NOI A.G. / S.p.A.
Roberto Cavaliere
r.cavaliere@noi.bz.it
T +39 0471 066 676

SkyAlps GTFS Export Tool

v1.1, 08.02.2023

Preliminary notes 1

SSIM format decoding specifications 2

Fare data information 2

Real-time data information 4

Additional airports metadata information 5

GTFS converter 5

agency.txt 6

stops.txt 6

routes.txt 8

trips.txt 10

stop_times.txt 11

calendar.txt 13

calendar_dates.txt 14

Preliminary notes

SkyAlps is an Italian airline operator managing flights at the Bolzano airport in South Tyrol. Thanks to the support of NOI, SkyAlps has initiated an innovation process that aims to share the data of the air services offered.

The first set of data which is shared is related to the planned timetable of the flights offered, which is made available through a machine-readable API, i.e. through the AeroCRS hub, to which SkyAlps is connected. The reference methods that will be used are the following:

- **Planned data**: https://docs.aerocrs.com/reference/getschedule
- **Fare data**: https://docs.aerocrs.com/reference/getfares

The access credentials have been made available by SkyAlps. After a more detailed evaluation of the data retrieved by the getschedule service, it has been noted that in case of changes in the schedules these are visible only through the SSIM format. Therefore, the data should be requested in this way and not in the JSON format. From a correct decoding of the SSIM format, please check this specification: https://www.slots-austria.com/jart/prj3/sca/uploads/data-uploads/downloads/e)%20Miscellanious/overall/SCR,%20SIR%20Quick%20Guide.pdf

As far as the **real-time data** is concerned, an additional API has been made available. The end-point is:

https://dataprovider.ifly.aero:8443/fidsdataproviderproxy/dataProvider.ashx

The access to the IP is filtered on the base of the source IP, and the access to the data also requires a token in the HTTP parameter. The API provides real-time and scheduled data, but since reference scheduled data are retrieved from the AeroCRS API, only real-time data is considered from this method.

SSIM format decoding specifications

A single data record is characterized by a pattern as follows:

```
3 BN 19520101J10NOV2210NOV22   4    BZO09000900+0100  DUS10451045+0100  DH4
BN 1952                           Y76                    000003
```

The main fields to be considered are:

- **flight_number**: in the example “BN 1952”
- **date_of_operation**: in the example “10NOV22” (the corresponding week day is provided as number, in this case ‘4’, since November 10th 2022 is a Thursday; the week days not considered are not set)
- **departure_airport_code**: in the example “BZO”
- **departure_time**: in the example “0900”, meaning 09:00. Please note that times are provided in UTC format
- **arrival_airport_code**: in the example “DUS”
- **departure_time**: in the example “1045”, meaning 10:45. Please note that times are provided in UTC format

The data should be retrieved from the API so that for each calendar data the single flights are returned. If a certain time interval is requested, than the API could return the different days in which a flight is scheduled, for example:

```
3 BN 19500201J16NOV2225NOV22 3 5  BZO08000800+0100 BER09500950+0100 DH4                                BN
1950               Y76           000024
```

In this case there are two flights to be considered, namely November 16th (Wednesday) and November 25th (Friday). In other word, a separate flight information should be considered for explicit week day reported (in this case associated to ‘3’ and ‘5’). In order to avoid this complex mapping, the solution is to interrogate the API for each calendar day.

Fare data information

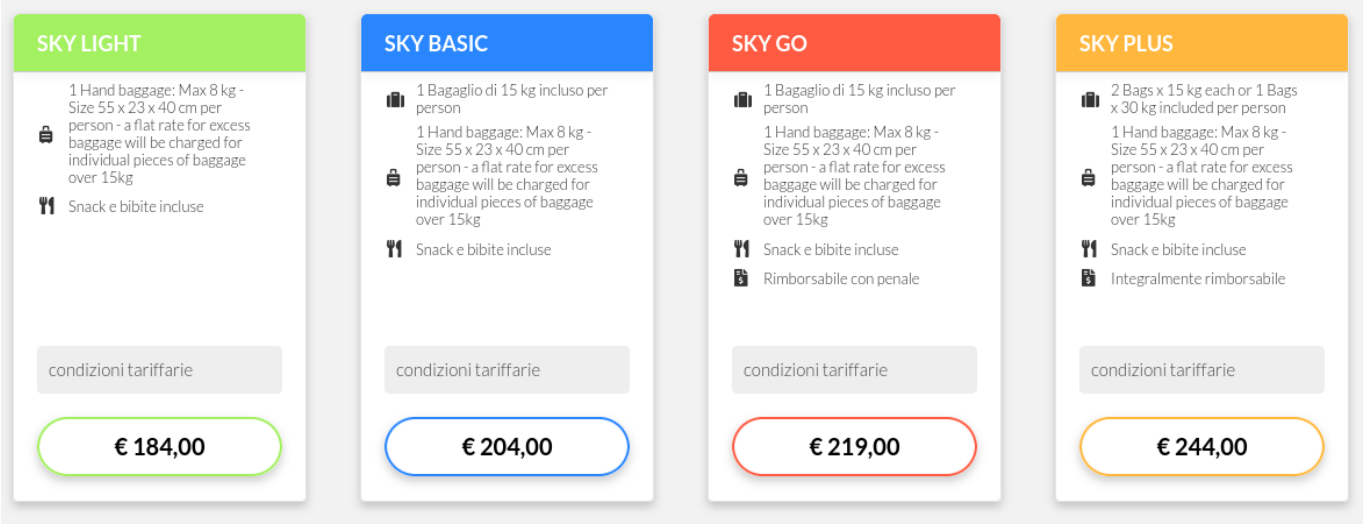
As already mentioned, the fare data can be retrieved through a separate service, documented at https://docs.aerocrs.com/reference/getfares

Basically, by giving in input a certain time interval, it is possible to get all prices associated to the routes planned in the given time interval. The prices are not associated to specific flights, but to the routes, i.e. to a combination of airport A to airport B (which could also include transfer flights!). These prices have to be intended as fare indications, so not as exact prices associated to the flights.

The list of fields provided are:

Parameter	Type	Description
count	Number	number of fares found
airlineDesignator	String	Airline Designator
airlineICAOcode	String	Airline 3 letter ICAO code
airlinename	String	Airline name
fromCode	String	Destination FROM code
toCode	String	Destination TO code
fromDate	String	Fare flight date range start
toDate	String	Fare flight date range end
classes	Array	List of class codes and the quantity available for each one
adultFareRT	Money	Fare for Adult (Round trip)
childFareRT	Money	Fare for Child (Round trip)
infantFareRT	Money	Fare for Infant (Round trip)
tax1RT	Money	Tax (Round trip)
tax2RT	Money	Tax (Round trip)
tax3RT	Money	Tax (Round trip)
tax4RT	Money	Tax (Round trip)
adultFareOW	Money	Fare for Adult (One Way)
childFareOW	Money	Fare for Child (One Way)
infantFareOW	Money	Fare for Infant (One Way)
tax1OW	Money	Tax (One Way)
tax2OW	Money	Tax (One Way)
tax3OW	Money	Tax (One Way)
tax4OW	Money	Tax (One Way)
chargeTaxOnReturnTrip	String	When searching for a RT, this will state if we charge the tax on both legs, first leg only or second leg only
notification	String	Fare notification

It is important to highlight that this set of attributes is repeated for each far package foreseen by SkyAlps, as indicated in the web-site once a flight selection is carried out. In this case the metadata JSON is properly organized so to have different sub-structure, one for each fare package.



Real-time data information

The API provides a JSON file with a list of flights divided into departures (DEP) and arrivals (ARR). Actually the API is not very self-descriptive, and the provided fields need an explanation. For the departures, following fields are provided:

- F = Flight Code
- EX = Expected Time
- SC = Scheduled Time
- D = Destinations List (multilanguage list)
- A = Airline Code
- S = Status Code (see explanation below)
- C = Checkin Code
- G = Gate Code
- GI = Gate information Time
- T = Terminal
- DC = Flag of delay. 0 = no flag. 1 = early flight, 2 = delayed flight

For the arrivals, following fields are provided:

- F = Flight Code
- EX = Expected Time
- SC = Scheduled Time
- D = Destinations List (multilanguage list)
- A = Airline Code
- S = Status Code (see explanation below)
- B = Belt Code
- T = Terminal

- Flag of delay. 0 = no flag. 1 = early flight, 2 = delayed flight

The status codes follow this convention:

- B = Boarding
- U = Last Call
- Z = Boarding Closed
- C = Check-In Opened
- D = Departed
- L = Landed
- G = Gate Number
- Y = Diverted
- X = Cancelled
- R = Baggage claim
- K = Check-In Closed

Additional airports metadata information

Additional airports metadata information could be made available through other sources. In particular, the open data available at https://ourairports.com/data/ are used. The additional airports metadata information are made available as a CSV file, and relate to all airports in the world. The file is regularly updated, so the proposed routine is to download and read this file once a day.

In order to create a matching between the airports provided by the above described mentioned web services, the field "**iata_code**" is used as key.

GTFS converter

The function requested to the GTFS is to map the available data provided through the web-services of the Open Data Hub (https://mobility.api.opendatahub.bz.it/v2/flat,node/Flight) into the GTFS format specification, which foresees data to be stored in a certain number of TXT files, as summarized in the following table.

FILENAME	**GTFS SPECIFICATION** (OPTIONAL / MANDATORY)
agency.txt	Mandatory
stops.txt	Mandatory
routes.txt	Mandatory
trips.txt	Mandatory
stop_times.txt	Mandatory
calendar.txt	Mandatory
calendar_dates.txt	Mandatory
fare_attributes.txt	Optional
fare_rules.txt	Optional
shapes.txt	Optional
frequencies.txt	Optional
transfers.txt	Optional
feed_info.txt	Optional

Table 1: GTFS specification (list of files).

So, let's consider each single mandatory file and analyze in detail all mandatory fields which need to be provided, in order to generate a valid GTFS export. Since the data provider shares only very basic information about the flights, let's just consider in this analysis only the mandatory fields.

AGENCY.TXT

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
agency_id	**Optional**	**[String]**. Is an ID identifying a transit agency. Can be optional in case of data related to a single agency.	**Not supported**
agency_name	**Mandatory**	**[String]**. Name of the transit agency.	**Supported**. Default value "**SkyAlps**"
agency_url	**Mandatory**	**[URL]**. URL of the transit agency.	**Supported**. Default value "**https://www.skyalps.com**"
agency_timezone	**Mandatory**	**[Enumerated]**. Timezone reference.	**Supported**. Default value "**Europe/Rome**"
agency_lang	**Optional**	**[Enumerated]**. Primary language reference.	**Not supported**
agency_phone	**Optional**	**[String]**. Phone contact of the transit agency.	**Not supported**
agency_fare_url	**Optional**	**[URL]**. Fare URL of the transit agency.	**Not supported**
agency_email	**Optional**	**[String]**. Mail contact of the transit agency.	**Not supported**

Table 2: GTFS specification (agency.txt).

STOPS.TXT

This file shall contain the information of the Bolzano airport and all airports for which a flight connection exists.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
stop_id	**Mandatory**	**[String].** Is an ID identifying the stop.	**Supported**. To be taken from the field **sname** of the Open Data Hub API, the **departure_airport_code** and **arrival_airport_code** must be properly parsed. All flights should be analyzed and this table should have a row for each airport connected to the airport of Bolzano.
stop_code	**Optional**	**[String]**. Contains a short name which may be useful for the passengers to recognize a stop.	**Supported**. Equivalent to stop_id

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
stop_name	**Conditionally required**[1]	**[String]**. The name of the stop, which may be useful for the passengers to recognize a stop.	**Supported**. Equivalent to stop_id
tts_stop_name	**Optional**	**String]**. "Readable" version of the stop name (for text-to-speech applications)	**Not supported**
stop_desc	**Optional**	**[String]**. Contains an additional description of the stop	**Not supported**
stop_lat	**Conditionally required**	**[Numeric]**. Latitude of the stop (WGS84)	**Supported.** Provided through the additional airports metadata CSV, field "**latitude_deg**"
stop_lon	**Conditionally required**	**[Numeric]**. Longitude of the stop (WGS84)	**Supported.** Provided through the additional airports metadata CSV, field "**longitude_deg**"
zone_id	**Conditionally required**[2]	**[String]**. Is an ID of a fare zone associated to the stop.	**Not supported**
stop_URL	**Optional**	**[URL]**. URL of a web page dedicated to the stop.	**Not supported**
location_type	**Optional**	**[Enum]**. Type of stop. • **'0' (or empty**): Stop Point • **'1'**: Station (Stop Area) • **'2'**: Entrance / Exit • **'3'**: Generic Node (to be used in combination with pathways.txt) • **'4'**: Boarding Area[3]	**Not supported**
parent_station	**Conditionally required**[4]	**[ID reference]**. Contains the ID of the parent station, with the following logic. • **location_type = '0' (or empty**): ID of parent station • **location_type = '1'**: field left empty • **location_type = '2'**: ID of parent station • **location_type = '3'**: ID of parent station • **location_type = '4'**: ID of stop point This field is:	**Not supported**

[1] Required for locations which are stops (location_type=0), stations (location_type=1) or entrances/exits (location_type=2).Optional for locations which are generic nodes (location_type=3) or boarding areas (location_type=4). The same applies also for stop_lat and stop_long.

[2] Required if "fare_rules.txt" is used.

[3] Actually, '0' refers to the concept of 'Quay' in NeTEx, while '4' to the concept of Scheduled Stop Point.

[4] Required if location_type is present

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
		• **Required** for locations which are entrances (**location_type=2**), generic nodes (**location_type=3**) or boarding areas (**location_type=4**). • **Optional** for stops/platforms (**location_type=0**). • **Forbidden** for stations (**location_type=1**).	
stop_timezone	**Optional**	**[Timezone]**. Timezone of the stop.	**Not supported** (since the timezone defined in agency.txt is sufficient)
wheelchair_boarding	**Optional**	**[Enum]**. Indication if wheelchair boarding is allowed. Possible values: For parentless stops: • **'0' (or empty**): no info • **'1'**: some vehicles supported • **'2'**: wheelchair boarding not possible For child stops: • **'0' (or empty**): as the parent station • **'1'**: some accessible path from outside the station to the specific stop point / platform • **'2'**: some accessible path from outside the station to the specific stop point / platform For station entrances / exits: • **'0' (or empty**): as the parent station • **'1'**: entrance is wheelchair accessible • **'2'**: no accessible path	**Not supported**
level_id	**Optional**	**[ID reference]**. Contains the ID of the level (levels.level_id, see levels.txt)	**Not supported**
platform_code	**Optional**	**[Text]**. Platform identifier.	**Not supported**

Table 3: GTFS specification (stops.txt).

ROUTES.TXT

This file shall contain the information of the connections linked to the Bolzano airport and all airports for which a flight connection exists.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
route_id	**Mandatory**	**[String].** Is an ID identifying the airport connected to the airport of Bolzano	**Supported**. It is the code of the departure / arrival airports, as defined for stop_id. There should be one record for each airport connected in one the directions (departures / arrivals).

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
agency_id	**Conditionally required**[5]	**[ID reference]**. Is the reference agency ID set in agency.txt	**Not supported**.
route_short_name	**Conditionally required**[6]	**[String]**. Short name of the "route".	**Supported**. Takes the same value as route_id.
route_long_name	**Conditionally required**	**[String]**. Long name of the "route".	**Not supported**.
route_desc	**Optional**	**[String]**. Description of the "route".	**Not supported**.
route_type	**Mandatory**	**[Enum]**. Type of transportation used on the "route". Default values[7]: • "0": Tram, Streetcar, Light rail • "1": Subway, Metro • "2": Rail • "3": Bus • "4": Ferry • "5": Cable tram • "6": Aerial lift, suspended cable car • "7": Funicular • "11": Trolleybus • "12": Monorail	**Supported**. Default value '**1100**'
route_URL	**Optional**	**[URL]**. URL of a specific web page about a "route"	**Not supported.**
route_colour	**Optional**	**[Color]**. Color of the "route", according to communication means to travelers	**Not supported.**
route_text_color	**Optional**	**[Color]**. Color of texts associated to the "route", according to communication means to travelers	**Not supported.**
route_sort_order	**Optional**	**[Integer]**. For sorting the "routes" in a certain order, for passenger information purposes	**Not supported.**
continuous_pickup	**Optional**	**[Enum]**. In case the geoinformation of a "route" is given (see shapes.txt), indicates the	**Not supported.**

[5] Mandatory only if agency.txt contains more than one record (i.e. multiple data providers).

[6] One of the two fields "route_short_name" and "route_long_name" must be specified.

[7] Extended value list available at: https://developers.google.com/transit/gtfs/reference/extended-route-typeshttps://developers.google.com/transit/gtfs/reference/extended-route-types

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
		possibility to use any points as pick-up point. Possible values: • "0": Continuous stopping pick-up • "1" (or empty): No continuous stopping pick-up • "2": Must phone agency to arrange continuous stopping pick-up • "3": Must coordinate with driver to arrange continuous stopping pick-up	
continuous_drop_off	**Optional**	**[Enum]**. In case the geoinformation of a "route" is given (see shapes.txt), indicates the possibility to use any points as drop-off point. Possible values: • "0": Continuous stopping drop-off • "1" (or empty): No continuous stopping drop-off • "2": Must phone agency to arrange continuous stopping drop off • "3": Must coordinate with driver to arrange continuous stopping drop off	**Not supported.**

Table 4: GTFS specification (routes.txt).

TRIPS.TXT

This file shall contain the information of all the flights departing and arriving to Bolzano.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
route_id	**Mandatory**	**[String]**. Is the ID of the defined in routes.txt	**Supported.**
service_id	**Mandatory**	**[String]**. Is the ID of the set of dates in which the "trip" (journey) takes place. Is a reference to the ID provided in calendar.txt or calendar_dates.txt	**Supported.**
trip_id	**Mandatory**	**[String]**. Is the ID of the "trip"	**Supported**. To be taken from the field **scode** of the Open Data Hub API
trip_headsign	**Optional**	**[String]**. Destination text. Can be overridden by the field stop_headsign in stop_times.txt, if set	**Not supported.**

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
trip_short_name	**Optional**	**[String]**. Short name of the "trip"	**Not supported.**
direction_id	**Optional**	**[Enum]**. Direction of the "trip" • "0": outbound • "1": inbound	**Supported**. To be set as '0' for flights departing from Bolzano, and '1' for flights arriving to Bolzano. Could be easily check by considering the order of the airports codes in the field sname of the Open Data Hub API
block_id	**Optional**	**[String]**. Is the ID of the block which the "trip" belongs	**Not supported.**
shape_id	**Conditionally required**[8]	**[String]**. Is the ID of the geospatial shape associated to the "trip"	**Not supported.**
wheelchair_accessible	**Optional**	**[Enum]**. Indicates if the "trip" is accessible to wheelchair users • "0" (or "empty"): no accessibility information available • "1": at least one rider in a wheelchair allowed • "2": no riders in wheelchairs can be accommodated	**Not supported.**
bikes_allowed	**Optional**	**[Enum].** Indicates if the "trip" allows the transportation of bikes: • "0" (or "empty"): no bike transport information available • "1": at least one rider with bike allowed • "2": no riders with bike can be accomodated	**Not supported.**

Table 5: GTFS specification (trips.txt).

STOP_TIMES.TXT

This file typically contains the departure and arrival times at all stops of the different scheduled trips. In this particular implementation, each trip shall be characterized by two records in this table: one record related to the departure time at the departing airport and one record related to the arrival time at the arriving airport.

[8] Mandatory only if the trip has a continuous pickup or drop-off behavior defined either in routes.txt or in stop_times.txt.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
trip_id	**Mandatory**	**[String]**. Is the ID of the "trip" (journey) defined in trips.txt.	**Supported**. Link to the trip.
arrival_time	**Conditionally required**[9]	**[Time]**. Is the arrival time of the "trip" in correspondence of the stop identified by stop_id.	**Supported**. To be taken from the field **smetadata -> sta** of the Open Data Hub API (in case the record is related to the arrival airport) or equal to departure_time (in case the record is related to the starting airport)
departure_time	**Conditionally required**[10]	**[Time].** Is the departure time of the "trip" in correspondence of the stop identified by stop_id	**Supported**. To be taken from the field **smetadata -> std** of the Open Data Hub API (in case the record is related to the starting airport) or equal to arrival_time (in case the record is related to the arrival airport)
stop_id	**Mandatory**	**[String]**. Is the ID of the stop defined in stops.txt.	**Supported**.
stop_sequence	**Mandatory**	**[Non negative integer].** Provides the ordering of the stops. Values must increase along the "trip" but do not need to be consecutive.	**Supported.** To be set as follows: **1 = departing airport; 2 = arrival airport**.
stop_headsign	**Optional**	**[Text]** Text that appears on signage identifying the trip's destination to passengers. This field overrides the default value trip_headsign set in trips.txt, to be used when the headsign changes between stops.	**Not supported**
pickup_type	**Optional**	**[Enum]** Indicates pick-up type. Possible values: • "0" (or empty): Regularly scheduled pickup. • "1": no pickup available. • "2": must phone agency to arrange pickup. • "3": must coordinate with driver to arrange pickup.	**Not supported**
dropoff_type	**Optional**	**[Enum]** Indicates dropoff type. Set as pickup_type.	**Not supported**
continuous_pickup	**Optional**	**[Enum]** Indicates that the rider can board the transit vehicle at any	**Not supported.**

[9] Required in case timepoint = 1 or not set.

[10] As for arrival_time.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
		point along the vehicle's travel path as described by shapes.txt, from this stop_time to the next stop_time in the trip's stop_sequence. Possible values: • "0": continuous stopping pickup. • "1" (or empty): continuous stopping pickup. • "2": must phone agency to arrange continuous stopping pickup. • "3": must coordinate with driver to arrange continuous stopping pickup.	
continuous_ dropoff	**Optional**	**[Enum]** Equivalent to continuous_pickup	**Not supported.**
shape_dist_traveled	**Optional**	**[Non negative float]** Actual distance travelled along the associated shape, from the first stop to the stop specified in this record	**Not supported.**
timepoint	**Optional**	**[Enum]** Indicates if arrival and departure times for a stop are strictly adhered to by the vehicle or if they are instead approximate. Possible values: • "0": times are considered approximate. • "1" (or empty): times are considered exact.	**Not supported.**

Table 6: GTFS specification (stop_times.txt).

CALENDAR.TXT

The Open Data Hub retrieves and provides all flights scheduled for the next 180 calendar days. For simplicity sake, there is a single service_id for each calendar day, to which all flights taking place on that place must be referred to. This means, that the table should contain 180 records, one for each calendar day.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
service_id	**Mandatory**	**[String]**. Identifies a set of dates when service is available for one or more routes. Each service_id value must be unique in a calendar.txt file.	**Supported**. To be taken from the field **scode** of the Open Data Hub API (just the part of the string after "_", i.e. the calendar day)

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
monday	**Mandatory**	**[Enum]** Indicates whether the service operates on all Mondays in the data range defined by start_date and end_date. Exceptions are indicated in calendar_dates.txt. Possible values: • "0": service not available • "1": service available	**Supported**. Easily set up based on the calendar day associated to the day associated to this specific service_id.
tuesday	**Mandatory**	**[Enum]** As monday	**Supported**. As monday
wednesday	**Mandatory**	**[Enum]** As monday	**Supported**. As monday
thursday	**Mandatory**	**[Enum]** As monday	**Supported**. As monday
friday	**Mandatory**	**[Enum]** As monday	**Supported**. As monday
saturday	**Mandatory**	**[Enum]** As monday	**Supported**. As monday
start_date	**Mandatory**	**[Date].** Indicates the start service day for the service interval. Expressed in the YYYYMMDD format	**Supported**. To be parsed as a function to the calendar day associated to this service_id.
end_date	**Mandatory**	**[Date].** Indicates the end service day for the service interval. The end date given is the last service day. Expressed in the YYYYMMDD format	**Supported**. As start_date

Table 7: GTFS specification (calendar.txt).

CALENDAR_DATES.TXT

Because of the choices related to calendar.txt, there won't be any exceptions to be highlighted there. Therefore to be formally correct the table should be available, but left empty.

Field Name	GTFS specification (optional / mandatory)	GTFS specification (description)	SouthTyrol specification
service_id	**Mandatory**	**[String].** Reference to the service_id given in calendar.txt.	**Supported**. To be left empty.
date	**Mandatory**	**[Date]** Date when service exception occurs.	**Supported**. To be left empty.
exception_type	**Mandatory**	**[Enum]** Indicates whether the service is available or not on the given date. This entry overrides the information specified in calendar.txt. Possible values: • "1": service added for the specified date; • "2": service removed for the specified date	**Supported**. To be left empty.

Table 8: GTFS specification (calendar_dates.txt).