chapters describe attributes that can be recorded manually. Attributes, that can be calculated automatically are not listed here.

### 4.1 Identifier (id)

Name	id	Identifier
Description	A unique number that identifies the segment.	
Class	Mandatory	
Comment	The segment must be in the range 1'000'000 to 9'999'999. The first digit designs the country: <ul style="list-style-type: none"><li>• CH/FL: 1</li><li>• AT (including South-Tyrol): 2</li><li>• DE: 3</li><li>• FR: 4</li><li>• IT (excluding South-Tyrol): 5</li></ul>	
Type	Integer	
Values	1'000'000 to 9'999'999	

## 4.2 Segment name (name)

Name	name	Segment name
Description	The common segment name of the route	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>• Leave empty if the segment doesn't has a name. Most segments don't have names.</li><li>• In case the segment has names in many languages, apply the most often used name.</li></ul>	
Type	String	
Values		
Example		

## 4.3 Segment name in a particular language (name\_xx)

Name	name_xx	Segment name in a particular language
Description	The common name of the segment in the language xx	
Class	Optional	
Comment	<ul style="list-style-type: none"><li>• Leave empty if the segment doesn't has a name. Most segments don't have names.</li><li>• Use the language tags according to <a href="#">ISO 639</a>.</li><li>• Don't copy the default name</li></ul>	
Type	String	
Values		
Example		

## 4.4 Skitour (ski)

Name	ski	Skitour
Description	Defines whether the segment is suitable as a <b>ski</b> tour (1) or not (0).	
Class	Optional	
Comment		
Type	Integer	
Values	0: Not suitable as a ski tour 1( <b>default</b> ): Suitable as a ski tour	

## 4.5 Snowshoe (snowshoe)

Name	snowshoe	Snowshoe
Description	Defines whether the segment is suitable as a <b>snowshoe</b> tour (1) or not (0).	
Class	Optional	
Comment		
Type	Integer	
Values	0 ( <b>default</b> ): Not suitable as a snowshoe tour 1: Suitable as a snowshoe tour	

## 4.6 Foot section (foot)

Name	foot	Foot section
Description	Defines whether the segment must be managed on foot (1) or its possible to manage it by ski/snowshoe (0). On the map such a segment will be displayed as a dotted line.	
Class	Optional	
Comment	Foot sections can be assigned manually or calculated automatically (see <a href="#">Furrer, 2024</a> )	
Type	Integer	
Values	0 ( <b>default</b> ): Not a foot section 1: A foot section	

Furrer (2024) developed an algorithm able to calculate automatically “foot section”. How should manual and automatic assignment interact (tbd)?

## 4.7 Extra caution section (caution)

Name	caution	Extra caution section
Description	Defines whether “extra caution” must be taken on the segment (1) or not (0). On the map such a segment will be displayed as a dashed line.	
Class	Optional	
Comment	“Extra Caution” sections can be assigned manually or calculated automatically (see <a href="#">Furrer, 2024</a> )	
Type	Integer	
Values	0 ( <b>default</b> ): No “extra caution” must be taken. 1: “Extra caution” must be taken.	

Furrer (2024) developed an algorithm able to calculate automatically “extra caution section”. How should manual and automatic assignment interact (tbd)?

## 4.8 DIN norm (din)

Name	din	Compliant to DIN norm
Description	Specifies, if the segment is compliant to the DIN norm xx	
Class	Optional	
Comment	Needed for DAV	
Type	Integer	
Values	0 ( <b>default</b> ): Its unknown, if the segment is compliant to the DIN norm 1: "The segment is compliant to the DIN norm"	

## 4.9 Manual Direction Type (mdir)

Name	mdir	Manual Direction
Description	Preferable travel direction, recorded manually.	
Class	Optional	
Comment	Its recommended to digitize always in the major direction. Usually that's from bottom (low elevation) to top (high elevation).	
Type	Integer	
Values	0 ( <b>default</b> ): Both directions. 1: Upwards (in the digitization direction). 2: Downwards (against the digitization direction).	

## 4.10 Target (target)

Name	target	Target
Description	The target of the segment. A target is a platform, where the segment can be published. Each platform is defined by a letter. If the letter is uppercase, the segment can be published. If the letter is lowercase, its not allowed to publish the segment.	
Class	Optional	
Comment	<p>Allowed platform tags:</p> <p>S: Skitouren guru P: Panico Books O: OutdoorActive M: Alpenvereins-Karten I: IGN France T Swisstopo X: No platform</p> <p>Special cases:</p> <ul style="list-style-type: none"><li>• If the field is empty or null, the segment can be published on all platforms <b>(default)</b></li><li>• Use the characters "X" or "x", if no platform is allowed to publish the segment.</li></ul> <p>Remarks</p> <ul style="list-style-type: none"><li>• Its not allowed to mix uppercase and lowercase letters</li></ul>	
Type	String	
Values		
Examples	<p>m: Publish on all platforms except on Alpenverein-Karten PS Publish only on Panico and Skitouren guru iP: Combination is not allowed: null: Publish on all platforms (empty string): Publish on all platforms x: Not to be published at all X: Not to be published at all</p>	

## 4.11 Comment (comment)

Name	comment	Comment
Description	Any comment in any language	
Class	Optional	
Comment		
Type	String	
Values		

## 4.12 Owner (owner)

Name	owner	Owner
Description	The owner of the segment, if its on the border: ms, ae, se, ud	
Class	Optional	
Comment		
Type	String	
Values		

The owner only must be specified, if its on the border.