

GitHub

This repository Search

Explore Features Enterprise Pricing

Sign up

Sign in

google / transitfeed

Watch 36

★ Star 109

Fork 38

## FeedValidator

bdferris edited this page on Oct 8 2014 · 4 revisions

The `feedvalidator` is a command line tool that checks a GTFS feed for problems and generates a HTML report. Running it on your transit data feed and fixing the issues that it finds can save you from display and routing problems down the road.

For a list of the errors and warnings that FeedValidator outputs, see the [FeedValidatorErrorsAndWarnings](#) page.

You may run the validator on an uncompressed directory of GTFS txt files or a ZIP file directly.

## Windows standalone version

We provide a Windows executable version for convenience; if you're running Windows, you should use this one. To get it, download the latest release from:

<https://github.com/google/transitfeed/releases/latest>

Once you have it downloaded and unzipped, there are a few ways to run it:

### Easiest

Drag a GTFS zip file or directory onto `feedvalidator.exe`. A window will pop up with the result of the validation test:



### Intermediate

Double-click `feedvalidator.exe`. A window will pop up and ask you to enter the location of your feed file or directory. You can type it in, or just drag a GTFS file or directory onto the window and hit Enter.

&lt;&gt;

①

🔗

📄

⚡

📊

▼ Pages 14

Home

BuildingPythonWindowsExecutables

FeedValidator

FeedValidatorErrorsAndWarnings

GoogleRandomQueries

KMLWriter

LatestReleaseVersion

Merge

Notes

ReleaseProcess

Revision History

ScheduleViewer

TransitFeed

UnusualTripFilter

Clone this wiki locally

<https://github.com/google/transitfeed/wiki>

Clone In Desktop

*C:\CD\trans\GTFS\free to feed.ch2*

## Expert

Go to the Windows command prompt, and type: `feedvalidator <name of feed file or directory>` . If you want to avoid the prompt at the end of the validation, use the `--noprompt` parameter.

## Python source code version

Use this version if you're on Mac OS X or Linux.

Before you run the feed validator you must install Python. It is tested with versions 2.4 and 2.5 and should work with 2.6. You can download Python from <http://www.python.org/download/>

See Home for details on downloading the source distribution.

## Running

Run the feed validator as

```
feedvalidator.py <feed filename>
```

The warning "Timezone not checked (install pytz package for timezone validation)" is normal. `feedvalidator` can not check the spelling of the `agency_timezone` until you install the `pytz` package.

## Extra help for Windows XP/2000 users

Open a new Command Prompt window. Change into the directory containing `feedvalidator.py` (Hint: type "`cd`" without quotes and drag the icon of the folder into command prompt window). Run `feedvalidator.py` . If you don't want to type the feed filename try dragging it into the command prompt window.

## Command line options

Usage: `feedvalidator.py [options] [<input GTFS.zip>]`

Validates GTFS file (or directory) `<input GTFS.zip>` and writes a HTML report of the results to `validation-results.html`.

If `<input GTFS.zip>` is omitted the filename is read from the console. Dragging a file into the console may enter the filename.

For more information see <https://github.com/google/transitfeed/wiki/FeedValidator>

Options:

<code>--version</code>	show program's version number and exit
<code>-h, --help</code>	show this help message and exit
<code>-n, --noprompt</code>	do not prompt for feed location or load output in browser
<code>-o FILE, --output=FILE</code>	

write html output to FILE or --output=CONSOLE to print all errors and warnings to the command console

-p, --performance output memory and time performance (Availability: Unix)

-m, --memory\_db Use in-memory sqlite db instead of a temporary file. It is faster but uses more RAM.

-d, --duplicate\_trip\_check Check for duplicate trips which go through the same stops with same service and start times

-l LIMIT\_PER\_TYPE, --limit\_per\_type=LIMIT\_PER\_TYPE Maximum number of errors and warnings to keep of each type

--latest\_version=LATEST\_VERSION a version number such as 1.2.1 or None to get the latest version from code.google.com. Output a warning if transitfeed.py is older than this version.

--service\_gap\_interval=SERVICE\_GAP\_INTERVAL the number of consecutive days to search for with no scheduled service. For each interval with no service having this number of days or more a warning will be issued

## Revision History

See the [TransitFeedDistribution](#) page.

