GP4 GPaediaMaker Guide

for version 1.75, march 2023

Introduction

The tool GPaediaMaker will help you create gpaedia and menu files with up-to-date season data for a GP4 mod. Input your data (driver, team, track) comfortably into three data sheets of your favourite calculation program. Use format CSV files (driverformat, teamformat, trackformat) either as standard or as user format, that can create non-standard gpaedia.
Using the input data from the tables together with constant gps and str text files and some language specific files the program builds gps files and str files according to your settings in the INI file.
There is no GUI, just an information output on the console and detailed output to a log file. CSV data input must be in english language using Unicode UTF16-LE characters. Automatic creation of other language files (Deutsch, español, Français, italiano) and generation of trackspecific gpaedia for a CSM mod are supported.

This guide will help you understand all features of the program. If you just want the basic information, read chapters Setting up GPaediaMaker and Quick start.

Free format

Since release 1.00 GPaediaMaker comes with the new feature **free format** and three new format CSV files in the data folder. You can now create gps files as you like them and are not limited to the standard format of the original GP4 gpaedia. You also have access to most of the translations.
See chapters Standard format CSV files and User format for a detailed explanation.

A new created data CSV file must have the same row structure and the same first column as the corresponding format CSV file. Old CSV data files used by earlier versions than 1.00 are not compatible with the new program. It is therefore recommended to use a separate directory structure for the new versions files. The program exe file **txt2gps_ff.exe** has got an "ff‘ ending distinguishing it from the old "txt2gps.exe“.

Old files can be converted to the new format with the tool convertcsv that is also useful for creating empty CSV data tables in standard format for a new season.

GPaediaMaker tutorial

In the tutorial How to update gpaedia and menu files with a new seasons data you will find additional information.

Content

Setting up GPaediaMaker

The GPaediaMaker package includes the following files, folders and archives:

- txt2gps_ff.exe
- gpaedia.ini
- data subfolder
- examples.zip
- convertcsv.zip
- GPaediaMaker.exe
- GPaediaMaker.cfg
- records_export subfolder

Copy the **exe** and the **ini** files into the program directory of your choice.
No installation is needed. The program should run under every windows system.
The **GPaediaMaker.exe** program is for optional use. See chapter front-end for details.

Copy the **data** folder into a directory of your choice, e.g. the program path.
This is the directory for not-season-dependent data, which is obligatory to run the program.
It contains

- 3 standard format CSV files "driverformat.csv", "teamformat.csv", "trackformat.csv" (UTF-16)
- 55 gpaedia gps text files (11 parts of the gps file for each of the 5 languages, UTF-16),
- 20 menu str text files (4 parts of the str file for each of the 5 languages, ANSI)
- 1 CSV table with characters and their width (UTF-16)
- 3 language specific files:
  - nationality.csv with the country names and national adjectives (UTF-16)
  - grand_prix.csv with national adjectives of the countries (UTF-16)
  - special.txt (+ alternative special_a.txt) with some extra information for translation (UTF-16)

Of course you can edit those files too and with the last three ones this may be occasionally necessary. Please keep in mind that the UTF-16 text files must end with a CRLF, otherwise txt2gps_ff will reject them with an errormessage.

examples.zip (optional) contains two examples for GP seasons 2001 and 1991.

- 2001 is the original season using standard format.
  Here you must pay attention to the fact that corner data speed is given in mph (mile per hours).
  The file "driver.csv" has got data to create trackspecific gps files to be used in a 2001 mod.
- 1991 is an example with a non-standard format. It comes with his own format files and you have to set the ini file parameter ***user_format*** (to 1) to tell the program to use them.
  It also uses trackspecific gps files, that can be created automatically by GPaediaMaker.
  See chapter Season 1991 example for more details.

Additional information is given in the examples readme files.

Create a directory for any example season and extract the container there, if you want to explore those files. The directory structure may look like this:

programm directory	*programpath*
data directory	*programpath*\data
2001 directory	*programpath*\GP2001 (optional)
1991 directory	*programpath*\GP1991 (optional)

convertcsv.zip (optonal) contains convertcsv.exe and the ini file convert.ini.
With this tool you can convert your CSV files created with versions older than 1.00 to the new free format or create new empty free format files. For a detailed description see chapter Convertcsv.

Copy the folder **records_export** (optional) into a directory of your choice, e.g. the program path.
Read the chapter Changing track records and the readme text file inside this folder for more information.

Before running the program to process your new generated gpaedia CSV files you have to adjust the ini file (e.g. "gpaedia.ini" or "gpaedia2013.ini") according to your directory settings and your intentions.

Editing the INI file

The INI file is the control center of GPaediaMaker. You define paths for in- and output and control gpaedia generation with various specifications, some of them mandatory, most of them optional.
The following example contains all possible specifications with some comments. In reality you may use a much smaller file omitting all lines that you don't need.

```
;--- gpaedia.ini ---
;--- GP4 GPaediaMaker configuration file ---
; language variables (0 to 4):
; %ln% ==> en, de, es, fr, it
; %language% ==> English, Deutsch, español, Français, italiano

; gps: language files to create
;     create: yes=1, no=0 (optional, default is 1,0,0,0,0)
gps = 1,1,1,1,1

; define season variable %season% (default: 2001)
season=2016

; last_year (default: %Season% - 1)
;last_year = 2001

; define variable %program_path% (default: path of exe file)
; program_path = d:\gpaedia\txt2gps_ff

; input path for constant txt files and standard CSV format files (needed)
in_path = %program_path%\data\

; input path for CSV (data and userformat) files and txt files (needed)
; if 2 comma-separated folders are specified, the program searches in the first
folder, then - if not found - in the second
; var_path = %program_path%\GP%season%\%season%extra, %program_path%\GP%season%\
var_path = %program_path%\GP%season%\

; output path for gps files and str files (needed; ev. created as subfolder)
out_path = %program_path%\GP%season%\

; gps file name (normal); if not specified, no gps file will be created.
ln_gps = GP%season%_%language%.gps

; gps file name trackspecific; if not specified, no trackspecific gps is created.
track_gps = GP%season%_%language%-%tracknum%.gps
;track_gps = GP%season%_%language%-%trackname%.gps
;forced_tracks = all

; trackinfo filename (normal is "%ln%_%trackname%_%season%.txt")
; if not specified, trackinfo files are not included
trackinfo_file = %ln%_%trackname%_%season%.txt

; alternative in_path special text file (default: "special.txt")
special = special_a.txt

; Process cornerdata 1=yes, 0=no (default: 1)
;corner_data = 0

; define corner data delimiter (default: Linefeed, allowed: /\,;)
; may be omitted because it is automatically detected since version 1.40
;corner_delimiter = /
```

```
; Read (corner data) speed as mph value (default: 0, read as km/h value)
;speed_in_mph = 1

; English driver-/team- or track-link names as capitals (default: 0, unchanged)
;driverlink_caps = 1
;teamlink_caps = 1
;tracklink_caps = 1

; tracklink_name values: 0,1,2,4 (default 0); controls the track link names
; also replaces the country and nat. adjective with the track name in str files
; 0 or 2: row "country" is taken; 1: row "circuit" (the track name) is taken
; 4: row 4 (after "circuit info name" is taken (to freely define the link name)
;tracklink_name = 1

; replace character for the multiple entries delimiter "|" (default: " | ")
; write "_" for a space
;tag = _-_

; force checking for fastestlap comma delimiters search for 3 (default: 2)
chk_fastestlap = 3

; Use user defined CSV format files (default = 0, standard files are used)
;user_format = 1

; Define conditional row selection values: 1,2 or 3 - where 3 means 1 or 2
; (default = 0)
; valid_rows=1 ==> valid_driverrows=1 AND valid_teamrows=1 AND valid_trackrows=1
;valid_rows = 1
;valid_driverrows = 1
;valid_teamrows = 1
;valid_trackrows = 2

; list extended characters codes (> 255) used by a CSV or txt file in the log file
; (CSV: file name and row plus column of data cell); default: 0
;log_extended = 1

; Generate str menu files (0,1,2 with default 0)
make_str = 1
; 2: str menu files CSM/TSM compatible
;make_str = 2

; Switch team/drivers or tracks
;sw_driver 9=10
;sw_driver 10=9

;sw_team 11=12
;sw_driver 21=23
;sw_driver 22=24
;sw_track 16=19

; tracklist (as alternative for sw_track specifications)
;tracklist = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,18,19

; Track 1 of driver tables replace lines (default = 1)
; only used for a mod supporting more than 17 tracks using two tracklists
;track_1=18

; name of log file (default = gpaedia; extension is always .log)
;log_file = gpaedia-2
```

The commentaries after the semicolons are explaining the the usage of this file. A detailed explanation of all parameters is given in chapter Specifications.

The INI file uses a couple of variables that are replaced by their actual values during processing:

%ln%	values: en, de, es, fr, it
%language%	values: English, Deutsch, español, Français, italiano
%season%	value: input-data behind label ***season*** = , e.g. 2013
%program_path%	value: path of program directory, e.g. d:\gpaedia\txt2gps_ff
%tracknum%	value: the number of the actual processed track
%trackname%	value: the name of the track, as given in your track table row "Circuit name".
%last_year%	value: input-data behind label ***last_year*** = , e.g. 1991; is normally only used, if not previous to ***season***; the variable is not used in the INI file itself, but by the program txt2gps_ff.exe

Quick Start

Necessary

If you just want to create GPaedia gps files with language support and do not care about the more sophisticated aspects like trackspecific data or user format files, the following example shows what to do:

Set the **gps** value:
```
gps = 1,1,1,1,1
```
Set the **season** value:
```
season = 2013
```

Set the values for the three **path**s; you can use the the variables %season% and %program_path%, which contains the path of the exe file as default:

```
in_path = %program_path%\data\            the directory with constant data files
var_path = %program_path%\GP%season%\     the directory with yout CSV files
out_path = %program_path%\GP%season%\out  the directory for created gps and str files
```

Set the value for **GPaedia language** dependent files:
```
ln_gps = %season%_%language%.gps
```

Deactivate trackspecific gps, format CSV files and menu str by adding leading semicolons:
```
;track_gps = %season%_%language%-%tracknum%.gps
;user_format = 1
;make_str = 1
```
Your data CSV files should have the format given by the standard format CSV files in folder data.

You are now ready to **Run** txt2gps_ff.
Execute txt2gps_ff.exe with your INI file as parameter, e.g. with "open with" or "drag and drop",
or start the GPaediaMaker.exe program as filebrowser, then find and doubleclick the INI file to process.
This will create the following five files in your ***out_path*** folder:

```
2013_English.gps, 2013_Deutsch.gps, 2013_español.gps, 2013_Français.gps,
2013_italiano.gps.
```

Optional Addings

Adding 1 – Trackinfo files:
If you have got trackinfo files (circuit descriptions not translated by the program), you can include them into your tracks gpaedia. To use names beginning with the 2-character-language attribut (en, de, es, fr, it) followed by the trackname and the season, such as "en_melbourne_2013.txt", you will specify:
```
trackinfo_file = %ln%_%trackname%_%season%.txt
```

To be sure that trackinfo files have been processed correctly, take a look into the "gpaedia.log" file.
For every trackinfo file there should be an entry like
```
Text file "d:\gpaedia\txt2gps_ff\GP2008\en_barcelona_2008.txt" has been processed.
```

If something went wrong you may find warning-messages like
```
>>> WARNING: file not found "d:\gpaedia\txt2gps_ff\GP2013\en_melbourne_2013.txt"
```
or
```
>>> WARNING: "d:\gpaedia\txt2gps_ff\GP2013\en_monza_2013.txt" has not UTF-16 file
format
```

See also chapter trackinfo files.

Adding 2 – Corner Data:
Corner data will be included into the gpaedia circuit information by default.
If you don't want that, you can specify
```
corner_data = 0
```
Normally corner data speed is read as km/h. If you want mph instead – as in the 2001 example – replace km/h with mph in your track.csv and specify
```
speed_in_mph = 1.
```

See chapter Corner data in the Data input part for more information.

Adding 3 – STR menu files:
As default str menu files will not be generated.
If you want do do that, you should specify

```
make_str = 1
```

or specify to get CSM/TSM compatibility:

```
make_str = 2
```

Both will create the following five str files in your ***out_path*** folder:
```
english.str, deutsch.str, español.str, français.str, italiano.str.
```

The menu files data are most often used by displays of race results. You can find additional information in the chapter STR Menu files.

Adding 4 – Switch settings:
Switch values can be set for drivers, teams or tracks. They exchange a driver, team or track by another entry of the same table.
That way you can fill your CSV files with data corresponding to reality and control with INI files, what kind of gpaedia you like to generate from them.
Look at switch specifications for detailed information about this.

Specifications

Inside the INI file you can always de/re-activate a value by adding/deleting a leading semicolon.
Some of the specifications are obligatory, some are optional (have a default):

gps This must be a list of 5 comma separated 0 or 1 as indicators for languages en,de,es,fr,it
Example: 1,1,0,0,0 to create only english and german files.
If not active, the default will create only english gps resp. str.
Only the first character of any list item is taken, e.g. 11 will be 1.

season Year of the new season; this value sets the variable %season%, wich can be used in some other specs also; it is optional (default = 2001) but should be active;
normally the "2001" in menu "race options" is replaced by this value; you will get further information in chapter User format part text files.

program_path program path; sets the variable %program_path% to be used in following specs; default is the path of the exe file. The %season% variable can be used here.
Actually this path is not limited to the program path of the exe file, so you may give it any value you like which can effectively shorten the following path definitions.

Working path specifications are necessary. You can use %season% and %program_path% here:

in_path The path from which the data files and standard format CSV files are read;

var_path The path(s) from which the new created CSV files, the user format CSV files, the user format part text files and the trackinfo textfiles are read.
This can also be a (comma separated) list of two paths. Then a file is searched for in the first part of the ***var_path*** list and – if not found – then in the second part.
This can be useful, if you want to create different gpaedia versions of one season using some part of the source files for both.

out_path The directory path for the created gps and str files. If not found it will be created.
Since version 1.52 subfolders of all levels are supported.

Since version 1.00 you don't have to care for the backslash at the end of a folders definition anymore. It is added by the program if needed.

*If one of the specs **ln_gps**, **track_gps**, **test_gps** or **trackinfo_file** ist not active, no gps files of that type will be created. With them you can can use the variables %ln%, %language% and %season% , inside **track_gps** and **trackinfo_file** %trackname% and %tracknum% also.*

ln_gps File name format of normal gps files
Example: `GP%season%_%language%.gps with season = 2011` creates gps files `GP2011_English.gps, GP2011_Deutsch.gps, GP2001_ español.gps` etc.

track_gps File name format of trackspecific gps files
Examples for `season = 2011`:
1): `track_gps = GP%season%_%language%-%tracknum%.gps` will create files like `GP2011_English-7.gps, GP2011_English-9.gps, ...`
2): `track_gps = GP%season%_%language%-%trackname%.gps` will create files like `GP2011_English-montreal.gps, GP2011_English-silverstone.gps, ...`

forced_tracks Specifies "all" or a list of positive or negative (track-) numbers.
If ***forced_tracks*** is specified as a list of numbers, then for every track(-number) included in this list, independent of the replace data in the drivertable, one of the following is forced:

- if the number is positive, this trackspecific gps file will be created
- if the number is negative, this trackspecific gps file will not be created

Only numbers, a minus or a comma are valid (spaces only at the beginning or the end).
Example: `forced_tracks = 13,15`
Wrong: `forced_tracks = 3, 9, 11`

If ***forced_tracks*** is specified as "`all`", then all trackspecific gps files will be created, even those that have successors with identical content which otherwise would not be done.
This case is useful when ***track_gps*** identifies tracks using the `%trackname%` variable instead of `%tracknum%`.

See also "driver replace" of chapter "Data input" and "GPaedia and CSM mod" for more information about trackspecific gpaedia.

test_gps File name format of a test gps file
Example: `GP%season%_test_%language%.gps` with `season = 2011` creates gps files `GP2011_test_English.gps, GP2011_test_Deutsch.gps` etc.
This was once needed to create gpaedia for a test track when only 17 track entries were possible. Since that has changed with the introduction of the switch specifications, you can use a track column (ev. greater than 17) instead and include it with sw_track.

trackinfo_file File name format of trackinfo files
Example: `%ln%_%trackname%_%season%.txt` creates files like
`en_melbourne_2011.txt, en_sepang_2011.txt, …`
`de_melbourne_2011.txt, de_sepang_2011.txt` etc.
with %trackname% taken from the "Circuit name“ entry of your track table.
See chapter Trackinfo files for further information.

special This parameter allows you to specify an alternative special text file in path ***in_path.***
The value must consist of the alternative files name without the path, e.g.
`special = special_a.txt`
The main reason for this is, that entries in the [country] part of the special text file may have to be different depending on the order of the tracks of a season.
See also chapter Language supporting files for further information.

corner_data This value decides, if corner data are created (default 1) or not (0).
You may want to omit corner data when using trackmaps without corner info.

speed_in_mph This value decides, if corner data values are read as km/h (default = 0) or as mph as necessary with the 2001 example (then set this value to 1).
It replaces the former ***corner_as_miles*** , which remains also valid.
This specification just adds the unit symbol behind the speed value but has no effect on it.

driverlink_caps Values 0 or 1 - default 0. These values decide, if in english language gpaedia the
teamlink_caps driver, team resp. track selection entries are displayed capitalized (1) or as given in the
tracklink_caps CSV file (0) which is normally lower case. This driver, team resp. track name is also displayed as title of the corresponding gpaedia.

tracklink_name Values 0 (default), 1, 2 or 4.
If set to 1, the track name is taken instead of the country (as with 2 or default 0).
If set to 4, tracktab row 4 - below "circuit info name" - will be read.
The name is shown in the gpaedia track selection and also as the track gpaedia title. It affects the str files too: the national adjective is searched by using the *tracklink_name* instead of the country. Example for 1: creating gpaedia for an US racing series where the tracks are distinguished only by the track name while the country is mostly the same.

tag With this parameter you can vary the output of delimiter "|" in multiple entries.
Default is " | “. If you are not satisfied with the look of this in the gpaedia output, you may define a replacement, specifying
tag = `_-_` *(where character "_“ will be replaced by a space)*.
Example: 1988mod double team "ags osella“ Team Principal entry:
Input: `Henri Julien|Enzo Osella`
Output: `Henri Julien - Enzo Osella`

chk_fastestlap With this parameter you can force the checking routine for the ***_fastestlap*** function entry to search for 3 comma delimiters instead of 2 to 3 (default). As ***_fastestlap*** parts (driver, team, record, date) can be either 4 or 3 (if date is omitted) it is normally only checked for a minimum of 2 delimiters (= 3 parts.)

chk_teamname This parameter controls the checking for a teamname match of drivertab and teamtab entries. Possible values are 0 or 1, default = 0 (no check).

user_format With this value set (to 1) the format CSV files "driverformat.csv", "teamformat.csv" and "trackformat.csv" as well as the part text files are taken from the ***var_path*** directory. If a user format file is not found in ***var_path***, it is searched for in ***in_path***. As default `(user_fomat = 0)` the standard format CSV files and part text files are taken from the ***in_path*** directory.

valid_driverrows
valid_teamrows
valid_trackrows
valid_rows

With these specifications CSV file rows can be processed conditionally.
Valid values are 1, 2 or 3; 3 means 1 or 2; default is 0 (row not processed at all).
They are only working in userformat mode.
`valid_rows = 1` means:
`valid_driverrows = 1` AND `valid_teamrows = 1` AND `valid_trackrows = 1`
The latter specification overrides the former.

log_extended If set to 1, extended characters codes (> 255) are listed in the log file with CSV file name and row, column of the cell; default: 0 (not listed); example message (char "œ" found):

```
Extended char 0153 found in "d:\gpaedia\txt2gps_ff\GP2001\track.csv" cell 4, 1
```

make_str This value decides, if menu str files are generated. Valid are 0,1 or 2.
Default is 0 (no menu files); 1 creates normal, 2 CSM/TSM compatible menu files.

*The **switch specifications** are available since version 1.02.*

sw_team After this value you can define a team switch like

```
sw_team 11=12.
```

This will take team 12 of your team.csv as team11.
If ***sw_team*** is specified and the corresponding ***sw_driver*** specs are not found, the program gives you a warning message.

sw_driver To take the drivers 23 and 24 as team11 drivers (21 and 22) put in:

```
sw_driver 21=23
sw_driver 22=24
```

Switch the gpaedia of drivers 9 and 10 (useful to handle the driver 9/10 bug)

```
sw_driver 9=10
sw_driver 10=9
```

Team and driver switch specifications effectively support team/driver-selection of a CSM mod.
Your team.csv and driver.csv can contain the complete data needed (more than 11 resp. 22 columns) and you control the output with gpaedia INI files containing the fitting switches.
It is possible to specify switches together with driver replace. You can even add additional driver replace rows for tracks 18 and on to be used by switched tracks.

sw_track In the gpaedia.ini 2013 example the track switches will take the track.csv columns 18 as track 14 and 19 as track 16:

```
sw_track 14=18
sw_track 16=19
```

It is also possible to assign a value to a range of tracks, e.g.

```
sw_track 4 to 17 = 20
```

or to assign an addition to a range of tracks using a leading "+“ character for the value behind the "=“ (only available together with the "to“), e.g.

```
sw_track 8 to 14 = +1
sw_track 15 to 17 = +2
```

to take track columns 9 to 15 instead of 8 to 14 and 17 to 19 instead of 15 to 17, thus leaving out track.csv columns 8 and 16; it will effectively replace the uncomfortable 7 specifications long sequence

```
sw_track 8 = 9
...
sw_track 14 = 15
```

The switch command and the first drivernumber have to be separated by at least one empty space.

tracklist As an alternative for sw_track specifications you can use a tracklist consisting of 17 comma separated numbers. For example:

```
tracklist = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,18,19
```

takes the listed track columns from your track.csv file.
In this example tracks in columns 15 and 17 were omitted.

track_1 This parameter defines the track number of the first row of replace lines in the driver table (default = 1). It's only used in a CSM mod that allows more than 17 tracks using tracklists. A track table can contain more than 17 tracks. The replace lines of the driver table, that correspond to the tracks, may therefore also contain more than 17 lines.
Then in the INI file that processes the second tracklist, ***track_1*** is set (normally to 18):

```
track_1=18
```

with which the second tracklist gets its driver replace values from the driver tables replace lines 18 (for track 1) and on, while lines 1 to 17 are for the first tracklist.
The value of this parameter is checked for the range 0 < ***track_1*** <= replace_lines.

*You can easily create different gpaedia versions for a mod that supports more than 17 tracks by defining a [tracklist] variable in the settings globalvars.ini file. Put all tracks into your track.csv and select the 17 used ones by **sw_track** (or **tracklist)** settings in your gpaedia INI files (one for every of the two mod-tracklists), e.g. no track switch specifications for mod-tracklist a and for mod-tracklist b set:*

```
sw_track 1 = 18
sw_track 2 = 19
sw_track 3 to 17 = 20
```

while track column 20 contains the empty track "none“; it's then necessary to set

```
track_1 = 18
```

For details read A complex example in chapter "GPaedia and CSM mod“ .
Switch values should be in the range of 1 to the CSV tables last column. This will be checked, so eventually you will encounter switch error messages and a program stop. This could also happen if you have inadvertently set double switch values. Settings of sw_track will be overriden by a tracklist specification.

log_file Selects a name for the log file (default = "gpaedia“; extension is always ".log“).

```
log_file = gpaedia-2
```

will put the log into the file "gpaedia-2.log“ in the ***var_path*** directory.

override 0|1: toggle for override existing files GPS, STR, LOG (Default = 1: yes).
If set to "0", existing files will be renamed and protected in this way.

As usual with INI files, leading and finishing spaces or tabs of the left and right parts (resp. the "=“) are ignored. Any of the following forms is valid:

```
season=2001
season = 2001
season            =  2001
```

Data input

Data input is done in english language into the three tables "driver.csv“, "team.csv“ and "track.csv“.
For every of these tables a CSV format file (i.e. "driverformat.csv“, "teamformat.csv“ and "trackformat.csv“) defines

- the structure of rows
- the first column with the english identifiers
- the translation of the identifiers (columns 2-5)
- the usage of a program function for translation and formatting (column 6 "Transform“)
- the way you want the output be arranged inside the gps (column 7 "Lineformat“).

The CSV files must be saved UTF-16 encoded, e.g. with OpenOffice/LibreOffice save in "Unicode“ resp. "Unicode UTF-16". You can also encode files with an editor like notepad++ ("UCS-2 Little Endian“).

If the INI parameter ***user_format*** is not set, the program reads the standard format files from the data directory ***in_path***. Otherwise the format files are read from the ***var_path*** directory first, which enables you to create a different format for every season with your own special changes.

Standard format CSV files

To get the same structure as the original GP4 you will use the standard format for a season.
The first row of each table (CSV data or format file) is for information only and not processed.

driverformat

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Name					#name	
3	Team	Team	Equipo	Ecurie	Scuderia	#team	
4	Date of Birth	Geb.-Datum	Fecha de nacimie	Date de naissanc	Data di nascita	_birthdate	\$\:\%\n
5	Nationality	Nationalität	Nacionalidad	Nationalité	Nazionalità	_nat2	\$\:\%\n
6	Residence	Wohnort	Residencia	Pays de résidenc	Residenza	_res	\$\:\%\n
7	Grand Prix Starts	Gefahrene Renne	Salidas en el Gra	Départs de grand	Gare disputate	_	\$\:\%\n
8	Points Scored	Gewonnene Punk	Puntos conseguid	Points marqués	Punti conquistati	_decimal	\$\:\%\n
9	Best Result	Bestes Rennerge	Mejor resultado	Meilleur résultat	Miglior risultato	_best	\$\:\%\n
10	Best Qualifying	Bestes Qualifikat	Mejor puesto	Meilleure qualifica	Miglior risultato in	_best	\$\:\%\n
11	First GP	Erster GP	Primer GP	Premier GP	Primo GP	_firstgp	\$\:\%\n
12	%lastyear% Championship Position	Platzierung %last	Puesto en el cam	Classement au c	Posizione nel mo	_best	\$\:\%\n
13	Previous Teams	Frühere Teams	Equipos anteriore	Ecuries antérieur	Scuderie precede	_list-Teams	\$\:\%\n
14	driver replace					#offset	
15	1						
16	2						
17	3						
18	4						
19	5						
20	6						
30	16						
31	17						
32	18						
33	19						
34	20						
35	21						

The blue marked rows - also labeled with the # as first character of the transform value - are mandatory. The #name and #team rows define driver and team names and are essential for the program. They must always be row 2 and 3. The row with the transform value #offset tells the program where the driver replace data will start. It must exist but the row number can change. The name has to begin with "driver replace“, the rest is optional.
In the standard driverformat replace rows for up to 21 tracks are available. If more are needed, this can be done with an userformat file containing additional replace rows.

teamformat

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Team					#team	
3	Team (short)					#team	
4	Team Principal	Teamleiter	Presidente	Président	Presidente	_name	\$\:\%\n
5	Technical Director	Technischer Direk	Director técnico	Directeur techniqu	Direttore tecnico	_name	\$\:\%\n
6	Drivers	Fahrer	Pilotos	Pilotes	Piloti	_list-Names	\$\:\%\n
7	Team Nationality	Herkunftsland des	Nacionalidad del	Nationalité de l'éc	Nazionalità scude	_nat	\$\:\%\n
8	Team Base	Heimatstandort d	Central del equipo	Siège de l'écurie	Sede scuderia	_res	\$\:\%\n
9	Number of years in F1	Jahre bei der F1	Años en F1	Nombre de saison	Numero di anni in	_	\$\:\%\n
10	Number of wins	Zahl der Siege	Victorias	Nombre de victoir	Numero di vittorie	_	\$\:\%\n
11	Number of drivers titles	Zahl der Fahrertite	Títulos de los pilo	Nombre de titres	Numero di mondia	_	\$\:\%\n
12	Number of constructors titles	Zahl der Konstruk	Títulos del constr	Nombre de titres	Numero di mondia	_	\$\:\%\n

The second and third row are reserved for the team name and it's short form. Entries in the "Team“ row 2 get into gpaedia but are actually displayed nowhere, because for them the performance file is responsible.

trackformat

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Circuit name					#	
3	Country					#country	
4	Circuit info name					#countrypart	
5	Address	Anschrift	Dirección	Adresse	Indirizzo	_ignoreLF	\$\:\%\n\n
6	%season% GP	GP %season%	GP del %season%	GP %season%	GP %season%		\$\:
7	GP Date	Datum	Fecha	Date	Data	_gpdate	\%_
8	Laps	Runden	Vueltas	tours	giri	_laps	(\%)\n\n
9	%lastyear% Results	Ergebnis %lastye	Resultados de %l	Résultats %lasty	Risultati %lastyea	_indent1	\$\:\n\%\n\n
10	Race Lap Record	Rundenrekord Re	Récord de vuelta	Record du tour	Record sul giro in	_recordlap-gp	\$\:\%\n\n
11	Qualifying Lap Record	Rundenrekord Qu	Récord de vuelta	Record en qualific	Record sul giro in	_recordlap-gp	\$\:\%\n\n
12	First GP	Erster GP	Primer GP	Premier GP	Primo GP	_	\$\:\%\n\n
13	Number of GPs held	Stattgefundene G	Número de GP al	Nombre de GP co	Numero di GP dis	_	\$\:\%\n\n
14	Circuit Length	Streckenlänge	Longitud de la pis	Longueur du circu	Lunghezza circuit	_circuitlength	\$\:\%\n\n
15	#info					#offset	
16	qualifying lap record	Rundenrekord Qu	Récord de vuelta	Record en qualific	Record sul giro in	_recordlap	\$\:\n\%\n\n
17	race lap record	Rundenrekord Re	Récord de vuelta	Record du tour	Record sul giro in	_recordlap	\$\:\n\%\n\n
18	%lastyear% seasons results	Saisonergebnisse	Resultados de la	Résultats de la sa	Risultati stagione	%lastyear%	\$\n\n
19	pole position	Pole-Position	Pole position	Pole position	Pole position	_fastestlap	\$\:\n\%\n\n
20	fastest lap	Schnellste Runde	Vuelta más rápida	Meilleur tour en c	Giro più veloce	_fastestlap	\$\:\n\%\n\n
21	race winner	Rennsieger	Ganador	Vainqueur	Vincitore	_list-driver+team	\$\:\n\%\n\n
22	corner data	Kurvendaten	Curvas de	vitesse d'entrée d	Dati curve	#offset	
23	corner 01: name gear km/h						
47	. . .						
48	corner 25: name gear km/h						

The track format is a bit more complicated because it contains data for three different displays:
the tracks gpaedia and the "track information“ with the two parts "info“ and "track data“ (= corner data).

Rows 2 to 3 are reserved for the circuits name and country. They are used by the track selection display and also to create STR menu files. Row 4 "Circuit info name“ is used as a header in gpaedias circuit info, so it should exist, but that is not mandatory. Here the transform and lineformat fields can be edited (see userformat 1991 example).
The rows after "Country“ and before "#info“ define the tracks gpaedia display. The "#info“ and the "corner data“ row have to exist to tell the program that the following rows contains data for the track information info resp. track data display.
The variables %season% and %lastyear% are replaced by their INI file values during processing.

The usage of columns "Transform“ and "Lineformat“ is described in detail in the chapter User format CSV files that tells you how to create special format files for a season.

CSV tables

Any data CSV file must have the same row structure and the same header entries (first column) as the corresponding format CSV file. With driver resp. track this is only necessary up to and including "driver replace“ resp. "corner data“.
To accomplish that you can use one of the examples as origin, use an empty template and copy the first column of the format file into it or convert resp. create spreadsheets using convertcsv.

If no data is available for a table item, put in a "N/A“, which will be translated into the other languages.
If you want no gps file output of some entry, even not of the identifier, write "#“ into the cell (must be a not mandatory row).
If a team or track is not used, write "none“ as the drivers, teams or tracks name (for detailed information see chapter empty columns).

The following tables show a season 2009 example including driver replace data.

Driver table

	A	B	C	S	T	AD	AE
1	2009	1	2	...	18	...	28
2	Name	lewis hamilton	heikki kovalainen		giancarlo fisichella		vitantonio liuzzi
3	Team	mclaren mercedes	mclaren mercedes		force india		force india
4	Date of Birth	3/1/1985	19/10/1981		14/1/1973		6/8/1981
5	Nationality	British	Finnish		Italian		Italian
6	Residence	Switzerland	Switzerland		Italy		N/A
7	Grand Prix Starts	35	35		212		39
8	Points Scored	207	83		257		5
9	Best Result	9 wins	1 win		3 wins		6th
10	Best Qualifying	13 poles	1 pole		3 poles		11th
11	First GP	Australia 2007	Australia 2007		Australia 1996		Imola 2005
12	%lastyear% Championship Position	1st	7th		19th		
13	Previous Teams	mclaren	renault, mclaren		minardi, jordan, benetton, sauber, renault, force india		red bull, toro rosso
14	driver replace						
15	1						
16	2						
26	12						
27	13				28		
28	14				28		
29	15				28		
30	16				28		
31	17				28		

Driver replace

Additional drivers may be added to the driver table in columns after 22 which may be used by switching (see gpaedia INI switch specification ***sw_driver***) or replacing drivers.

Replacing a driver (1-22) is easily done by putting in the number of the additional driver into the cell of the original drivers "driver replace“ area. For ***test_gps*** this is the "driver replace“ row, for ***track_gps*** the row below "driver replace“ corresponding to the track given in the track.csv file. In the picture above liuzzi replaces fisichella in races 13-17.

By activating the INI file specification ***track_gps*** you can create trackspecific gps files using the data of additional drivers. The program is looking for driverchanges and generates only the necessary trackspecific gps files. See also chapter GPaedia and CSV mod.
Under ***test_gps*** no trackspecific files are created, but the driver replace values are considered.

Team table

	A	B	C	D
1	2009	1	2	3
2	Team	mclaren mercedes	scuderia ferrari	bmw sauber
3	Team (short)	mclaren	ferrari	bmw sauber
4	Team Principal	Martin Whitmarsh	Stefano Domenicali	Mario Theissen
5	Technical Director	Paddy Lowe	Aldo Costa	Willi Rampf
6	Drivers	lewis hamilton, heikki kovalainen	felipe massa, kimi raikkonen	robert kubica, nick heidfeld
7	Team Nationality	British	Italian	German
8	Team Base	Woking, UK	Maranello, Italy	Silverstone, UK
9	Number of years in F1	43	59	3
10	Number of wins	162	208	1
11	Number of drivers titles	12	15	0
12	Number of constructors titles	8	16	0

The data of row "Team“ should match the team name of the driver table, but this is not mandatory. In a mod the team names of the driver selection menu are defined by performance files. "Team (short)“ defines the team entry in gpaedias team selection list and also the header of the teams gpaedia.
You may fill more than 11 columns with team data. The additional values are checked too. With the gpaedia INI switch specification ***sw_team*** you can easily switch teams by exchanging columns.

Track table

	A	B	C	D
1	2009	1	2	...
2	Circuit name	melbourne	sepang	
3	Country	australia	malaysia	
4	Circuit info name	Albert Park	Sepang	
5	Address	220 Albert Road, South Melbourne, VIC 3205 AUSTRALIA	Sepang International Circuit, Pusat Pentadbiran Litar Jalan Pekeliling, 64000 KLIA, Selangor Darul Ehsan, MALAYS	
6	%season% GP			
7	GP Date	March 29 2009	April 5 2009	
8	Laps	58	56	
9	%lastyear% Results	1) Lewis Hamilton - McLaren 2) Nick Heidfeld - BMW Sauber 3) Nico Rosberg - Williams 4) Fernando Alonso - Renault 5) Heikki Kovalainen - McLaren 6) Kazuki Nakajima - Williams 7) Sebastien Bourdais - Toro Rosso 8) Kimi Raikkonen - Ferrari	1) Kimi Raikkonen - Ferrari 2) Robert Kubica - BMW Sauber 3) Heikki Kovalainen - McLaren 4) Jarno Trulli - Toyota 5) Lewis Hamilton - McLaren 6) Nick Heidfeld - BMW Sauber 7) Mark Webber - Red Bull 8) Fernando Alonso - Renault	
10	Race Lap Record	michael schumacher, 2004, ferrari, 1m 24.125	juan pablo montoya, 2004, williams bmw, 1m	
11	Qualifying Lap Record	michael schumacher, 2004, ferrari, 1m 24.408	michael schumacher, 2004, ferrari, 1m 33.074	
12	First GP	1996	1999	
13	Number of GPs held	13	10	
14	Circuit Length	5.303km/3.295 miles	5.543km/3.376 miles	
15	#info			
16	qualifying lap record	michael schumacher, 2004, ferrari, 1m 24.408	michael schumacher, 2004, ferrari, 1m 33.074	
17	race lap record	michael schumacher, 2004, ferrari, 1m 24.125	juan pablo montoya, 2004, williams bmw, 1m	
18	%lastyear% seasons results			
19	pole position	lewis hamilton, mclaren, 1m 26.572, 15 march 2008	felipe massa, ferrari, 1m 35.748, 22 march 2	
20	fastest lap	heikki kovalainen, mclaren, 1m 27.418, 16 march 2008	nick heidfeld, bmw sauber, 1m 35.366, 23 m	
21	race winner	lewis hamilton, mclaren	kimi raikkonen, ferrari	
22	corner data			
23	corner 01: name gear km/h	Jones 3 145	Turn 1 2 76	
47	. . .			
48	corner 25: name gear km/h			

You can write more than 17 columns with track data. The values of the additional columns are checked too and you can select the 17 tracks for gpaedia using the gpaedia INI switch specification ***sw_track***.

Corner data

a) Title
You don't need to put any track data into the cells of the "corner data“ row.
The corner data title is normally automatically created by the program using the track name from your table. Track "imola“ for example will get the corner data title(s)

```
imola corner data | Imola Kurvendaten | Curvas de Imola |
Imola, vitesse d'entrée dans les virages | Dati curve Imola .
```

It is generated only if ***corner_data*** is set in your INI file, otherwise the corner data title is empty.
You can however fill the track tables "corner data“ row with entries for tracks; those are then taken as corner data titles even if no corner data is created (`corner_data = 0`), but they are not translated.

b) Data

Since version 1.70 the maximal number of corner rows is defined by the corner data used in the file "track.csv" and has no longer the upper limit 25. An empty corner cell ends the tracks corner data.
Otherwise for every corner the data is put into three parts of the cell:
1. The name of the corner; this may be empty (or spaces) or may contain "-“ which is converted to empty. If no corner data is available, put in "N/A“ here. In this case the other two parts are omitted.
2. The gear, written like "5“ or "5th Gear“ or "Gear 5“.
3. The speed as a number in km/h (default) or in mph, if ***speed_in_mph*** = 1 is specified in the INI file. Values are converted into mph resp. kmh using rounded factors 0.625 resp. 1.600 (as in original GP4; exact factors would be 1.609 resp. 0.621) . Since version 1.53 they are then rounded up or down to the next 5er value, e.g. 210 km/h = 131,25 mph => 130 mph.

The default delimiter of this three parts in the track.csv is a linefeed.
You can use an alternative delimiter, e.g. "/“ (allowed are / \ , and ;).
Setting the alternative delimiter by specifying **corner_delimiter** in the INI file, e.g. with
`corner_delimiter = /`
is not necessary anymore because the delimiter is automatically detected from the first corner data entry, that is from corner 1 of track 1.

Examples:

	corner_delimiter LF	corner_delimiter /
1) corner 01 speed_in_mph = 1	Tamburello 3 80	Tamburello / 3 / 80
2) corner 09 speed_in_mph = 0:	Copse 7 280	Copse / 7 / 280

GPaedia output for all languages:

	English	Deutsch	español	Français	italiano
1)	1: Tamburello 3rd Gear 80mph/128kmh	1: Tamburello 3. Gang 80mph/128km/h	1: Tamburello Marcha 3 128km/h	1 : Tamburello Vitesse : 3e 80mph/128km/h	1: Tamburello Terza 80 mph/128 km/h
2)	9: Copse 7th Gear 175mph/280kmh	9: Copse 7. Gang 175mph/280km/h	9: Copse Marcha 7 280km/h	9 : Copse Vitesse : 7e 175mph/280km/h	9: Copse Settima 175 mph/280 km/h

c) Corner data limits

GP4 will accept a maximum of 84 corner data lines in the GPS file below the corner data title and the CrLf. All remaining lines will not be shown in the trackdata display.
As newer circuits sometimes have more than 20 corners and every corner often uses 4 lines (name, gear, speed and CrLf), the available line space is short. For example the Singapore/Marina Bay track has got 23 corners and would need 23*4 = 92 lines with all corners named. Fortunately 15 corners don't have names and therefore only 15*3 + 8*4 = 45 + 32 = 77 lines are used.

To be prepared for tracks with a greater number of corners, since version 1.59 GPaediaMaker has got a method to deal with this. It checks the cornerlines of each track wether they exceed the maximum of 84. If this is the case, for this tracks display the CrLF at the end of each corners data is omitted.
Example: track "Marina Bay" intentionally made bigger with all corner names filled in.
The packed display in the right frame uses one line less for every corner than the normal display left.
A Singapore track with all corners named would use only 23*3 = 69 lines instead of 92 this way.

```
973 [NAK]marina bay corner data

1: Sheares Corner
    3rd Gear
    80mph/131kmh

2: Turn 1
    4th Gear
    95mph/154kmh

3: Turn 2
    2nd Gear
    55mph/90kmh

...
```

```
973 [NAK]marina bay corner data

1: Sheares Corner
    3rd Gear
    80mph/131kmh
2: Turn 1
    4th Gear
    95mph/154kmh
3: Turn 2
    2nd Gear
    55mph/90kmh

...
```

In the log file a check corner data table is added to inform about the number of corners and the used lines. Even the rare case that a corner name needs a line wrap is considered.

```
check corner data
Num Track               Corners   Lines used
 1  melbourne            16        64
 2  sakhir               15        45
 3  shanghai             16        48
...
 14 monza                11        44
 15 marina bay           23        92   69  packed
 16 sochi                19        57
...
```

The trackinformation is enhanced also and will look like this:

```
trackinformation (info and corner data)
no.  track sw_track label       track name            corners, lines used, packed
 1     1     1      951 #         melbourne
 2     3     3      953 #         shanghai
 3     4     4      954 #         baku
 4     11    11     955 #         hockenheim
 ...
 14    15    15     973 #       marina bay corner data   23   69   packed
 15    16    16     974 #       sochi corner data        19   57
 16    17    17     975 #       suzuka corner data       18   65
 17    2     2      976 #       sakhir corner data       15   45
```

Trackinfo files

For every track (and every language) a trackinfo file can be added that contains the "Circuit Info"- text of the track gpaedia. These files have to be edited separately as UTF-16 encoded text and put into the ***var_path*** directory. The header must be included in the trackinfo text at the beginning of the file. Headers are (en,de,es,fr,it) `"Circuit Info:", "Streckeninfo:", "Información sobre el circuito:", "Caractéristiques :"` and `"Note sul tracciato:"`.

Even if the files have got no CRLF at the end of the file, txt2gps_ff does not mind.
If the specification ***trackinfo_file*** is set in the INI file and the trackinfo file names are fitting that entry, the trackinfo text files are automatically processed and put into the gps file(s) at the end of the circuits gpaedia display part, that is after the last identifier before "#info".
If the spec is set and the program does not find a trackinfo file for the track actually processed, no circuit info is put into the created gps file and a WARNING message is written into the log file. For a track with name or country name "test" (or "Test") only an information message is written instead.

Linefeeds

Normally you will put \n into the "Lineformat" field of your (user-)format table to get a linefeed, but sometimes it is useful having linefeeds inside the data.

The (6, 8 or 10) lines in the "%lastyear% Result" data must be generated by internal linefeeds.
A linefeed inside the "Previous Teams" entry will avoid text running into the driver picture.

Because in this case the two functions ***_list-teams*** resp. ***_list-Teams*** automatically add a linefeed after the 2nd, 6th, 10th … team name, you don't have to add them manually, but you can do it, if you want to have a different format, thus deactivating the automatic.

Other functions with automatic linefeeds are ***_fastestlap*** and ***_recordlap*** for "pole position" and "fastest lap" resp. "qualifying lap record" and "race lap record". In these cases manually added linefeeds are not considered. CrLf is added by the program after the second part of the record list.

If the transform function of a line is ***_ignoreLF***, all linefeeds in the data are ignored for gpeadia creation, as done with the tracks "Address" row.

Remarks:

- If you put in N/A for **pole position** or **fastest lap**, be sure to add a second line thus avoiding a temporarily incomplete display.
- Be careful not to create a CRLF (unicode hex: 00 0D 00 0A) instead of a linefeed inside a table cell because this would be interpreted as an end of line in your CSV file and will most certainly create confusion.
- If you want to have an additional line between a header and the following data, don't put linefeeds at top of all of your data cells, but use the Lineformat cell of a (user-)format table instead.

Empty columns

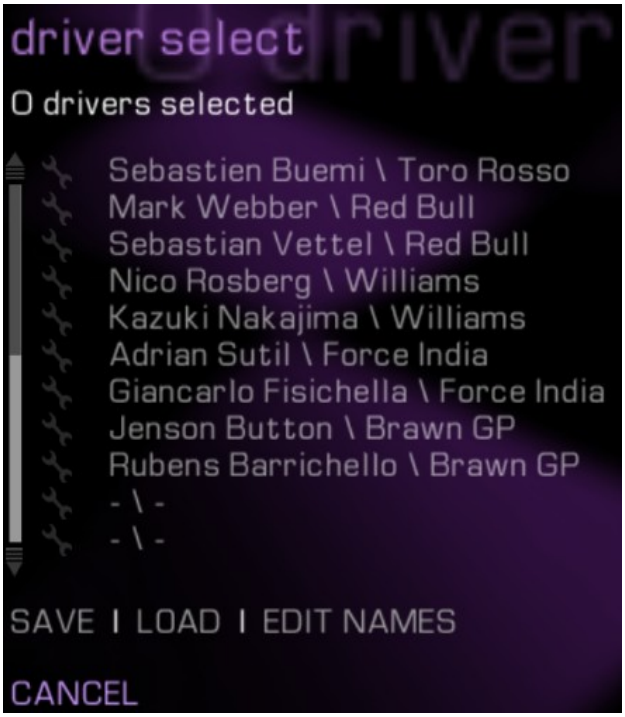
Sometimes a f1 season has got less than 11 teams or less than 17 tracks. In a CSM mod you can omit the cars of a team using tweaking with the file numcars.ini in the gptweak_data folder.
GPaediaMaker has got the feature of using "none“ as a name of a driver, team or track to omit gpaedia output. Rows of a column with the name "none“ can be left empty because they are not checked for values. In the gpaedia selection displays for driver, team and tracks those entries will not show up.
In the driver selection a "- \ -“ will be shown for a driver "none“.
Remember that the GPxPatch performance text file(s) will override gpaedias driver table and thus define the display, so you may have to edit them too.
In the menu display the track selection row of a track "none“ will be empty, but may still be chosen.

In a column named "none“ the second row will automatically be set to "none“ also. These are the rows "Team“ in driver.csv, "Team (short)“ in team.csv and "Country“ in track.csv.

Example: F1 season 2009 is with only 10 teams

2009 driver.csv (first 13 rows):

	A	B	V	W	X
1	2009	1	. . .	21	22
2	Name	lewis hamilton		none	none
3	Team	mclaren mercedes			
4	Date of Birth	3/1/1985			
5	Nationality	British			
6	Residence	Switzerland			
7	Grand Prix Starts	35			
8	Points Scored	207			
9	Best Result	9 wins			
10	Best Qualifying	13 poles			
11	First GP	Australia 2007			
12	%lastyear% Championship Position	1st			
13	Previous Teams	mclaren			



In early releases (before v1.00) "no one“ was used for empty driver columns. This is replaced by "none“ during conversion with the convertcsv tool, as gpaedia driver names must not always consist of two parts.

Multiple Entries

You can put multiple (up to 3) entries into a data CSV files cell that are separated by character "|“.
This feature is only for special cases and normally not used.
Note: it will only work correct with entries in one line (without internal linefeeds).

Example: In 1988 season some teams (e.g. AGS and Osella) have only one driver, so the mod puts two one-driver-teams into one team entry. Using the multiple entry delimiter the CSV file entry looks like

Team	ags ford | osella
Team (short)	ags osella
Team Principal	Henri Julien | Enzo Osella
Designer	Christian Vanderpleyn | Ignazio Lunetta
Team Base	Gonfaron, France | Verolengo, Italy
Team Nationality	French | Italian
GP Debut	Italy 1986 | Argentina 1980
GP Entries	18 | 189
GP Victories	0 | 0
Championship Points	1 | 5
Best Constructors Championship	12th | 12th
Best Drivers Championship	19th | 19th
Pole Positions	0 | 0
Fastest Laps	0 | 0

Spaces before or behind the "|“ are optional; because of trimming, these spaces are not considered.
With the default output delimiter " | “ the following (english) gps output will be generated:

```
Team Principal: Henri Julien | Enzo Osella
Designer: Christian Vanderpleyn | Ignazio Lunetta
Team Base: Gonfaron, France | Verolengo, Italy
Team Nationality: French | Italian
GP Debut: Italy 1986 | Argentina 1980
GP Entries: 18 | 189
GP Victories: 0 | 0
Championship Points: 1 | 5
Best Constructors Championship: 12th | 12th
Best Drivers Championship: 19th | 19th
Pole Positions: 0 | 0
Fastest Laps: 0 | 0
```

You can vary the output delimiter character with the ***tag*** specification in gpaedia.ini, e.g.
`tag = _-_`
will define " - “ (character "_“ is replaced by a space).
Thus you will get

```
Team Principal: Henri Julien - Enzo Osella
Designer: Christian Vanderpleyn - Ignazio Lunetta
Team Base: Gonfaron, France - Verolengo, Italy
Team Nationality: French - Italian
GP Debut: Italy 1986 - Argentina 1980
GP Entries: 18 - 189
GP Victories: 0 - 0
Championship Points: 1 - 5
Best Constructors Championship: 12th - 12th
Best Drivers Championship: 19th - 19th
Pole Positions: 0 - 0
Fastest Laps: 0 - 0
```

Problem: Multiple Entries should not be used if cells of a column were connected into one line display which could be done by omitting a \n at the end of the "Lineformat".

If in the above example the "GP Debut“ row would be defined different using a free format with three rows instead like the following

```
English    | Deutsch  | Español  | Français | Italiano   | Transform  | Lineformat|
-----------+----------+----------+----------+------------+------------+-----------+
           .          .          .          .            .            .
-----------+----------+----------+----------+------------+------------+-----------|
F1 Debut   | F1 Debut | F1 Debut | F1 Début | F1 Debutto |            | \$\:      |
-----------+----------+----------+----------+------------+------------+-----------|
Debut year |          |          |          |            | _          | \%,_      |
-----------+----------+----------+----------+------------+------------+-----------|
Debut GP   |          |          |          |            | _grandprix | \%\n      |
-----------+----------+----------+----------+------------+------------+-----------|
...
```

and the data input using multiple entries would look like

F1 Debut	
Debut year	1986 | 1980
Debut GP	Italian Grand Prix | Argentine Grand Prix

then instead of the wanted
```
F1 Debut: 1986 British Grand Prix - 1980 Argentine Grand Prix
```

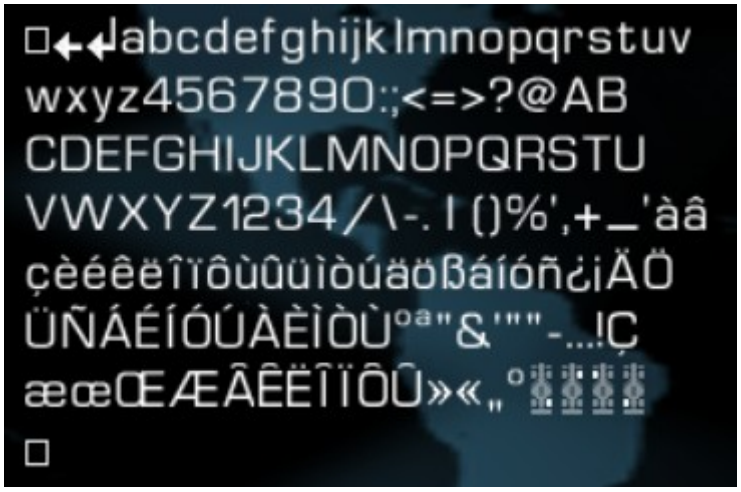
you will get a confused output
```
F1 Debut: 1986 - 1980, Italian Grand Prix - Argentine Grand Prix
```

Restrictions using Unicode

Because the program internally works with Ascii code, it uses Unicode <=> Ansi conversion routines to handle UTF-16LE critical characters with code values greater than 255 (7 of those exist, see below).
You are able to ***use the full character set*** of the style_font1250.gpf for gpaedia in your format and data CSV files. The special structure of the Windows code table 1252 is used to convert critical characters.

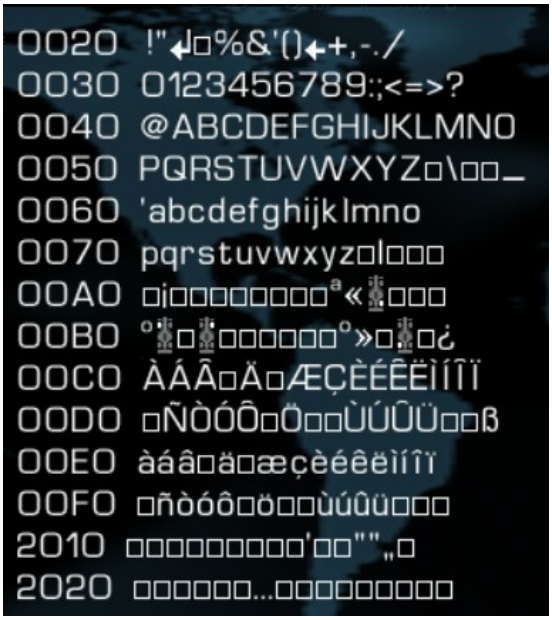
The left image shows the font as displayed in tool "GPIBrowser 1.2“ by Lo2k.
The right shows an example display from a gps file created by GPaediaMaker (v1.28).

GP4 font style_font1250.gpf



GPaediaMaker example

```
§*#abcdefghijklmnopqrstuv
wxyz4567890:;<=>?@AB
CDEFGHIJKLMNOPQRSTU
VWXYZ1234/\-.|()%',+_'àâ
çèéêëîïôùûüìòúäößáíóñ¿¡ÄÖ
ÜÑÁÉÍÓÚÀÈÌÒÙ°ª"&'‟”–…!
ÇæœŒÆÂÊËÎÏÔÛ»«„°¬³±½
$
```



Font displayed in code order

Above the text corresponding to the font image.
Code 21 = U+0015 is used as a special label in gps files.
Several characters from the range 007D to 00BE are not displayable, e.g. §, {, } and the the currency symbols.
Some Unicodes are used as special GP4 characters and their normal usage is therefore not available:

```
U+0024  square (invalid char.)   instead of $
U+002A  arrow left               instead of *
U+0023  return arrow             instead of #
U+00B1  car tire front left      instead of ¬
U+00B3  car tire front right     instead of ³
U+00BD  car tire back right      instead of ±
U+00AC  car tire back left       instead of ½
```

The $ is visible in the font but not displayable. Its code U+0024 is already used as special character.

Critical Unicode characters are those, that use two bytes.

Unicode	Escape	Ansi	Keyboard	Char	Description
0152 = 338	U+0152	008C	Alt+0140	Œ	Latin Capital Ligatur Oe
0153 = 339	U+0153	009C	Alt+0156	œ	Latin small Ligatur Oe
2019 = 8217	U+2019	0092	Alt+0146	’	Right single quotation mark
201C = 8220	U+201C	0093	Alt+0147	“	Left double quotation mark
201D = 8221	U+201D	0094	Alt+0148	”	Right double quotation mark
201E = 8222	U+201E	0084	Alt+0132	„	Doulbe low-9 quotation mark
2026 = 8230	U+2026	0085	Alt+0133	…	Horizontal ellipsis

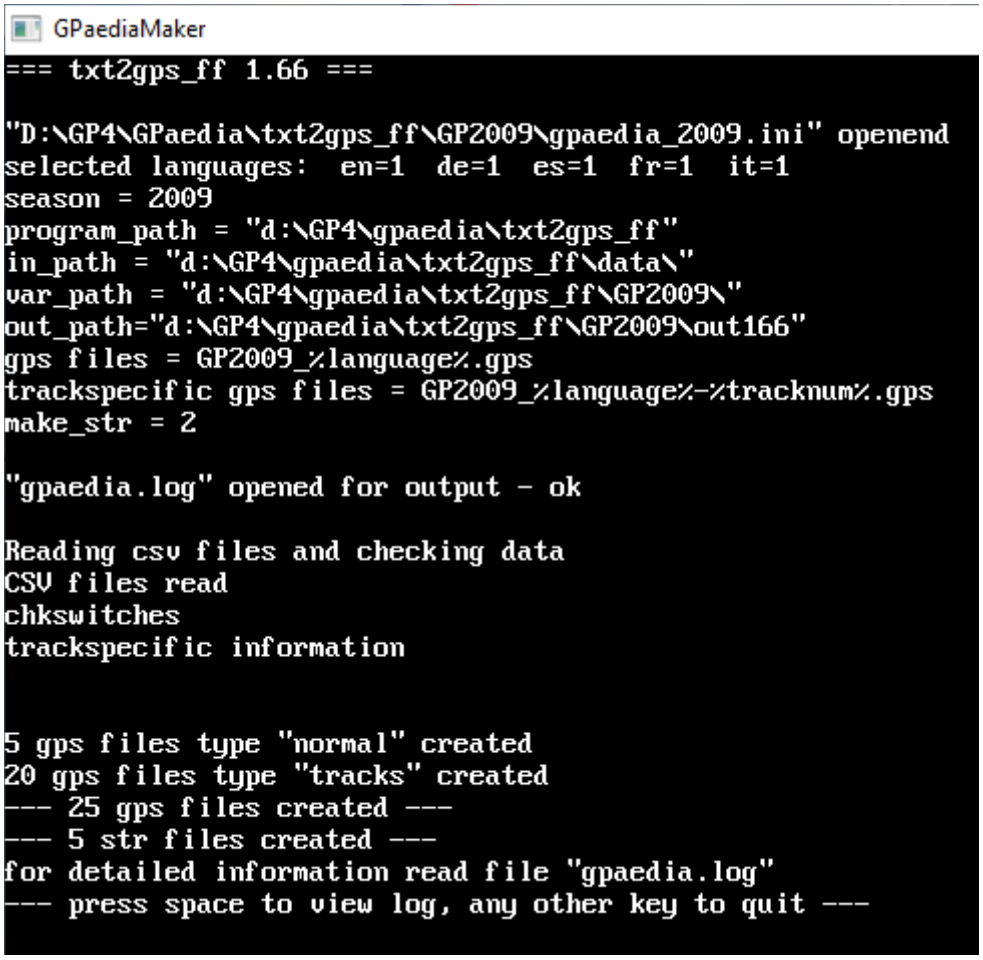
If the program detects a critical Unicode character in one of your tables, it internally switches to "extended mode“ displaying a message on the console and in the log file. In this mode some special routines for input and output are used that can handle this case, though the performance is a bit slower then.
If you have set the INI file specification ***log_extended***, that by default is not active, in the log file below the text file messages all used extended characters are displayed together with the unicode decimal code and the code point character. If the character was found in a CSV file, the file name as well as row and column of the cell in which the character was found are displayed also.

User format text files (part text files and trackinfo files) taken from the ***var_path*** are always processed by routines that can handle Unicode critical characters, regardless of the mode. In extended mode you will notice a "(UTF-16 preserved)“ appendix in the log file message if such a file has been processed successfully. Text files from ***in_path*** are processed by a standard routine except the `%ln%_part9.txt` and the `%ln%_part10.txt` files which are always created in extended mode.

You can write characters used as delimiter resp. text identifyer in the CSV file (e.g. the semicolon and quotation mark) inside a text field. A quotation mark must be doubled there, which often is done automatically by calculation programs.

Running txt2gps_ff

To run the program, just double click on txt2gps_ff.exe, if your INI file is "gpaedia.ini“ in the program path. Else drag and drop your INI file onto the programs name or use the "open with“ option.
If everything is alright, you will see a short information like the following:



The first part of this screen shows your INI file settings, the other shows informations related to files. Only the most important information is displayed on the screen. Details are written into the log file. The complete screen has a maximum size of 800x800px with room for 100 columns and 50 rows.

About errors messages see chapters Checking and Log File below.

Different INI files

Because txt2gps_ff can get the name of the gpaedia ini file as a parameter, you can use different INI files (e.g. "gpaedia2009.ini“, "gpaedia2010.ini“ etc.).

In CSM mods that contain team selection or track selection using more than 11 teams or 17 tracks it is useful to create the gpaedia with more than one INI file.
For example if you have a selection of 5 teams as team11, you can create 5 different INI files, each of them using his own output folder and different team and driver switch parameter.
Have a look at the chapter A complex example in section GPaedia and CSM mode and/or read the GPaediaMaker Tutorial for detailed information.

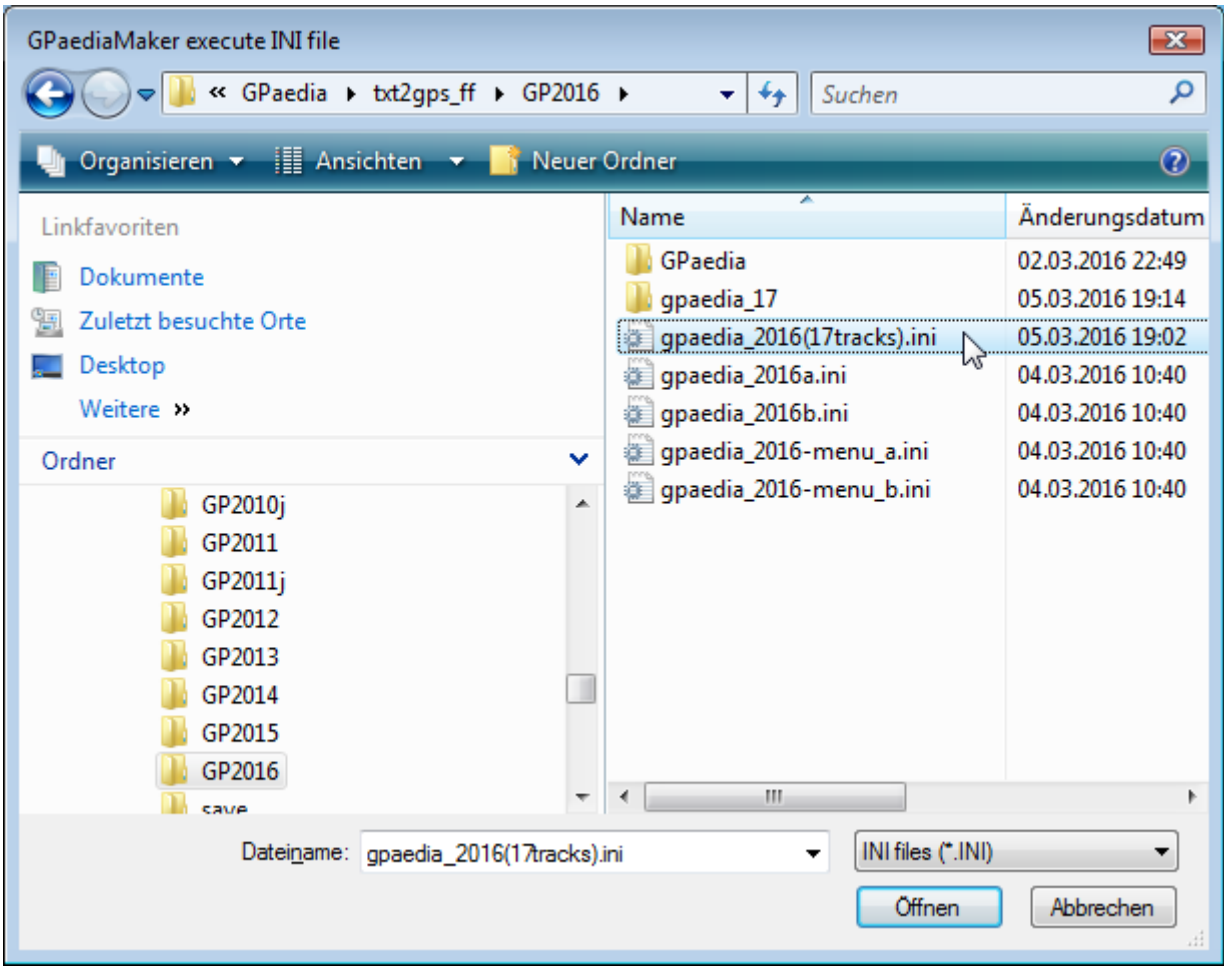
The front-end program "GPaediaMaker.exe" opens a dialog window. Here you can process INI files quickly by doubleclicking.

Front-end

The front-end "GPaediaMaker.exe“ is not necessary to run the processing program "txt2gps_ff.exe“ but gives some additional comfort. There are two ways to use it:

1. Without parameters
2. With one INI file parameter

Without parameters GPaediaMaker.exe will pop up an open file dialog box.
You can successively select an INI file and start processing it with txt2gps_ff.exe by doubleclicking.



With one parameter GPaediaMaker.exe just calls txt2gps_ff.exe, so this is quite the same as executing txt2gps_ff.exe itself.

If GPaediaMaker.exe is not residing in the same directory as txt2gps_ff.exe, than the configuration file **GPaediaMaker.cfg** has to be provided in the same folder as GPaediaMaker.exe with information about the executing program and the starting folder for the open file dialog. It may look like this:

```
;--- GPaediaMaker configuration ---

; Program path and name from txt2gps_ff.exe
program = d:\gpaedia\txt2gps_ff\txt2gps_ff.exe

; Folder for Start of FILE-OPEN-DIALOG
inifolder = d:\gpaedia\txt2gpx-ff
```

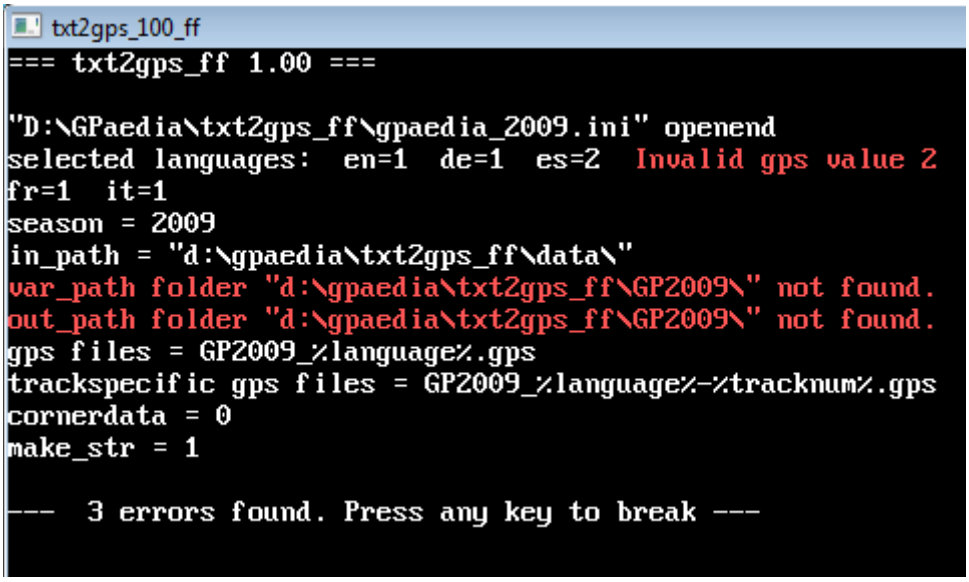
Adjust the specifications to your own folder structure.

Checking

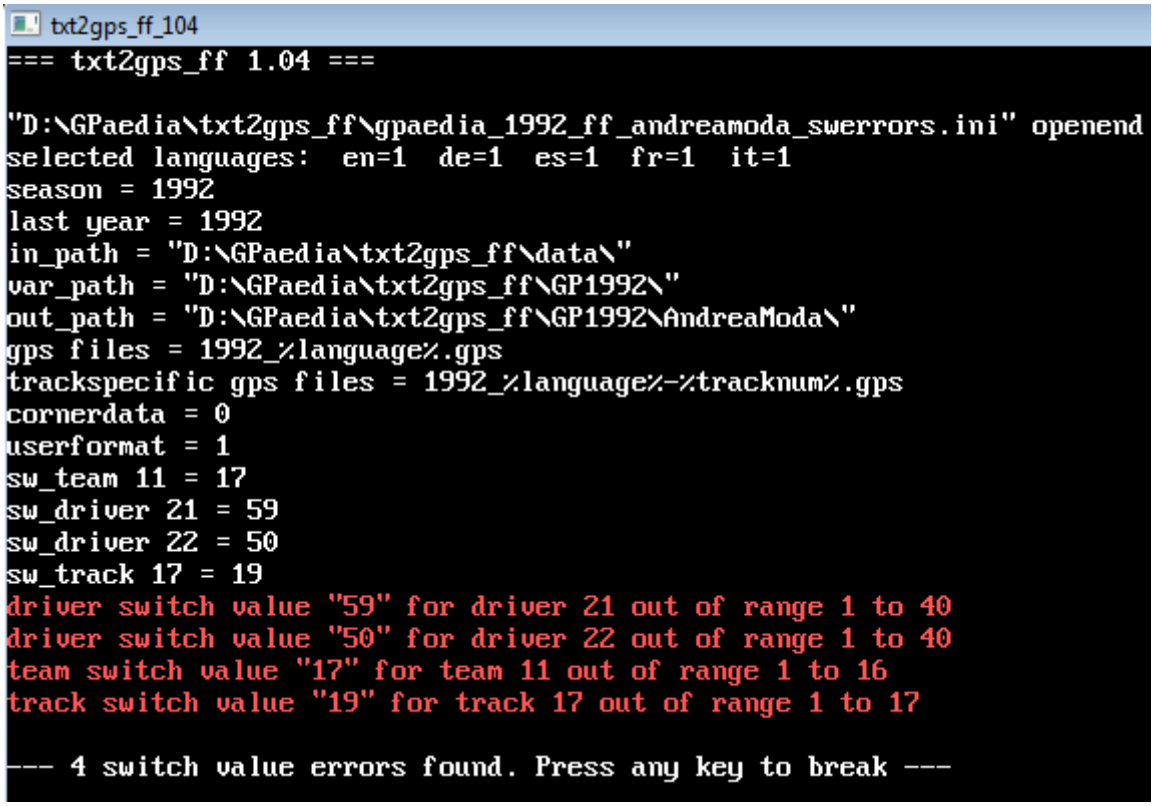
The program performs several checking routines. Two error types may occur: a critical error displayed in red colour will lead to a program stop, a non-critical error is only listed in the log file or eventually displayed at the console in yellow colour .

While reading the INI file, the specifications are checked for validity. After reading the CSV and text files a format check is performed comparing format and data files. Errors found here are always critical.
CSV data files are checked for empty entries and sometimes for correct format: functions ***_fastestlap*** and ***_recordlap*** are checking the correct number of commas, ***corner data*** are checked for a valid format and a valid number of lines used. These checkings are only producing non-critical errors.

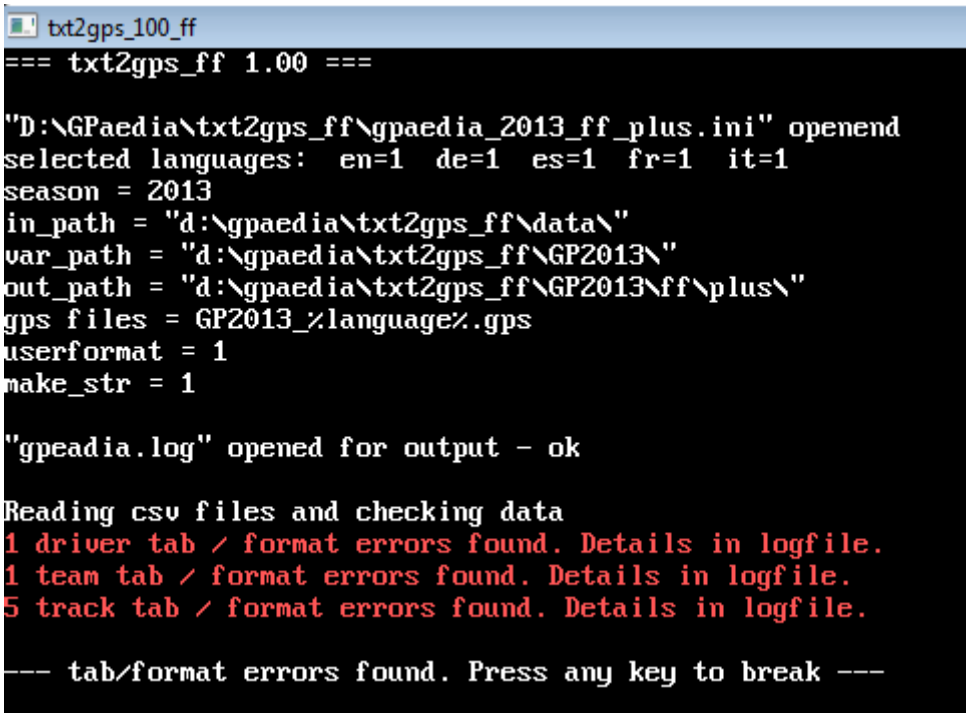
Invalid INI file values will lead to **critical errors**:



Invalid switch values are critical:



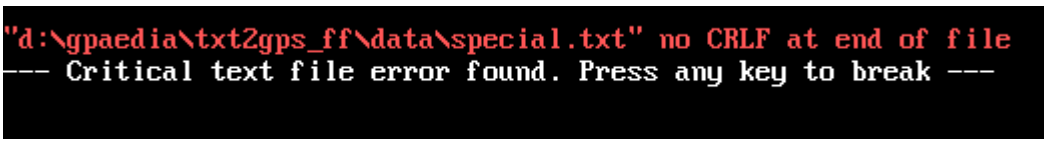
Format errors are also critical, as in this example:



The log files output then shows the details:

```
...
chkformats driver
>>> WARNING: driverformat row 6 : unknown transform value "__"
drivertab_offset= 13
-- 1 driver tab / format errors found -

chkformats teams
>>> WARNING: teamformat row 9 : unknown transform value "_x"
-- 1 team tab / format errors found -

chkformats tracks
>>> WARNING: trackformat row 3 : unknown transform value "_names"
>>> WARNING: trackformat row 10 : unknown transform value "_recordlaps"
info_row  14
>>> WARNING: trackformat row 20 : unknown transform value "_list-driver-team"
tracktab_offset = 21 (corner data - row)
-- 3 track tab / format errors found -
--- Program stopped ---
```

If a text file in the data folder is processed, txt2gps_ff first checks if there is a CRLF at the end of the file. If it is not found, you will get an errormessage like the following:



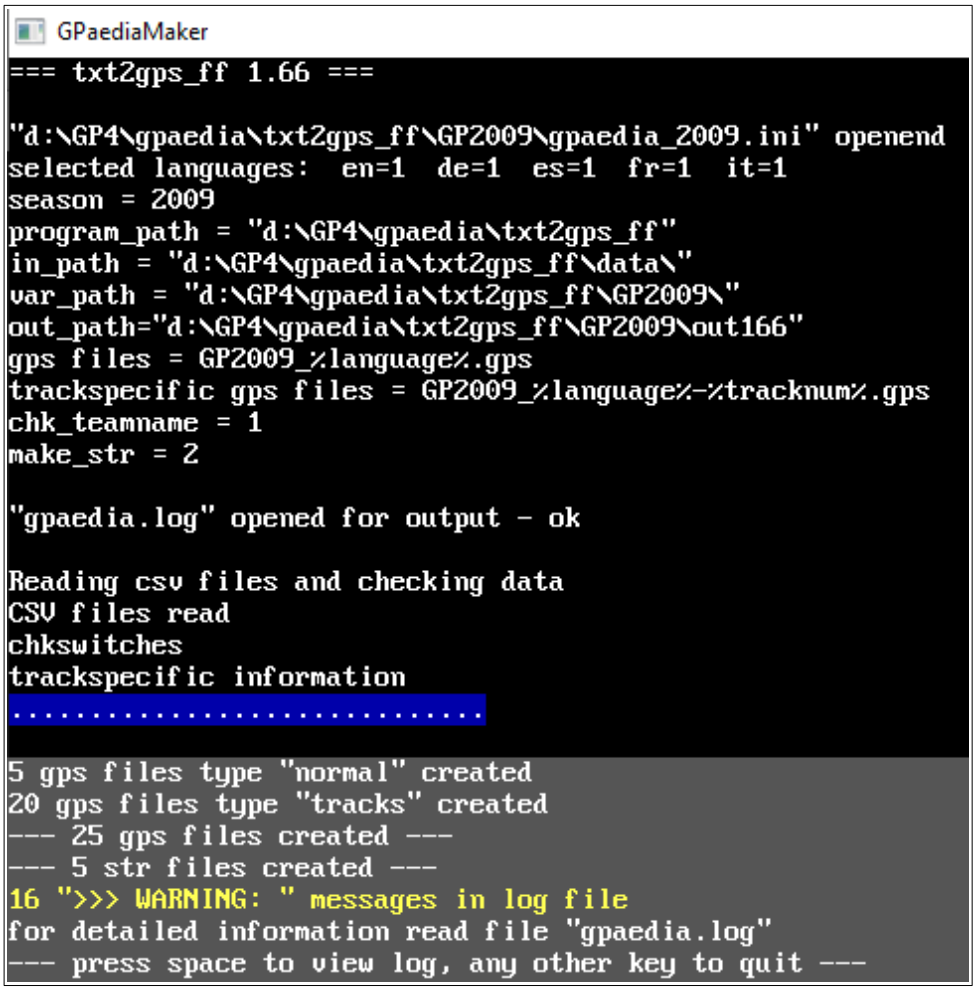
You have to add an empty line at the end of special.txt to be able to retry the program without running into this error again.

Non-critical errors are normally written into the log file or in special cases displayed yellow on screen.

In the example below we see a log file section with warning messages from data checking routines that show mistakes in the tables data input:

```
chkformats tracks
info_row  14
tracktab_offset = 21 (corner data - row)
>>> WARNING: track  1 = "melbourne" row  7  :  "Laps" no data
>>> WARNING: "sebastian vettel, 2011,
red bull, 1m 23,529" track 1 row 10:
             delimiter "," found  4 times instead instead of 3
>>> WARNING: "lewis hamilton, mclaren" track 1 row 18:
             delimiter "," found  1 times instead instead of 2 to 3
>>> WARNING: "
1m 24.922" track 2 row 18:
             delimiter "," found  0 times instead instead of 2 to 3
>>> WARNING: track  18 = "Yeongam" row  18  :  "pole position" no data
```

If warning messages are written into the log file, you will see a yellow summary message in the console output (available since version 1.35). Example 2009:



In the example image the blue and grey background colours make the three areas of the screen visible:

1. The file area above "...". It is visible on screen only, because the log file is not yet opened.
2. The "progress" line (marked blue) with one point for every gps file written.
   This line will be deleted once the gps and str file output is finished.
3. The additonal area (marked dark grey) with warning messages and summary informations.
   It may be scrolled if more lines are written than the screen can take. Thus areas 1 and 2 will not be overridden. It does not matter if some message lines vanish with scrolling, because this information is also available in the log file.

For non-critical errors displayed at the console see chapter Language supporting files.

Log file

The program writes processing information into a log file in the ***out_path*** directory. The default name of this file is "gpaedia.log“, but you may choose a different name using the INI file specification ***log_file***.

In the log file you will find not critical errors written as lines beginning with ">>> WARNING“ which will help you with searching for mistakes in you data input.
The example below shows a normal log file beginning. The first part consists of information and ev. checking messages from reading the CSV format and data files.

```
=== gpaedia.log ===
logfile for txt2gps_ff version 1.60
processing "D:\GP4\GPaedia\txt2gps_ff\GP2019\gpaedia_2019a.ini"

"d:\gp4\gpaedia\txt2gps_ff\data\special.txt" openend
 14             capitals
 10             lcases
 14             unames
 4              country

Reading language specific data

Read from csv-files:

"d:\gp4\gpaedia\txt2gps_ff\GP2019\driverformat.csv":
Table with dimension (37,6) read successfully

"d:\gp4\gpaedia\txt2gps_ff\GP2019\driver.csv":
Table with dimension (37,24) read successfully

"d:\gp4\gpaedia\txt2gps_ff\GP2019\teamformat.csv":
Table with dimension (13,6) read successfully

"d:\gp4\gpaedia\txt2gps_ff\GP2019\team.csv":
Table with dimension (13,11) read successfully

"d:\gp4\gpaedia\txt2gps_ff\GP2019\trackformat.csv":
Table with dimension (47,6) read successfully

"d:\gp4\gpaedia\txt2gps_ff\GP2019\track.csv":
Table with dimension (47,22) read successfully

chkformats driver
drivertab_offset = 16

chkformats teams

chkformats tracks
info_row  15
tracktab_offset = 22 (corner data - row)

check drivertab

check teamtab

"d:\gp4\gpaedia\txt2gps_ff\data\char_width.csv":
Table with dimension (150,1) read successfully

check tracktab
check corner data
Num Track               Corners   Lines used
 1  melbourne            16        64
 2  sakhir               15        45
```

```
 3  shanghai              16        48
 4  baku                  20        60
 5  barcelona             16        60
 6  monte carlo           19        72
 7  montreal              13        52
 8  paul ricard           15        59
 9  spielberg             8         32
 10 silverstone           18        72
 11 hockenheim            13        46
 12 hungaroring           14        42
 13 spa-francorchamps     19        72
 14 monza                 11        44
 15 marina bay            23        78
 16 sochi                 19        57
 17 suzuka                18        65
 18 mexico city           17        51
 19 austin                20        60
 20 interlagos            15        57
 21 yas marina            21        63
 22 none                  0         0

"d:\gp4\gpaedia\txt2gps_ff\data\nationality.csv":
Table with dimension (58,9) read successfully

"d:\gp4\gpaedia\txt2gps_ff\data\grand_prix.csv":
Table with dimension (42,6) read successfully

Trackspecific gpaedia for swap.ini
gptrack13     =  13         Circuit name
gptrack14     =  13         Circuit name
gptrack15     =  13         Circuit name
gptrack16     =  13         Circuit name
gptrack17     =  13         Circuit name
--------------------------------------------------------------------------------
en             gps file - No. 1 of type normal

Init names of textfiles for input:
"d:\gp4\gpaedia\txt2gps_ff\data\en_part1.txt"
"d:\gp4\gpaedia\txt2gps_ff\data\en_part2.txt"
. . .
```

If trackspecific gps is created, the log file gives you information about how to install them into a mod, which you can do by editing the swap.ini and/or the globalvars.ini file(s).
If ***forced_tracks*** is specified in the INI file, then e.g. after `forced_tracks = 9` the log file has got a line `gptrack9      =  9 ;forced`

The lines following the bar will show the log of gps and str file creation.
These parts not only give you information about processing but also about the structure of the gpaedia including the order of driver, team and track data, as in the following log section (*where # is a replacement for the character code 0x15*):

```
driver gpaedia list
no.   driver  sw_driver  replace  label   driver\team
 1      1        1         1      55 #     lewis hamilton\mercedes#
 2      2        2         2      56 #     valterri bottas\mercedes#
 3      3        3         3      57 #     sebastian vettel\ferrari#
 4      4        4         4      58 #     charles leclerc\ferrari#
 5      5        5         5      59 #     max verstappen\red bull#
 6      7        7         7      61 #     daniel ricciardo\renault#
 7      8        8         8      62 #     nico hülkenberg\renault#
 8      9        10        10     63 #     kevin magnussen\haas#
 9      12       12        12     65 #     lando norris\mclaren#
```

```
10      18      18      18      66 #    alexander albon\toro rosso#
11      13      13      13      67 #    sergio pérez\racing point#
12      15      15      15      68 #    kimi räikkönen\alpha romeo#
13      17      17      17      70 #    daniil kvyat\toro rosso#
14      11      11      11      75 #    carlos sainz jr.\mclaren#
14a                             384 #   driver best laps#
15      6       6       6       1018 #  pierre gasly\red bull#
16      14      14      14      1019 #  lance stroll\racing point#
17      16      16      16      1020 #  antonio giovinazzi\alpha romeo#
18      19      19      19      1021 #  george russell\williams#
18a                             1023 #  (not used)#
19      10      9       9       1024 #  romain grosjean\haas#
19a                             1115 #  hotseat <VARIABLE> players selected#
20      21      21      21      1273 #  - \ -#
21      20      20      20      1352 #  robert kubica\williams#
22      22      22      22      1475 #  - \ -#
```

Menu file creation is documented in detail since version 1.05.

In the following ending part of the log you can see, how the national adjectives for "italiano.str“ were created by using the track.csv row 3 ("country“) to search for the adjective in file grand_prix.csv.

```
Strfile "d:\gp4\gpaedia\txt2gps_ff\data\it_strpart3.str" has been processed.
Write grand prix national adjectives
track         sw_track      country              gp adjective
 1             1            australia            " d'Australia"
 2             2            bahrain              " del Bahrain"
 3             3            china                " di Cina"
 4             4            azerbaijan           " d'Azerbaigian"
 5             5            spain                " di Spagna"
 6             6            monaco               " di Monaco"
 7             7            canada               " del Canada"
 8             8            france               " di Francia"
 9             9            austria              " d'Austria"
 10            10           great britain        " di Gran Bretagna"
 11            11           germany              " di Germania"
 12            12           hungary              " d'Ungheria"
 13            13           belgium              " del Belgio"
 14            14           italy                " d'Italia"
 15            15           singapore            " di Singapore"
 16            16           russia               " della Russia"
 17            17           japan                " del Giappone"

Strfile "d:\gp4\gpaedia\txt2gps_ff\data\it_strpart4.str" has been processed.

--- strfile created , language =  4 ---

------------------------------------------------------------------------------

--- 10 gps files created ---

--- 5 str files created ---
```

Extended log

If in the INI file `log_extended = 1` is set, then some additional information is shown in the log. They are about critical UTF characters in CSV or text files or season patches in the part6 text file. A line "`log extended`" is added to the log files header.

a) In a CSV file a critical UTF16 character was found that could not be converted to ANSI:

```
Extended char 8211 U+2013 found in "d:\gp4\gpaedia\txt2gps_ff\GP2018\driver.csv" cell 5, 20
 8211  U+2013
>>> utf2ansi failed: "2013"
```

b) Critical characters in a part text file

```
 8230  U+2026
 8230  U+2026
Text file "d:\gp4\gpaedia\txt2gps_ff\data\data_153\it_part9.txt" has been processed.
```

c) Critical characters in a trackinfo file

```
 339  U+0153
 339  U+0153
Text file "d:\gp4\gpaedia\txt2gps_ff\GP2001org\trackinfo\fr_spa-francorchamps_2001.txt" has
been processed (UTF-16 preserved).
```

d) Season replace in part6 text file

```
Label 471 : 1998 => 2018
Label 1071: 2000 => 2018
Label 1557: 2001 => 2018
Label 1672: 2001 => 2018
Label 2170: 2001 => 2018
Text file "d:\gp4\gpaedia\txt2gps_ff\data\data_153\en_part6.txt" has been processed.
```

Error messages

Critical

File messages

```
INI file not found %gpaedia_ini%
Error when opening file %gpaedia_ini%
Special text file not found %special_txt%
Error when opening special text file %special_txt%
%csvfile% not found
%csvfile% has not UTF-16 file format
UTF-16 text file not found %file%
Error when opening UTF-16 text file %file%
Text file not found %file%
Error when opening text file %file%
No UTF-16 text file found %file%
%txtfile% no CRLF at end of file
--- Critical text file error found. Press any key to break ---
Str file not found %file%
--- Critical strfile error found. Press any key to break ---
```

INI file specification checking

```
program_path folder %progpath% not found.
var_path folder %varpath% not found.
var_path2 folder %varpath2% not found.
in_path folder %inpath% not found.
out_path folder %outpath% could not be created.
subfolder %subfolder% could not be created in %curdir%
Invalid gps value list %lng_lst%: %part_count% parts
Invalid gps value %lng_list(i)%
Invalid character %char% found at position %pos% (forced_tracks)
```

```
Invalid cornerdata value %cornerdata%
Invalid corner_delimiter value %cornerdelim% (only / \ , or ; are allowed)
Invalid speed_in_mph value %speed_in_mph%
Invalid driverlink_caps value %driverlink_caps%
Invalid teamlink_caps value %teamlink_caps%
Invalid tracklink_caps value %tracklink_caps%
Invalid tracklink_name value %trackink_name%
Invalid make_str value %make_str%
Invalid userformat value %userformat%
Invalid valid_rows value %valid_rows%
Invalid valid_driverrows value %valid_driverrows%
Invalid valid_teamrows value %valid_teamrows%
Invalid valid_trackrows value %valid_trackrows%
%zeile%: invalid switch specification
%zeile%: driver index %driver(i)% not valid. (sw_driver)
%zeile%: team index %team(i)% not valid. (sw_team)
%zeile%: track index %track(i)% not valid. (sw_track)
%zeile%: track indices %i% to %k% not valid. (sw_track)
%zeile%: number of entries %k% <> 17 (tracklist)
%zeile% LF Invalid tracklist value %tracklist(i)% at position %i% (tracklist)
Invalid track_1 value %track_1%
Invalid tag value %tag%
Invalid chk_fastestlap value %chk_fastest%
Invalid log_extended value %log_extended%
```

Format checking

```
%errors% driver tab / format errors found. Details in logfile
%errors% team tab / format errors found. Details in logfile
%errors% track tab / format errors found. Details in logfile
```

Switch value checking

```
driver switch value %switch% for driver %driver% out of range 1 to %driver_columns%
double driver switch value %switch% for drivers %driver1% and %driver2%
team switch value %switch% for team %team% out of range 1 to %team_columns%
double team switch value %switch% for teams %team1%" and %team2%
track switch value %switch% for track %track% out of range 1 to %track_columns%
double track switch value %switch% for tracks %track1% and %track2%
Invalid track_1 value %track_1% : greater than %driver_replace_lines%
driver switch index %UBound(driver_sw)% out of range, max = %driver_columns%
team switch index %UBound(team_sw)% out of range, max = %team_columns%
track switch index %UBound(track_sw)% out of range, max = %track_columns%
```

Warning

Console messages

```
%trackspec_errors% track specific errors found. Check dimension of driverformat file.
%logwarning% ">>> WARNING:" messages in log file
-- %errors% driver tab / format errors found --
-- %errors% team tab / format errors found --
-- %errors% track tab / format errors found --
driver_sw %driver% missing for team_sw %team_of_driver%
%txtfile% no CRLF at end of file
No valid corner delimiter found in corner 1 of track 1: %corner1 of track1%
nationality.csv %lng% %country_column% %adjective% %country_name% : Length %len% >
 %lmax% maximum for track linkdata
grandprix.csv %lng% %grandprix_column% %grandprix_no% %grandprix% : Length %len% >
 %lmax% maximum
%lng% track %tracknr% %translated_trackname% :  Length %len% > %MaxTrackLen% max: >
 too long for track selection menu
File %filename% could not be renamed
```

Log file messages

```
UTF-16 file not found %file%
Error when opening file %filename%
%file% has not UTF-16 file format
%txtfile% no CRLF at end of file

track %sw_tracknum% : replace row %irow% greater than drivertab rows %driver_rows%
driver row %row% : invalid conditional number %number%
team row %row% : invalid conditional number %number%
track row %row% : invalid conditional number %number%
driverformat row %row% : unknown transform value %transform%
drivertab row %row% tab = %driver_replace_data% <> format %driver_replace_format%
drivertab row %row% tab = %drivertabname% <> format = %formatdrivername%
'driver replace' row not found in driverformat
teamformat row %row% : unknown transform value %transform%
teamtab row %row% tab = %teamname% <> format = %formatteamname%
%trackformat_csv% row %row% : unknown transform value %transform%
tracktab row %row% tab = %trackname% <> format = %formattrackname%
'#info' row not found in trackformat
'corner data' row not found in trackformat
driver %number% = %drivername% missing drivername
driver %number% = %drivername% missing team
driver %number% %driverteam% <> %teamname% of team %teamnumber%
driver %number% = %drivername% row %row% : %formatdrivername% no data
team %number% %teamname% missing teamname
team %number% = %teamname% row %row% : %formatteamname% no data
track %number% = %trackname% missing circuit name
track %number% = %trackname% row %row% : %formattackname% no data
track %number% %trackname% invalid corner data %cornerdata% at corner %corner% :
%numberof% delimiter
%datalist% %tracknr% row %row%: %CRLF% TAB(14) delimiter %delimiter)% found %k% times
instead instead of %range%
Adjective %part1% not found in nationality.csv for language %lng%
Adjective %part2% not found in nationality.csv for language %lng%
Word 'Grand Prix' or 'GP' missing
Adjective %adjective% not found in grand_prix.csv for language %lng%
replace row %irow% greater than drivertab rows %driver_rows%
%txtfile% no CRLF at end of file
track %track% no gp location found in row 3 'track name': %track_name%
Country %country% : no adjective found in grand_prix.csv for language %lng%
Adjective %nat% : no country in nationality.csv for language %lng%
driver %driver%, replace %entry% out of range 1 to %driver_columns%
no entry found for replace driver number %entry%
track %tracknum% no gp location found in row %row% 'track name': %track_name%
 missing circuit country
Country %country%: no adjective found in grand_prix.csv for language %lng%
Adjective %adjective%: no country in nationality.csv for language %lng%
%tspec_errors% track specific errors found. Check dimension of  driverformat > file.
%lines_used% used corner data lines are more than maximum 84
%circuitlen% invalid circuit length data
File %filename% could not be renamed
```

Extended log file information

is displayed without the `">>> WARNING:"` prefix

```
Label %Label%: %year% => %season%
Extended char %ansicode% U+%Codepoint hex% found in "%csvfile%" cell %row%,%column%
%Ansicode% U+%Codepoint hex%
unicode U+%Codepoint hex%
```

Language supporting files

The files ***nationality.csv***, ***grand_prix.csv*** and the special text file (default: ***special.txt)*** support translation. All three files are part of the ***in_path*** folder, but can also be in the ***var_path*** folder as a season specific file.

In ***nationality.csv*** for every driver and every circuit the country name and it's national attribut are found. If you need more you can add them there. The country name should not be more than 16 characters long, else there will be a faulty display. To avoid this you will encounter an error message.

In ***grand_prix.csv*** every circuit has got the english country name - that has to be the same as the one in the "nationality.csv" - and all national attributes of the grand prix name to support str file generation. If you need more you can add them there.
The maximal length of a grand prix name in the race menu display is 32 characters build from titel plus national adjective, e.g. "Gran Premio de Gran Bretaña“. The program will check this and warn you if entries are too long.
This CSV file is used for creating the STR files and also by the function ***_grandprix*** where for the german language the last column 7 "Großer Preis" is used instead of column 3 "Grand Prix".

The ***special.txt*** has four sections: [capitals], [lowcases], [names] and [country].
Rows with a leading semicolon are overriden.

- Under [capitals] you find a list of team names or name parts that were translated into uppercase like BAR and HRT or partly uppercase like ROKit. Add new names here if necessary.
- Under [lowcases] parts of names are listed that are not converted to uppercase during translation. For example "pedro de la rosa“ will be translated into "Pedro de la Rosa“ because of the "de“ and "la“ in the list. Add new name parts here if necessary.
- The [names] section is related to names with special characters or different spelling in other languages. Entries consist of five parts containing the spelling of the five languages en|de|es|fr|it separeted by delimiter "|“. Spaces at the beginning and the end of each part are ignored.
- The [country] section gives you the possibility to use shorter versions of country names which is often necessary to avoid a currupted gpaedia circuits display as in the following example:

In this french gpaedia circuits display (of 2008 season) you see a gap under the "France“ circuit entry. "Grande-Bretagne“ should show up here, but it's 15 characters long and that is one more than allowed. A track linkdata name should not be more than 13 or eventually 14 characters long though a length up to 16 - especially for track 11 or 17 as the last entries of the column - may eventually be possible. Sometimes a name of critical length is shown, but the link to the circuits gpaedia does not work anymore, e.g. "United Kingdom“.
Character number 14 of the first six names will in most cases overlap with the names of the next column thus only 13 characters are recommended here.

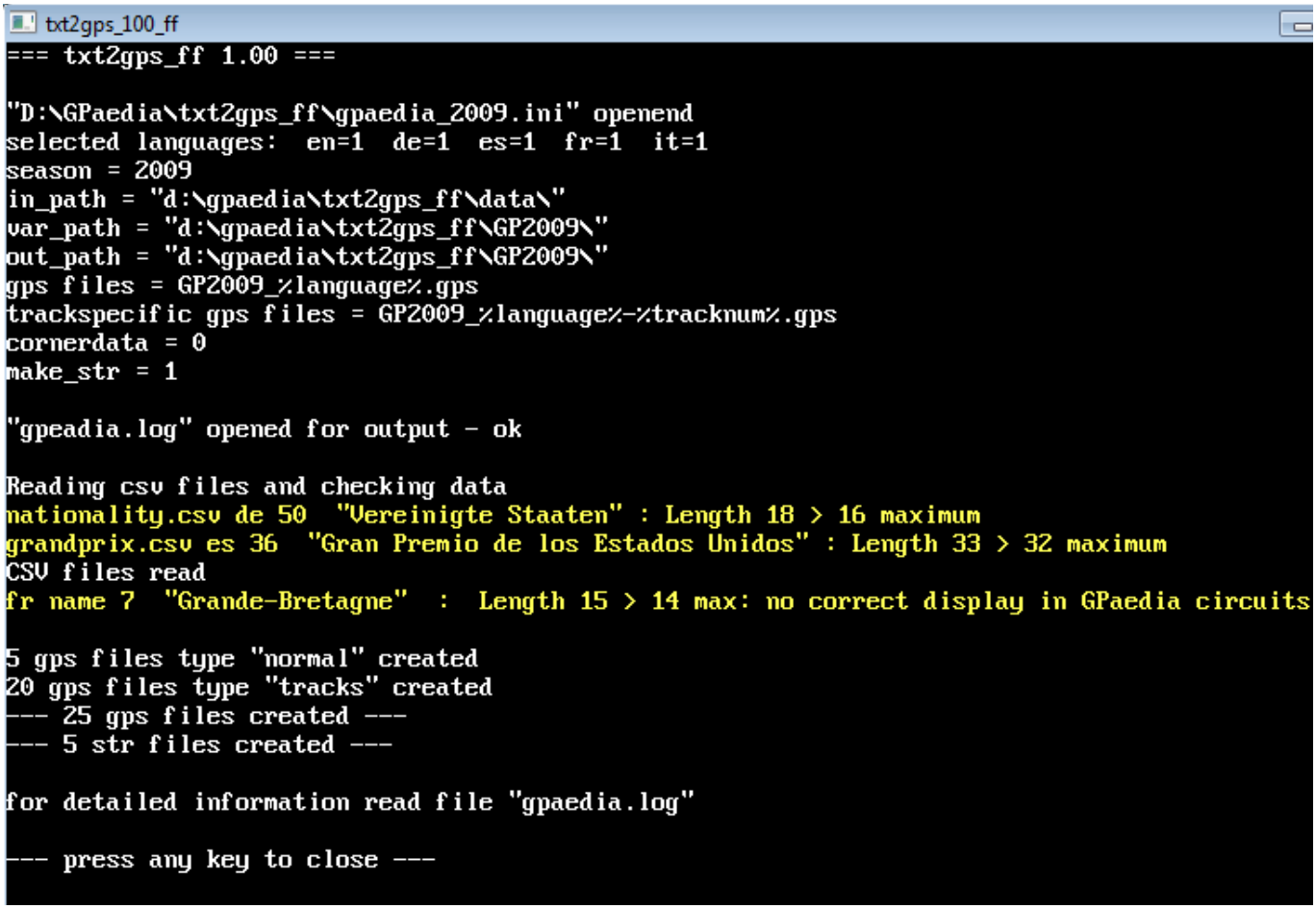
This can be fixed using the following entry in section [country] of special.txt
`Grande-Bretagne|Gr. Bretagne`
The format is "`country|shortcountry`“ where spaces at the beginning or the end of the two parts are ignored. This special text file section is checked for all tracklink names with length>13.

Entries in the [country] section may be different depending on the order of the tracks of the actual season. You can create alternative special text files in the *in_path* folder and activate them using the INI file parameter ***special - s***ee chapter Specification for information about this.

Length checking

The program is checking the length of the cicuit names two times:
1. After reading the *nationality.csv* for a country name length> 16 a warning message is displayed.
2. While processing gpaedia track link data the program is checking the length of either the circuit country or the circuit name - the latter only if ***tracklink_name*** is set to "1". If the tracklink name ist greater than 13, it will look into the special.txt [country] section and take the short version if one is provided there. If the tracklink name is now still too long to fit in the link place - more than 13, 14 or 16, depending on the tracknumber - a warning message will inform you of that problem.

The length of entries in grand_prix.csv is checked after reading the CSV file. The maximal length of the grandprix name plus the grandprix adjective is 32.
Warning messages will look like this on the console:



- The first (yellow) message indicates that this entry in ***nationality.csv*** ("de“ = german column, row 50) is too long and should be exchanged by a shorter version.
- The second message tells you that this entry in ***grand_prix.csv*** ("es“ = spanish column, row 36) is too long and has to be replaced by a shorter one.
- The third message is displayed during processing the gpaedia links. Here we have got a 15 characters long name "Grande Bretagne“ that does not fit in gpaedia circuit selection place 7, where only length 14 is possible (though it could be in place 11, where 16 characters are allowed). To avoid this problem you should add an entry into the special.txt [country] section containing a shorter version, for example:

  ```
  [country]
  Grande-Bretagne|Gr.-Bretagne
  ```

  If this is used, a replace info is added to the section "linkdata tracks" of the log file:
  `Gr.-Bretagne <== Grande-Bretagne`

User format

After setting the ***user_format*** specification in your INI file (`userformat = 1)`, you will have two possibilities to work with this:

1) With **user format CSV** files you can create your own CSV file structure which will mostly be independent of the standard format.
2) With **user format part text files** you can change the otherwise constant gps parts that are not affected by the content of the CSV files.

In both cases the userformat files have to reside in the ***var_path*** folder, normally the seasons folder.
As the program - if in userformat mode - seeks a file first in the ***var_path*** and then in the ***in_path*** folder, you only have to provide the edited files in the former and can leave the unchanged files in the latter.

User format CSV files

If you want to make format files in non-standard form, you can copy the three standard format files driverformat.csv, teamformat.csv and trackformat.csv into your ***var_path*** season directory and change them there.
Do not delete any of the mandatory rows (# - marked in the "Transform“ column), otherwise you won't get a reasonable result. Of course your data CSV files must now have the format of your edited format CSV files, which means that the first (english) column has to be the same, except the less important driverreplace or cornerdata rows. A further exception occurs while using conditional row processing.

Deleting rows is easy. No further action is needed.
Adding new rows is more of a challenge because you have to

- put translations of your identifier column 1 into columns 2-4
- fill the Transform column 6 with the name of the convert function
- fill the Lineformat column 7 with the output structure string.

Transform

A cell of this column contains a function name (starting with a "_“) of the list given in the Build in function overview or can be empty. This function defines how the data CSV values of that row are translated resp. formatted.

Lineformat

Here you put in a string that defines the way the identifier and the data is put into the gps file.
Five substitutes - four of them consist of two characters beginning with a backslash - can be used that are replaced during processing:

Subst	Replacement
\$	Identifier resp. his translation
\:	Language specific colon, that is " : “ for french, ": “ for other languages
\%	Transformed value from you data CSV
\n	CrLf to start a new line
_	The underscore is replaced by a space character; it is needed to put a space at the beginning or the end of a value, because transform functions other than the empty one skip leading or trailing blanks.

Season 1991 example

With free format files you can make gpaedia files the way you want them or you can recreate gps files of mods that came with non-standard format gpaedia thus providing them with full language support.
Let's have a look at the gpaedia of the 1991 mod to see how this is done.

A comparison of 2001 and 1991 gpaedia will help you find out what kind of format changes are necessary (*# in the following examples is a replacement of the character code 21*).

- driver gpaedia 1991
- team gpaedia 1991
- track gpaedia 1991

driver gpaedia 1991

season 1991 example

2001	1991
696 #Team: Mclaren Date of Birth: 28/9/1969 Nationality: Finnish Residence: Monaco Grand Prix Starts: 145 Points Scored: 383 Best Result: 18 wins Best Qualifying: 26 Poles First GP: USA 1991 2000 Championship Position: 2nd Previous Teams: Lotus# 698 #Team: McLaren	Date of Birth: 21/03/1960 Nationality: Brazilian First GP: 1984 Brazil -1991 stats- Grand Prix Starts: 16 Points Scored: 96 Best Result: 7 wins Best Qualifiying: 8 poles Championship position: 1 World Champion!#

1991 driverformat:

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Name					#name	
3	Team	Team	Equipo	Ecurie	Scuderia	#team	
4	Date of Birth	Geb.-Datum	Fecha de nacimien	Date de naissance	Data di nascita	_birthdate	\$\:\%\n
5	Nationality	Nationalität	Nacionalidad	Nationalité	Nazionalità	_nat	\$\:\%\n
6	First GP	Erster GP	Primer GP	Premier GP	Primo GP	_firstgp	\$\:\%\n
7	-1991 stats-	-1991 Statistiken-	-1991 estadística-	-1991 statistiques-	-1991 statistiche-		\$\n
8	Grand Prix Starts	Gefahrene Rennen	Salidas en el Gran	Départs de grand p	Gare disputate	_	\$\:\%\n
9	Points Scored	Gewonnene Punkte	Puntos conseguido	Points marqués	Punti conquistati	_decimal	\$\:\%\n
10	Best Result	Bestes Rennergeb	Mejor resultado	Meilleur résultat	Miglior risultato	_best	\$\:\%\n
11	Best Qualifying	Bestes Qualifikatio	Mejor puesto	Meilleure qualificat	Miglior risultato in	_best	\$\:\%\n
12	Championship Position	Platzierung	Puesto en el camp	Classement au cha	Posizione nel mon	_best	\$\:\%\n
13	driver replace for testing					#offset	
14	driver replace for track: 1						
30	- - -						
31	17						

Row 7 only has the identifier. Transform is empty (no checking) and lineformat does not have an \%.
As the other rows are quite similar to standard format rows and therefore not hard to generate.

Created driver gpaedia (en | es

```
698 #Team: McLaren Honda
Date of Birth: 21/03/1960
Nationality: Brazilian
First GP: Brazil 1984
-1991 stats-
Grand Prix Starts: 16
Points Scored: 96
Best Result: 7 wins
Best Qualifying: 8 poles
Championship Position: 1
```

```
698 #Equipo: McLaren Honda
Fecha de nacimiento: 21/03/1960
Nacionalidad: Brasilieño
Primer GP: Brasil 1984
-1991 estadística-
Salidas en el Grand Prix: 16
Puntos conseguidos: 96
Mejor resultado: 7 victorias
Mejor puesto: 8 puestos de salida
Puesto en el campeonato: 1
```

team gpaedia 1991

season 1991 example

2001 | 1991

```
685 #Chairman & CEO:  Ron Dennis
Technical Director:  Adrian Newey
Drivers: Mika Hakkinen, David Coulthard
Team Nationality:  British
Team Base: Woking, UK
Number of years in F1: 34
Number of wins: 129
Number of drivers titles: 11
Number of constructors titles: 8 #
```

```
686 #Team Principal: Ron Dennis
Drivers: Ayrton Senna, Gerhard Berger
Team Nationality: British
GP Debut: 1966 Monaco
1991 Victories: 8
1991 poles: 10
1991 points: 139
1991 Championship position: 1#
```

1991 teamformat:

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Team					#team	
3	Team (short)					#team	
4	Team Principal	Teamleiter	Presidente	Président	Presidente	_name	\$\:\%\n
5	Drivers	Fahrer	Pilotos	Pilotes	Piloti	_list-names	\$\:\%\n
6	Team Nationality	Nationalität des Te	Nacionalidad del e	Nationalité de l'écu	Nazionalità scuder	_nat	\$\:\%\n
7	First GP	Erster GP	Primer GP	Premier GP	Primo GP	_firstgp	\$\:\%\n
8	1991 victories	Siege 1991	1991 victorias	1991 nombre de vic	1991 numero di vitt	_	\$\:\%\n
9	1991 poles	Poles 1991	1991 puesto de sa	1991 pole	1991 pole position	_	\$\:\%\n
10	1991 points	Punkte 1991	1991 puntos	1991 points	1991 punti	_	\$\:\%\n
11	1991 championship position	Platzierung 1991	Puesto en el camp	Classement au cha	Posizione nel mon	_best	\$\:\%\n

Transform for "Team Nationality“ could also have been "nat2“ to be more compatible with original GP4.
The transform value "_“ in rows 8 to 10 just gives the value as it is, but checking is done.
In row 11 you could also have written "_“ instead of the transform value "_best“.

Created team gpaedia (en | es):

```
686 #Team Principal: Ron Dennis
Drivers: Ayrton Senna, Gerhard Berger
Team Nationality: British
First GP: Monaco 1966
1991 victories: 8
1991 poles: 10
1991 points: 139
1991 championship position: 1
```

```
686 #Presidente: Ron Dennis
Pilotos: Ayrton Senna, Gerhard Berger
Nacionalidad del equipo: Británico
Primer GP: Mónaco 1966
1991 victorias: 8
1991 puesto de salida: 10
1991 puntos: 139
Puesto en el campeonato de 1991: 1
```

track gpaedia 1991

season 1991 example

circuit gpaedia 2001 | 1991

```
666 #Australia Albert Park

Address: 220 Albert Road, South Melbourne
VIC 3205 AUSTRALIA

2001 GP: March 4 2001 (58 Laps)

2000 Results:
 1) Michael Schumacher - Ferrari
 2) Rubens Barrichello - Ferrari
 3) Ralf Schumacher - Williams
 4) Jaques Villeneuve - BAR
 5) Giancarlo Fisichella - Benetton
 6) Ricardo Zonta - BAR

Race Lap Record: Heinz-Harald Frentzen,
1997, Williams, 1m 30.585s

Qualifying Lap Record: Jacques Villeneuve,
1997, Williams, 1m 29.369

First GP: 1996

Number of GPs held: 5

Circuit Length: 3.303 Miles/5.302km

Circuit Info: Smooth track surface,
. . .
#
684 #Autodromo Jose Carlos Pace
```

```
Interlagos/Sao Paulo BRAZIL
24 March 1991
71 Laps
4.309 km
2.679 miles

1991 Results

Classification:
 1) Ayrton Senna, McLaren
 2) Riccardo Patrese, Williams
 3) Gerhard Berger, McLaren
 4) Alain Prost, Ferrari
 5) Nelson Piquet, Benetton
 6) Jean Alesi, Ferrari

Fastest Lap: 1:20.436
Nigel Mansell, Williams

Pole Position: 1:16.392
Ayrton Senna, McLaren

Weather:
Warm and overcast turning wet

Circuit Info: Bumpy track surface,
. . .
#
```

circuit info 2001 | 1991

```
951 #qualifying lap record:
jacques villeneuve, 1997,
williams, 1m 29.369

race lap record:
heinz-harald frentzen, 1997,
williams, 1m 30.585

2000 seasons results

pole position:
mika hakkinen, mclaren,
1m 30.556, 11 march 2000

fastest lap:
rubens barrichello, ferrari,
1m 31.481, 12 march 2000

race winner:
michael schumacher, ferrari#
1991 seasons results
```

```
pole position:
ayrton senna, mclaren

fastest lap:
jean alesi, ferrari

race winner:
ayrton senna, mclaren#
```

circuit trackdata (corner data) in 1991 mod is the same as in original and does not fit the tracks.

	A	B	C	D	E	F	G
1	English	Deutsch	Español	Français	Italiano	Transform	Lineformat
2	Circuit Name					#	
3	Country					#country	
4	Circuit Info Name					_	\%\n
5	Circuit Place					_	\%\n
6	GP Date					_gpdate	\%\n
7	Laps	Runden	Vueltas	tours	giri	_laps	\%\n
8	Circuit Length	Streckenlänge	Longitud de la pis	Longueur du circu	Lunghezza circuit	_circuitlength	\%\n\n
9	%lastyear% Results	Ergebnis %lastye	Resultados de %l	Résultats %lasty	Risultati %lastyea		\$\n\n
10	Classification	Klassifikation	Clasificación	Classement	Classificazione	_indent1	\$\:\n\%\n\n
11	Fastest Lap	Schnellste Runde	Vuelta más rápida	Meilleur tour en c	Giro più veloce	_list	\$\:\n\%\n\n
12	Pole Position	Pole-Position	Pole position	Pole position	Pole position	_list	\$\:\%\n\n
13	#info					#offset	
14	%lastyear% seasons results	Saisonergebnisse	Resultados de la	Résultats de la sa	Risultati stagione	%lastyear%	\$\n\n
15	pole position	Pole-Position	Pole position	Pole position	Pole position	_list-driver+team	\$\:\n\%\n\n
16	fastest lap	Schnellste Runde	Vuelta más rápida	Meilleur tour en c	Giro più veloce	_list-driver+team	\$\:\n\%\n\n
17	race winner	Rennsieger	Ganador	Vainqueur	Vincitore	_list-driver+team	\$\:\%\n
18	corner data	Kurvendaten	Curvas de	vitesse d'entrée d	Dati curve	#offset	

1991 trackformat:

- The third row is mandatory, but here you can edit the transform and lineformat fields. We see "_“ as transform value, a function that will not change the table values. Lineformat in row 4 only jumps to the next line while in standard format an empty line would follow.
- In rows 4 to 8 we have a part of five lines after which an extra line is added by an additional "\n“ at the end of the Lineformat entry.
- In "%lastyear% Results“ the variable %lastyear% will be replaced by the value given in the INI file, i.e. 1991. Actually you could have written "1991“ here directly.
- In "Classification" the transform function ***_indent1*** performs an indent of 1 space at the beginning of every line of the data.
- In rows "Fastest Lap“ and "Pole Position“ the function **_list** is given instead of the usual **_fastestlap**, because the 1991 records do not come in standard format.
- Rows 15 to 17 of the circuit info part are not for records but for driver and team only, thus **_list-driver+team** is the proper transform function here.
- Corner data are omitted.

Created circuit gpaedia 1991 (en | es):

```
666 #Phoenix Street Circuit
Phoenix USA
10 March 1991
81 laps
3.720 km/2.311 miles

1991 Results

Classification:
 1) Ayrton Senna, McLaren
 2) Alain Prost, Ferrari
 3) Nelson Piquet, Benetton
 4) Stefano Modena, Tyrrell
 5) Satoru Nakajima, Tyrrell
 6) Aguri Suzuki, Larousse

Fastest Lap:
1:26.758
Jean Alesi, Ferrari

Pole Position: 1:21.434
Ayrton Senna, McLaren
```

```
666 #Phoenix Street Circuit
Phoenix USA
March de 10 de 1991
81 vueltas
3,720 km

Resultados de 1991

Clasificación:
 1) Ayrton Senna, McLaren
 2) Alain Prost, Ferrari
 3) Nelson Piquet, Benetton
 4) Stefano Modena, Tyrrell
 5) Satoru Nakajima, Tyrrell
 6) Aguri Suzuki, Larousse

Vuelta más rápida:
1:26.758
Jean Alesi, Ferrari

Pole position: 1:21.434
Ayrton Senna, McLaren
```

The weather info is included into the trackinfo file. Weather and circuit info are omitted here.
Created circuit info (en | es):

```
977 #1991 seasons results

pole position:
nigel mansell, williams

fastest lap:
nigel mansell, williams

race winner: nigel mansell, williams
977 #Resultados de la temporada 1991
```

```
Pole position:
Nigel Mansell, Williams

Vuelta más rápida:
Nigel Mansell, Williams

Ganador: Nigel Mansell, Williams
```

1991 track data:

	A	B	C	D	E
1	1991	1	2	3	4
2	Circuit Name	phoenix	interlagos	imola	monte-carlo
3	Country	United States	brazil	san marino	monaco
4	Circuit Info Name	Phoenix Street Circuit	Autodromo Jose Carlos Pace	Autodromo Enzo e Dino Ferr	Circuit de Monaco
5	Circuit Place	Phoenix USA	Interlagos/Sao Paulo BRAZIL	Imola, SAN MARINO	Monte Carlo MONAC
6	GP Date	10 March 1991	24 March 1991	28 April 2004	12 May 1991
7	Laps	81	71	61	
8	Circuit Length	3.720 km / 2.311 miles	4.309 km / 2.679 miles	5.040 km / 3.132 miles	3.328 km / 2.068 mil
9	1991 Results				
10	Classification	1) Ayrton Senna, McLaren 2) Alain Prost, Ferrari 3) Nelson Piquet, Benetton 4) Stefano Modena, Tyrrell 5) Satoru Nakajima, Tyrrell 6) Aguri Suzuki, Larousse	1) Ayrton Senna, McLaren 2) Riccardo Patrese, William 3) Gerhard Berger, McLaren 4) Alain Prost, Ferrari 5) Nelson Piquet, Benetton 6) Jean Alesi, Ferrari	1) Ayrton Senna, McLaren 2) Gerhard Berger, McLaren 3) Jyrki Jarvi Lehto, Dallara 4) Pierluigi Martini, Minardi 5) Mika Hakkinen, Lotus 6) Julian Bailey, Lotus	1) Ayrton Senna, Mc 2) Nigel Mansell, Wil 3) Jean Alesi, Ferrari 4) Roberto Moreno, E 5) Emanuele Pirro, D 6) Thierry Boutsen, L
11	Fastest Lap	1:26.758 Jean Alesi, Ferrari	1:20.436 Nigel Mansell, Williams	1:26.531 Gerhard Berger, McLaren	1:24.368 Alain Prost, Ferrari
12	Pole Position	1:21.434 Ayrton Senna, McLaren	1:16.392 Ayrton Senna, McLaren	1:21.877 Ayrton Senna, McLaren	1:20.344 Ayrton Senna, McLa
13	#info				
14	1991 seasons results				
15	pole position	ayrton senna, mclaren	ayrton senna, mclaren	ayrton senna, mclaren	ayrton senna, mclare
16	fastest lap	jean alesi, ferrari	nigel mansell, williams	gerhard berger, mclaren	alain prost, ferrari
17	race winner	ayrton senna, mclaren	ayrton senna, mclaren	ayrton senna, mclaren	ayrton senna, mclare
18	corner data				

Build in functions overview

function	description and examples
	if transform is empty (or spaces), no translation, formatting or checking is done; this is suitable for identifiers without values or with values that should not be trimmed
_	gives back the trimmed value, no translation or formatting, only checking is done
_best	best result (race or qualifying); entry "World Champion" possible instead of 1st examples (input => en | de | es | fr | it): 1) 1 win => 1 win | 1 Sieg | 1 victoria | 1 victoire | 1 vittoria 2) 5 wins => 5 wins |5 Siege | 5 victorias | 5 victoires | 5 vittorie 3) 1 pole => 1 pole | 1 Pole | 1 puesto de salida |1 pole |1 pole position 4) 2nd => 2nd | 2. | 2.º | 2e | º 5) World Champion => Weltmeister|Campeón mundial|Champion du monde| Campione del mondo 6) N/A => N/A | k.A. | N/P | s.o. | N/A
_birthdate	date of birth; example: `28/9/1969 => 28/9/1969 | 28.9.1969 | 28/9/1969 | 28/09/1969 | 28/09/1969`
_circuitlength	length of a circuit; if the program finds delimiter "/", it will adopt the order of km/miles for the translated output; in spanish only the km value is written if one exists; in french km is converted to m; the length may also be given in km or miles only; then the missing value is calculated and the output is given in format "`x.xxx miles/y.yyykm`" with "miles" translated; a commentary at the end after an "(" will be accepted; if an entry does not fit into any of the allowed formats, a warning message will be put into the log file. examples: `1) 5.419 km/3.367 miles => 5.419 km/3.367 miles | 5,419 km/3,367 Meilen | 5,419 km | 5 419 m/3,367 miles | 5,419 km/3,367 miglia` `2) 3.295 mi / 5.303 km => 3.295 mi/5.303 km | 3,295 mi/5,303 km | 5,303 km | 3,295 mi/5 303 m | 3,295 mi/5,303 km` `3) 5.451km (2007) => 3.388 miles/5.451km (2007) | 3.388 Meilen/5,451km (2007) | 3.388 miles/5.451km (2007) | 5.451km (2007)| 3.388 miglia/5.451km (2007)` `4) 3.37 miles => 3.370 miles/5.422km | 3,370 Meilen/5,422km | 5,422km | 3,370 miles/5,422km | 3,370 miglia/5.422km` `4) 5.451 km/ => 5.451 km | 5,451 km| 5,451 km| 5 451 m| 5,451 km` `6) /3.37 miles => 3.37 miles | 3,37 Meilen | 3,37 miles | 3,37 miles | 3,37 miglia`
_country	the (not case sensitive) country name is translated using the file "nationality.csv"; if not found, no translation is done and an error message is put into the log file; it is possible to specify two countries with "&" or "/" as delimiter example: `UK => UK | GB | Reino Unido | R-U | Regno Unito`
_countrypart	a country name found inside a list of words is translated using the file "nationality.csv"; example (circuit info name): `Turkey Istanbul =>` `Turkey Istanbul | Türkei Istanbul | Turquía Istanbul | Turquie Istanbul | Turchia Istanbul`
_date	english date format "tt month jjjj" converted to actual language; example: see **_fastestlap** date translations
_decimal	replaces "." in a decimal value with "," in all languages other than english
_fastestlap	for fastest race lap resp. fastest qualifying lap; input list "driver, team, record, date" where date may be omitted; transformation of the lists parts is done by the functions **_name**, **_team**, **_recordlap** and **_date** resp.; CrLf is added automatically examples: `lewis hamilton, mclaren, 1m 36.325, 19 October 2008 =>` `lewis hamilton, mclaren,<CrLf>1m 36.325, 19 October 2008 |` `Lewis Hamilton, McLaren,<CrLf>1m 36,325s, 19. Oktober 2008 |` `Lewis Hamilton, McLaren,<CrLf>1min 36.325 s, 19 de octubre de 2008 |` `Lewis Hamilton, McLaren,<CrLf>1'36''325, octobre 19 2008 |` `Lewis Hamilton, McLaren,<CrLf>1m 36.325s, ottobre 19 2008`

function	description and examples
_firstgp	input: a country name and (after a space) another term (usually a year); a left parenthesis also shows the beginning of the second term; country is translated by function **_country**, the term is added as it is; examples: `Australia 2001 => Australia 2001 | Australien 2001 | Australia 2001 | Australie 2001 | Australia 2001` `Europe (Brands Hatch) 1985 => Europe (Brands Hatch) 1985 | Europa (Brands Hatch) Europa (Brands Hatch) 1985 | Europa (Brands Hatch) 1985`
_gpdate	english date "month dd yyyy" or "month dd, yyyy" converted to actual language; the year may be ommitted; examples: `March 29, 2009 => March 29, 2009 | 29. März 2009 | 29 de marzo de 2009 | 29 mars 2009 | 29 marzo 2009` `March 29 => March 29 | 29. März | 29 de marzo | 29 mars | 29 marzo`
_grandprix	input is a national adjective followed by (a space and) the phrase "Grand Prix"; the adjective is translated using the "grandprix.csv" file; the “Grand Prix” phrase is translated inside the program; example: `Spanish Grand Prix => Spanish Grand Prix | Großer Preis von Spanien | Gran Premio de España | Grand Prix d'Espagne | Gran Premio di Spagna`
_ignoreLF	deletes all LF characters from the given string; with this function you can limit a columns width using linefeeds without transfering them into your gps file; example: address of a circuit.
_indent1	makes an indent of 1 space for every line of the given data; checking is not done; example 1991: %lastyear% Results 1) Ayrton Senna, McLaren 2) Alain Prost, Ferrari 3) Nelson Piquet, Benetton 4) Stefano Modena, Tyrrell 5) Satoru Nakajima, Tyrrell 6) Aguri Suzuki, Larousse gpaedia `1990 Results:` `1) Ayrton Senna, McLaren` `2) Alain Prost, Ferrari` `3) Nelson Piquet, Benetton` `4) Stefano Modena, Tyrrell` `5) Satoru Nakajima, Tyrrell` `6) Aguri Suzuki, Larousse`
_laps	expects a number and adds the word "laps" resp. its translation; typically used with several values assembled into one line; example: `66 => 66 laps | 66 Runden | 66 vueltas | 66 tours | 66 giri`
_list	input is a comma separated list; translation is done by converting every first word of a list item to uppercase;
_list-names **_list-Names**	input is a comma separated list of names; translation is done by converting every list item with the **_name** function; **_list-Names** converts english names to uppercase too; example (driver row in team.csv): `eddie irvine, pedro de la rosa =>` `eddie irvine, pedro de la rosa / (en) *with _list-names*` `Eddie Irvine, Pedro de la Rosa | *(en) with _list-Names*` `Eddie Irvine, Pedro de la Rosa | (de, es, fr, it) *with both*`

function	description and examples
_list-teams **_list-Teams**	input is a comma separated list of team names; translation is done using the **_team** function that considers the entries in section [capitals] of the special.txt file. **_list-Teams** converts english teamnames to uppercase too; example („previous teams“ in driver.csv, {LF} stands for a linefeed): `minardi, renault, {LF}mclaren, renault =>` ` minardi, renault, {LF}mclaren, renault / (en) with _list-teams` ` Minardi, Renault, {LF}McLaren, Renault | (en) with _list-Teams` ` Minardi, Renault, {LF}McLaren, Renault | (de, es, fr, it) with both` If no linefeed is found in the data, the program will automatically add linefeeds after 2 and then after every 4th teamname.
_name	input is a space separated list of words; english standard is lowercase; translation to uppercase uses the file "special.txt" for a name affix or a germanic umlaut; special cases ("mclaren" or "heinz-harald") are handled by the program routine. Examples: `1) juan pablo montoya (en) => Juan Pablo Montoya (de,es,fr,it)` `2) heinz-harald frentzen (en) => Heinz-Harald Frentzen (de,es,fr,it)` `3) kimi raikkonen => Kimi Räikkönen (de es, fr, it)`
_nat	Nationality: the national adjective is translated using the file “nationality.csv”; if not found, no translation is done and an error message put into the log file; two adjectives with "/" or "&" as delimiter can be written; "&" will be translated; example: `German & Finnish => German & Finnish | Deutsch & Finnisch |` ` Alemán y Finlandés | Allemand et Finlandais | Tedesco e Finlandese`
_nat2	same as **_nat** but now the adjective for spanish, french and italian language is given in lower case letters in the way the original GP4 gpaedia does. example: `German/Finnish => Nationality: German/Finnish | Deutsch/Finnisch |` ` aalemán/finlandés | allemand/finlandais | tedesco/finlandese`
_points	expects a number and adds the word "points" resp. its translation; typically used with several values assembled into one line; example: `12 => 12 points | 12 Punkte | 12 Puntos | 12 Points | 12 Punti`
_record	record time format transformed from english to actual language time format; example (used by **_fastestlap** or **_recordlap**): `1m 36.325 => 1m 36.325 | 1m 36,325s | 1min 36.325 s | 1'36''325 | 1m 36.325s` `1:24.125 => 1:24.125 | 1:24,125 | 1:24.125 | 1:24.125 | 1:24.125`
_recordlap	for race lap / qualifying lap record as in circuit info display; input list "driver, year (jjjj), team, laprecord"; transformation of the lists parts is done by the functions **_name**, _, **_team** and **_record**; if year longer than 4 (e.g. “10-03-2004”), only the 4 right characters are taken. Linefeeds are ignored; <CrLf> is added automatically in gpaedia. Example: `michael schumacher, 2004,<CrLf>ferrari, 1m 24.408 =>` ` michael schumacher, 2004,<CrLf>ferrari, 1m 24.408 |` ` Michael Schumacher, 2004,<CrLf>Ferrari, 1m 24,408s |` ` Michael Schumacher, 2004,<CrLf>Ferrari, 1min 24.408 s |` ` Michael Schumacher, 2004,<CrLf>Ferrari, 1'24''408 |` ` Michael Schumacher, 2004,<CrLf>Ferrari, 1m 24.408s`
_recordlap-gp	for recordlap display as in circuits gpaedia; in general the same as **_recordlap** but: a) english driver names are written upper case b) linefeeds are ignored, no <CrLf> added after second part. example: `michael schumacher, 2004,{LF}ferrari, 1m 24.408 =>` ` Michael Schumacher, 2004, Ferrari, 1m 24.408 |` ` Michael Schumacher, 2004, Ferrari, 1m 24,408s |` ` Michael Schumacher, 2004, Ferrari, 1min 24.408 s |` ` Michael Schumacher, 2004, Ferrari, 1'24''408 |` ` Michael Schumacher, 2004, Ferrari, 1m 24.408s`

function	description and examples
_res	residence (e.g. of a driver or a team); one of the following two input formats is expected: a) town, country town is not translated, country by function **_country** b) country [delim country2] , where "delim" can be "&" or "/"; country is translated by **_country** function, delim character by a program routine. examples: `1) Switzerland => Switzerland | Schweiz | Suiza | Suisse | Svizzera` `2) Curitiba, Brazil => Curitiba, Brazil | Curitiba, Brasilien | Curitiba, Brasil | Curitiba, Brésil | Curitiba, Brasile` `3) Finland & UK => Finland & UK | Finnland & GB | Finlandia y Reino Unido | Finlande et R-U | Finlandia e Regno Unito` `4) Finland/UK => Finland/UK | Finnland/GB | Finlandia/Reino Unido | Finlande/R-U | Finlandia/Regno Unito`
_speed	input is a number (representing a speed like the number in a corner data cell) in mph (miles per hour) or km/h (kilometers per hour); the value is read corresponding to the ini file specification ***speed_in_mph;*** it is rounded to the next 5s; example (with ***speed_in_mph*** = 0): `145 =>` `90mph/145kmh | 90mph/145km/h | 145km/h | 90mph/145km/h | 90 mph/145 km/h`
_team	input is a list of words separated by spaces representing a team name; translation is done into uppercase words; abbreviations are regarded as specified in part [capitals] of file "special.txt"; "Mc" at the beginning is handled; examples: `1) red bull racing => red bull racing (en)| Red Bull Racing (de,es,fr,it)` `2) bmw sauber => bmw sauber (en) | BMW Sauber (de,es,fr,it)`
%drivers% **%Drivers%**	trackspecific driver names for use with teamtable row "Drivers": input is a comma separated list of names that can contain the variables %drivera% and %driverb% which are replaced by the trackspecific drivernames of the team taken from the entries in the driver table. translation is done by converting every name with the **_name** function; **_%Drivers%** is the same but converts english names to uppercase too; example 2001 (drivers row in team.csv): `eddie irvine, %driverb% =>` `races 1 to 4:` `eddie irvine, luciano burti | (en) with _%drivers%` `Eddie Irvine, Luciano Burti | (en) with _%Drivers%` `Eddie Irvine, Luciano Burti | (de, es, fr, it) with both` `races 5 to 16:` `eddie irvine, pedro de la rosa | (en) with _list-names` `Eddie Irvine, Pedro de la Rosa | (en) with _list-Names` `Eddie Irvine, Pedro de la Rosa | (de, es, fr, it) with both`

User format part text files

Part text files contain the part of a gps or str file that is not changed by the season data that you have put into your CSV files. If you want to create some of them different from the standard form, you can copy the according files from your ***in_path*** folder into your ***var_path*** and change them there using a text editor.

You can make use of user format gpaedia text files, that have names like
`%ln%_part%xx%.txt (%ln% = en,de,es,fr,it and %xx% = 1 to 11)` for gps file generation,
and/or user format menu text files, that have names like
`%ln%_strpart%y%.txt (%y% = 1 to 4)` for str file generation.

Unlike user format CSV files, user format text files will only be used in special cases.

Example Fsone2009 mod:

The Fsone 2009 mod has got some changed data in the season independent parts to create a mod specific appearance. To get the part text files of existing gps files, my tool GPS2TXT is useful.
The gps file parts of the Fsone 2009 contained the following differences compared to the standard gps (only english parts are documented here):

part	Original	Fsone 2009
2	902 []Close the Grand Prix 4 program.[]	902 []Close GP4 FSone 2009.[]
4	83 []championship season[]	83 []championship season 2009[]
4	96 []main menu[]	96 []GP4 FSone 2009[]
6	471 []1998 levels[]	471 []2009 levels[]
6	1071 []2000 levels[]	1071 []2009 levels[]
6	1557 []2001 levels[]	1557 []2009 levels[]
10	731 []exit from game, are you sure?[]	731 []Exit GP4 FSone 2009?[]

The term "[]" stands for the control code U+0015.

These changes were also made in the other languages. Additional changes occurred in the english part5 and part11 only. Therefore the following part text files have to be included into the ***var_path*** folder to create GPaedia preserving the data of the original mod :

```
en_part2.txt, de_part2.txt, es_part2.txt, fr_part2.txt, it_part2.txt,
en_part4.txt, de_part4.txt, es_part4.txt, fr_part4.txt, it_part4.txt,
en_part10.txt, de_part10.txt, es_part10.txt, fr_part10.txt, it_part10.txt,
en_part5.txt,
en_part11.txt.
```

The program will now take these files instead of the standard part text files with the same names provided in the folder ***in_path*** (default: "data").

The part6 text files are not in the list because its modifications were already handled by the program, that replaces the following values with %season% (here "2009"):

label	replaced	new
471	1998	2009
1071	2000	2009
1557	2001	2009
1672	2001	2009
2170	2001	2009

These changes are not only done to the standard part6 text file from the ***in_path*** folder. If you provide a part6 file in the ***var_path*** folder, then in userformat it is read from there and the the programs season actualization is done as well, but this time a value to be replaced could be different and in this case it would have no affect.

Conditonal row processing

In ***userformat CSV*** files you have the option to include rows, that may not be processed or may only be processed after setting one or more of the *INI file specifications* ***valid_rows***, ***valid_driverrows***, ***valid_teamrows*** or ***valid_trackrows.***
This feature allows you to put all available data into your csv files and use the INI file to decide which of them are used. You don't need multiple CSV files this way.

We will look at a driver CSV file for 2016 F1 season as an example, with parts of the driverformat CSV file to the left and the driver data CSV file to the right:

English	. . .	Transform	Lineformat
Name		#name	
Team		#team	
Date of Birth		_birthdate	\$\:\%\n
Nationality		_nat2	\$\:\%\n
Residence		_res	\$\:\%\n
First GP		_firstgp	\$\:\%\n
Grand Prix Starts		_	\$\:\%\n
Best Result		_best	\$\:\%\n
Best Qualifying		_best	\$\:\%\n
#2 Fastest Laps		_	\$\:\%\n
#2 Podiums		_	\$\:\%\n
#1 Points Scored		_decimal	\$\:\%\n
2015 Championship Position		_best	\$\:\%\n
Previous Teams		_list-Teams	\$\:\%\n
driver replace		#offset	

2016	1
Name	lewis hamilton
Team	mercedes amg pet
Date of Birth	07/01/1985
Nationality	British
Residence	Monaco
First GP	Australia 2007
Grand Prix Starts	167
Best Result	43 wins
Best Qualifying	49 poles
Fastest Laps	28
Podiums	87
Points Scored	1867
2015 Championship Position	Champion
Previous Teams	mclaren
driver replace	

Some rows (marked yellow) of the format CSV file contain the conditional marker "#n“ at the first two positions of the first column. In the data CSV file these rows have to exist too, but without the #n.
As second character after "#“ we see "1“ or "2“. Any other character here will say, that this line should not be processed at all. If it has a value greater than 2, a *warning message* is put into the *log file*.
The row identifier is then starting with the 3rd character. As it is trimmed you may add a space for better readability.

If you specify ***valid_driverrows*** = `1`, row #1 will be processed and row #2 will be omitted.
If you specify ***valid_driverrows*** = `2`, then row #1 is omitted and rows #2 will be processed.
If you specify ***valid_driverrows*** = `3`, both rows #1 and #2 will be processed because 3 means 1 or 2.
If nothing is specified, none of the rows after #n are processed.

In the team and track CSV files you can use the same method by specifying ***valid_teamrows*** resp. ***valid_trackrows***.

The specification ***valid_rows*** is a short way to set all valid rows specification at once.
`valid_rows = 1` means
`valid_driverrows = 1` AND `valid_teamrows = 1` AND `valid_trackrows = 1`.

The latter specification overrides the former. After
```
valid_rows = 1
valid_teamrows = 2
```
values are set to valid_driverrows = 1, valid_teamrows = 2 and valid_trackrows = 1.

Conditional markers in mandatory rows - e.g. in the driver CSV in the first two rows (Name and Team) - are ignored.

GPaedia and CSM mod

Copy the gps and str files you just have created into the mods (CSM or not) GPaedia folder. Trackspecific gpaedia is only possible using CSM. You have to write some entries into the swap.ini. To use trackspecific data, select a track in the CSM menu, not the [full season] option that some mods offer mainly to use with the racetype "championship season".

A standard example

Supposed you generated the gps (for season 2009) with

```
ln_gps       = GP%season%_%language%.gps
track_gps    = GP%season%_%language%-%tracknum%.gps.
```

The variable %tracknum% must be the one that is used in the globalvars.ini file as tracknumber.
The picture below shows the driver replace part of a 2009 driver table.

2009	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10											27									
11			23					25			27									
12			23					25			27									
13			24					25			27							28		
14			24					25			27							28		
15			24					25			27							28		
16			24					25		26	27							28		
17			24					25		26	27							28		

Marked are the (six) driver changes and the numbers of the corresponding tracks in the leftmost column. Driver changes occur in all tracks 10 to 17, but as some driver replace rows have the same content than a row above. e.g. 12=11, not for all of them trackspecific gps files will be generated.
The program only creates the necessary files and gives you a protokoll list inside the head of the log-file:

```
Trackspecific gpaedia for swap.ini
gptrack10     =  10         hungaroring
gptrack11     =  11         valencia
gptrack12     =  11         spa-francorchamps
gptrack13     =  13         monza
gptrack14     =  13         singapore
gptrack15     =  13         suzuka
gptrack16     =  16         interlagos
gptrack17     =  16         abu dhabi
```

For %tracknum% 10, 11, 13 and 16 gpaedia files will be created, e.g. `GP2009_English-10.gps`.
To support the other tracks with driver changes we have two alternative possibilities:

- Define a variable list in the mods swap.ini
- Define a new variable with a value list depending on %trackname% in the globalvars.ini

After this the swapping of the gps (and str) files is done in the [Swapfiles] part of swap.ini.

a) Variable list in swap.ini

Using a variable list in the swap.ini has the advantage that eventually the globalvars.ini file has not to be changed.

```
[Vars]
; track specific gpaedia
gptrack10     = 10
gptrack11     = 11
gptrack12     = 11
gptrack13     = 13
gptrack14     = 13
gptrack15     = 13
gptrack16     = 16
gptrack17     = 16

[Swapfiles]
File1  =GPaedia\GP2009_English.gps,MenuData\PC\GP2001\GP2001_English.gps
File2  =GPaedia\GP2009_Français.gps,MenuData\PC\GP2001\GP2001_Français.gps
File3  =GPaedia\GP2009_Italiano.gps,MenuData\PC\GP2001\GP2001_Italiano.gps
File4  =GPaedia\GP2009_Deutsch.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File5  =GPaedia\GP2009_español.gps,MenuData\PC\GP2001\GP2001_español.gps
File6  =GPaedia\GP2009_English-(gptrack%tracknum%).gps,MenuData\PC\GP2001\GP2001_English.gps
File7  =GPaedia\GP2009_Deutsch-(gptrack%tracknum%).gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File8  =GPaedia\GP2009_español-(gptrack%tracknum%).gps,MenuData\PC\GP2001\GP2001_español.gps
File9  =GPaedia\GP2009_Français-(gptrack%tracknum%).gps,MenuData\PC\GP2001\GP2001_Français.gps
File10 =GPaedia\GP2009_italiano-(gptrack%tracknum%).gps,MenuData\PC\GP2001\GP2001_italiano.gps
```

b) Variable list in globalvars.ini

In the "globalvars.ini“ there will bei a [trackname]-part that provides track selection. Generate a new variable beneath the [trackname] specs, in this example we will name it "gptracknum“.
Specifiy the tracknumbers. Sometimes in the [trackname] section the first track belongs to Var2; you have to begin with `Var11=10` etc. in this case.
Add the new variable to the ***modify*** spec at the end of the [trackname] data, to make sure it will be set to the value corresponding to the selected track.

```
[trackname]
...
Modify      =gptracknum,tracknum, ...

[gptracknum]
AllowChange       =0
Name              =Track number for gpaedia
VarCount          =17
Var10             =10
Var11             =11
Var12             =11
Var13             =13
Var14             =13
Var15             =13
Var16             =16
Var17             =16
```

Edit the "swap.ini“ with entries like this:

```
[Swapfiles]
File1  =GPaedia\GP2009_English.gps,MenuData\PC\GP2001\GP2001_English.gps
File2  =GPaedia\GP2009_Deutsch.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File3  =GPaedia\GP2009_español.gps,MenuData\PC\GP2001\GP2001_español.gps
File4  =GPaedia\GP2009_Français.gps,MenuData\PC\GP2001\GP2001_Français.gps
File5  =GPaedia\GP2009_italiano.gps,MenuData\PC\GP2001\GP2001_italiano.gps
File6  =GPaedia\GP2009_English-%gptracknum%.gps,MenuData\PC\GP2001\GP2001_English.gps
File7  =GPaedia\GP2009_Deutsch-%gptracknum%.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File8  =GPaedia\GP2009_español-%gptracknum%.gps,MenuData\PC\GP2001\GP2001_español.gps
File9  =GPaedia\GP2009_Français-%gptracknum%.gps,MenuData\PC\GP2001\GP2001_Français.gps
File10 =GPaedia\GP2009_italiano-%gptracknum%.gps,MenuData\PC\GP2001\GP2001_italiano.gps
```

Instead of the track number you could also use the %trackname% variable.

If you have generated the gps for season 2009 with

```
ln_gps       = GP%season%_%language%.gps
track_gps    = GP%season%_%language%-%trackname%.gps
```

the swapfiles-part will than look like

```
[Swapfiles]
File1  =GPaedia\GP2009_English.gps,MenuData\PC\GP2001\GP2001_English.gps
File2  =GPaedia\GP2009_Deutsch.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File3  =GPaedia\GP2009_español.gps,MenuData\PC\GP2001\GP2001_español.gps
File4  =GPaedia\GP2009_Français.gps,MenuData\PC\GP2001\GP2001_Français.gps
File5  =GPaedia\GP2009_italiano.gps,MenuData\PC\GP2001\GP2001_italiano.gps
File6  =GPaedia\GP2009_English-%trackname%.gps,MenuData\PC\GP2001\GP2001_English.gps
File7  =GPaedia\GP2009_Deutsch-%trackname%.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File8  =GPaedia\GP2009_español-%trackname%.gps,MenuData\PC\GP2001\GP2001_español.gps
File9  =GPaedia\GP2009_Français-%trackname%.gps,MenuData\PC\GP2001\GP2001_Français.gps
File10 =GPaedia\GP2009_italiano-%trackname%.gps,MenuData\PC\GP2001\GP2001_italiano.gps
```

Here we have a problem: because GPaediaMaker generates only the necessary trackspecific gps, some trackspecific gpaedia will not be done; in this case from the list

```
gptrack10     =  10         hungaroring
gptrack11     =  11         valencia
gptrack12     =  11         spa-francorchamps
gptrack13     =  13         monza
gptrack14     =  13         singapore
gptrack15     =  13         suzuka
gptrack16     =  16         interlagos
gptrack17     =  16         abu dhabi
```

for Spa-Francorchamps, Singapore, Suzuka and Abu Dhabi the GPaedia would not be generated.
You can avoid this by using the specificaton ***forced_tracks*** in your ini file with

```
track_gps = GP%season%_%language%-%trackname%.gps
forced_tracks = all
```

that forces creation of every trackspecific gpaedia.

I prefer the track number method because that way it's easier to see what's going on.
You may even use %tracknum% as well as %trackname% in the track_gps specification. That normally does not make sense but may give you more information while testing.

Method **b)** using a variable list in the globalvars.ini is more flexible and thus suited to complicated mods as shown in the chapter Creating GPaedia for a CSM mod with team- and track-selection from the tutorial How to update gpaedia and menu files with a new seasons data .

A complex example

If you are creating gpaedia for a mod with team selection (e.g. team11 as variable), you will need to run the program several times, one for every possible %team11% value.

In the 1988 mod (by tony and friends) we have 6 options for team11: Zakspeed, Larrousse, EuroBrun, Rial-Dallara, AGS-Osella and ColoniSpa (where Rial, Dallara, AGS and Osella are teams with only one driver). We put their data into the team.csv beginning with team number 11 and into the driver.csv beginning with driver number 21 in the given order.
Six runs are needed, each of them using a separate INI file. For the str menu files you can include the `make_str = 2` into only one of the six.

Two parts in the INI file are team11 specific:
a) the ***out_path*** specification like `out_path=%program_path%\GP%season%\%team11%` i.e.

```
out_path=%program_path%\GP%season%\Zakspeed
out_path=%program_path%\GP%season%\Larrousse
etc.
```

b) the ***sw_track*** specifications

Zakspeed: no spec

Larrousse:
```
sw_team 11=12
sw_driver 21=23
sw_driver 22=24
```

EuroBrun:
```
sw_team 11=13
sw_driver 21=25
sw_driver 22=26
```

Rial-Dallara:
```
sw_team 11=14
sw_driver 21=27
sw_driver 22=28
```

AGS-Osella:
```
sw_team 11=15
sw_driver 21=29
sw_driver 22=30
```

Colonispa:
```
sw_team 11=16
sw_driver 21=31
sw_driver 22=32
```

The driver.csv replace data will look like

	A	B	C	D	E	F	G	Q	R	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AO
1	1988	1	2	3	4	5	...	15	...	21	22	23	24	25	26	27	28	29	30	31	32	33	...	37
2	Name	nels	sato	jona	julia	nige		adri		pier	bern	yan	phil	osc	stef	and	alex	phil	nico	gab	rob	mar		pier
3	Team	lotu	lotu	tyrr	tyrr	willi		mina		zak	zak	larr	larr	eur	eur	rial	dall	ags	ose	col	col	willi		larr
	. . .																							
19	driver replace																							
20	1																							
21	2																							
22	3																							
23	4																							
24	5																							
25	6							35																
26	7							35																
27	8							35																
28	9							35																
29	10							35																
30	11					33		35																
31	12					34		35																
32	13							35																
33	14							35																
34	15							35				36												
35	16							34				37												

Coloured lines in the driver replace area shall point out, which trackspecific gps will be created.

Light Blue with driver change number : gps created for every team11
Dark Blue with driver change number : gps not created because it is the same as track 6
Dark Yellow with driver change number : gps created only for team11 = Larrousse

The driver replace check routine creates exactly the trackspecific gpaedia needed. Thus the gps for track 13 will not be created even if the replace number of driver 5 has changed (from 34 to empty = 5) because the replace values are the same as those of track 6.

As a result you get the following trackspecific gps:

6,11,12, 16	for Zakspeed, Eurobrun, Rial-Dallara, AGS-Osella and ColiniSpa
6,11,12,15,16	for Larrousse

To avoid a very long list of specifications in the globalvars resp. swap ini file, you need to have the same gps file list for every team11 run. This is done by forcing trackspecific file creation with the INI files specification ***forced_tracks***. In all INI files except "Larrousse" we add the following specification:

```
forced_tracks = 15
```

As an alternative (with slightly better performance) you may add the complete list in all six INI files:

```
forced_tracks = 6,11,12,15,16
```

With this you should now run the INI files (all but Larrousse again) to generate the correct gps.

The log file header section contains a list "`Trackspecific gpaedia for swap.ini`" with the values to put into the [vars] section of the mods swap.ini file.

Install the gpaedia files by copying the 6 subfolders AGS-Osella, ColoniSpa, EuroBrun, Larrousse, Rial-Dallara and Zakspeed with the gps files into the mods gpaedia folder (i.e. data\Files\GPaedia) and the menu files by copying the folder "ldata" into the mods directory (i.e. data\Files\).
Now the swap.ini file has to be edited. This time the alternative method, that does not change the globalvars.ini, is used to specify the trackspecific number.

```
[vars]
gpaedianum6  =6
gpaedianum7  =6
gpaedianum8  =6
gpaedianum9  =6
gpaedianum10 =6
gpaedianum11 =11
gpaedianum12 =12
gpaedianum13 =6
gpaedianum14 =12
gpaedianum15 =15
gpaedianum16 =16

[Swapfiles]
File1 =Files\GPaedia\%team11%\1988_English.gps,MenuData\PC\GP2001\GP2001_English.gps
File2 =Files\GPaedia\%team11%\1988_Deutsch.gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File3 =Files\GPaedia\%team11%\1988_español.gps,MenuData\PC\GP2001\GP2001_español.gps
File4 =Files\GPaedia\team11%\1988_Français.gps,MenuData\PC\GP2001\GP2001_Français.gps
File5 =Files\GPaedia\team11%\1988_italiano.gps,MenuData\PC\GP2001\GP2001_italiano.gps
File6 =Files\GPaedia\%team11%\1988_English-(gpaedianum%tracknum).gps,MenuData\PC\GP2001\GP2001_English.gps
File7 =Files\GPaedia\%team11%\1988_Deutsch-(gpaedianum%tracknum%).gps,MenuData\PC\GP2001\GP2001_Deutsch.gps
File8 =Files\GPaedia\%team11%\1988_español-(gpaedianum%tracknum%).gps,MenuData\PC\GP2001\GP2001_español.gps
File9 =Files\GPaedia\%team11%\1988_Français-(gpaedianum%tracknum%).gps,MenuData\PC\GP2001\GP2001_Français.gps
File10 =Files\GPaedia\%team11%\1988_italiano-(gpaedianum%tracknum%).gps,MenuData\PC\GP2001\GP2001_italiano.gps
File11 =ldata\english.str,ldata\english.str
File12 =ldata\deutsch.str,ldata\deutsch.str
File13 =ldata\español.str,ldata\español.str
File14 =ldata\français.str,ldata\français.str
File15 =ldata\italiano.str,ldata\italiano.str
```

If for a certain track no gpaedianum is specified, GP4 will take the (not trackspecific) standard file.

See also the chapter Creating GPaedia for a CSM mod with team- and track-selection from the tutorial How to update gpaedia and menu files with a new seasons data .

STR menu files

Since version 0.900 you can create str menu files which do not exactly belong to gpaedia but are closely related to it. With them it is possible to display the ingame racing menus of all five languages with data of the actual season, i.e. the **gp_location** and the **grand_prix_name** (in the examples below "Suzuka" and "Japanese Grand Prix" resp. their translations). This may be desired by users who are starting the game in another language than english.

The **gp_location** is taken from the columns of track table row 2, the tracks name. If the cell is empty, you will be warned by a message in the log file except for a circuit named "none“, were all menu file values - circuit_name, country, and grand_prix_name - were set to empty.
To create the **grand_prix_name** the national adjective is read from the ***grand_prix.csv*** by using the track tables row 3 ("country") value as searching argument - except for track "Test", which is handled by the program. The grand_prix_name is then build by GP4 from the adjective. Examples (country = Brazil):

- english adjective = Brazilian, grand_prix_name = Brazilian Grand Prix,
- español adjective = de Brasil, grand_prix_name = Gran Premio de Brasil.

If the specification ***tracklink_name*** in the gpaedia ini is set to 1 or 4 instead of 0 or 2 (for country), the circuits name (tracktab row 1) resp. the entry from tracktab row 4 (in "track.csv" row 5) is taken instead. This str file country resp. tracklink name is not used by an ingame racing menu but in the "track records" display (main menu => workshop => view circuit records) as title, while the drop down list entries there

come from the gps files country/circuit list.

Menu files will always have the names `english.str, deutsch.str, español.str, français.str` and `italiano.str`. Each of them is put together using 4 part files (en_strpart1.str, en_strpart2.str etc.) and the season dependent data created from the track table (look at the Data input chapter for details).
To create standard str menu files, you have to specifiy in the gpaedia ini file:

```
; Generate str menu files (default: 0)
make_str = 1
```

To use the str menu files just copy them into the ldata folder of you GP4 mod.
Using CSM you will have to write entries into the swap.ini, for example (if File 10 was the last one):

```
[SwapFolder]
…
File11 =ldata\english.str,ldata\english.str
File12 =ldata\deutsch.str,ldata\deutsch.str
File13 =ldata\español.str,ldata\español.str
File14 =ldata\français.str,ldata\français.str
File15 =ldata\italiano.str,ldata\italiano.str
```

If you use menu str files in a mod, you should deactivate the GPxPatch/GPxTrack option ***Catch menu strings*** to avoid some confused race menu title display like "Japanese Grand Prix de Japón".

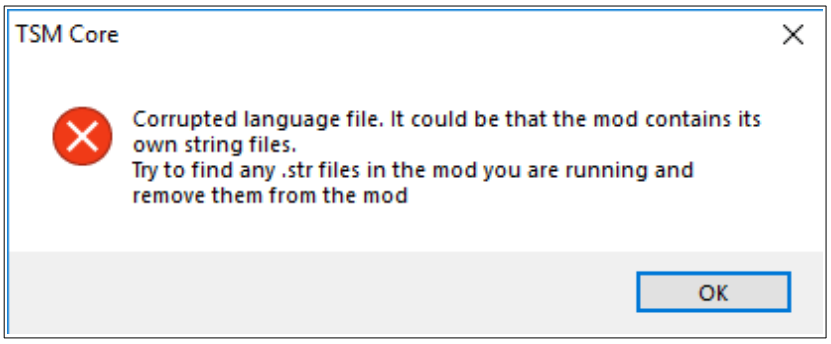
In a mod without menu files this option should be active so that the GPxPatch program can take care of the menu displays reading the country and grand prix adjective from the track ini file. If in this case CSM/TSM is used, it is best to have only one ini file in the tracks folder. Otherwise GPxTrack may not find it because the program does not know it's TSM name but expects a name like the one of the .dat file.

CSM/TSM compatibility

Since version 1.03 you can also specify

```
make_str=2
```

to create menu files compatible with CSM/TSM.
This specification works with every language active while loading the mod by replacing the first tracks gp location name (e.g. "Phoenix“ in 1991 season) in the str file with "Albert Park“, to avoid the TSM core error message "Corrupted language file...“, after which the track set would not be loaded.



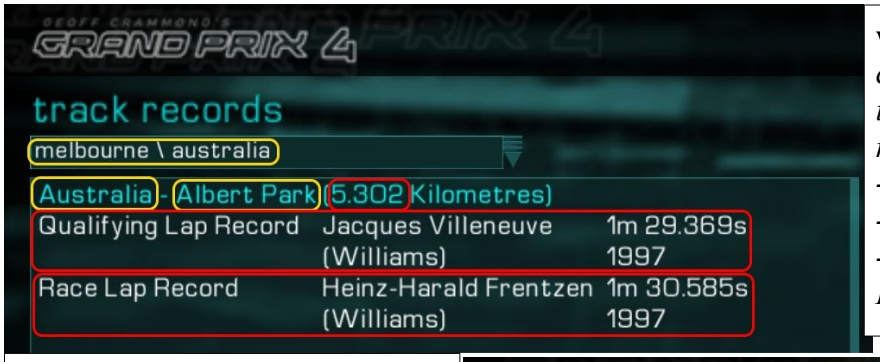
The TSM program then overrides the gp_location of the active str file with the value of the tracks settings.ini file parameter "Name" ("Phoenix" in the 1991 example). Be sure that this is specified as you want it, e.g. in 1991mod track 1:

```
Name=Phoenix.
```

This method supposes that you will run the game in your favourite start-up language.
If you change the language ingame, you will have an unwanted display if the season does not have "Albert Park" as the first track. The gp_location of the first track was not overriden in the chosen str file, because that was not active at start. The now unfitting "Albert Park" will be displayed in the race menus and as title in the workshop/track records menu instead of the seasons first trackname.

Changing track records

In the two displays "track records" of the workshop section and "circuit records" of the ingame menus a qualifying and a race lap record are shown together with the circuit lenght resp. number of laps.
They are not taken from a gpaedia or menu file and therefore cannot be edited by GPaediaMaker directly.



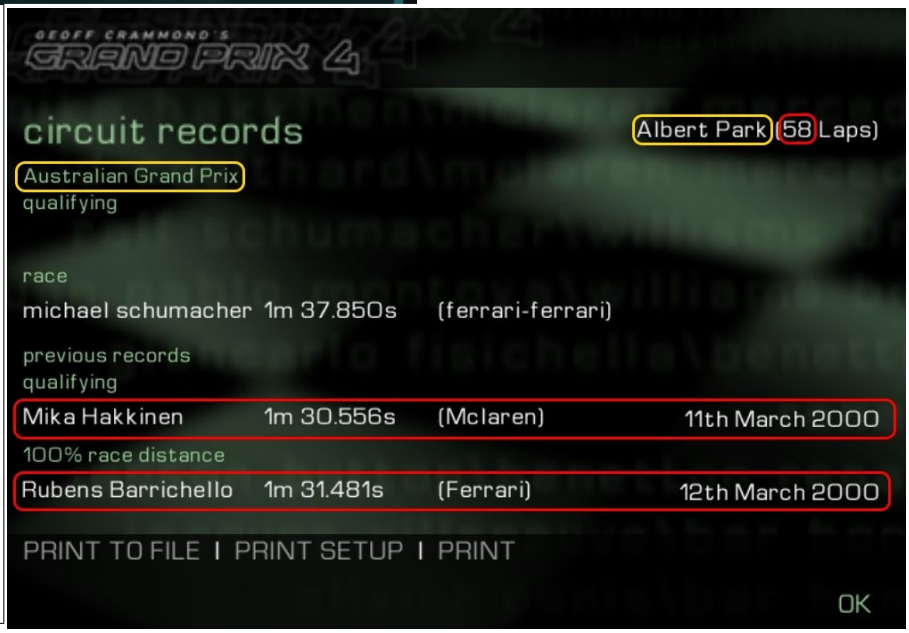
workshop / track records
circuit selection: track ini or menu
title country-gp: track ini or menu
record data from track ini:
- Circuit length
- Qualifying Lap Record
- Race Lap Record
Record date format yyyy.

ingame menu
"circuit records"

circuit name: track ini or menu
gp: track ini or menu or GP4Info

record data from track ini
*- Laps number *)*
- Qualifying record
- Race record
Record date format dd-mm-yyyy

**) If the laps number is not specified in the tracks INI file, it is taken from the tracks GP4Info.*



To update them to a new season you have to use another tool. **GPxPatch**/GPxTrack can read the values from the tracks ini file and overwrite the track records. This method is supported by GPaediaMaker (v1.58 or higher) using the data from the track.csv file that in standard format already has all what is needed, example:

2018	1
Circuit name	melbourne
Country	australia
Laps	58
Race Lap Record	michael schumacher, 2004, ferrari, 1m 24.125
Qualifying Lap Record	lewis hamilton, 2017, mercedes, 1m 22.188
Circuit Length	5.303km/3.295 miles
#info	

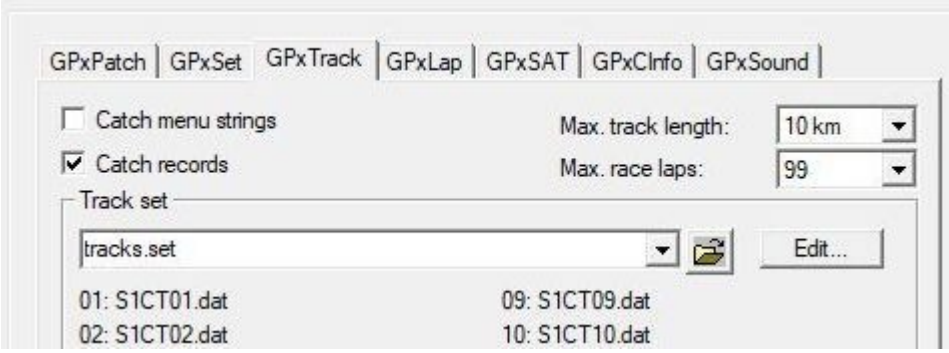
The addon tool ***record_export.exe*** reads the data from the track.csv and exports them into a text file formatted ready to be used by GPxTrack. You will find a detailed description of this tool in it's readme text file. The exported "records.txt" contains the record data (Race Lap Record, Qualifying Lap Record, Laps, Circuit Length) of all tracks.
You can copy GPxTrack sections into a tracks ini file created with TSM. The new parameters are not supported by the TSM-GUI but you can always edit them manually.

Records test file example for a 2018 F1 season:

```
--- Records 2018 ---

;1 = Melbourne
laps=58
laprecord=1m 24.125, Michael Schumacher, Ferrari, 07-03-2004
laprecordqualify=1m 22.188, Lewis Hamilton, Mercedes, 2017
lengthmeters=5303
```

In the GPxTrack help file - section lap record - the formats are given as
`m:ss.ttt, driver, team, dd-mm-yyyy, f`or example:
`1:31.481, Rubens Barrichello, Ferrari, 12-03-2000`
but actually `1m 31.481` and `12/03/2000` will work also. The year minimum is 1978.

To show the new records in the displays you have to check [Catch records] in GPxPatch/GPxTrack. Until version 4.52 of GPxPatch this is only possible with [Catch menu strings] set also which may lead to an ugly display of the race menus grand prix title like "Bahrain Grand Prix de Bahreïn".



[Catch records] set without setting [Catch menu strings] will be possible with GPxPatch v1.53.

To see lap records of all tracks, choose track loading method "all tracks" in the CSM mods configurables. Resulting displays:

Caused by the original GP4 the circuit records display has bugs: in the new records the team name is sometimes not displayed and if the qualifiying time is not a new "(Record)", the format gets distroyed.

You can save the circuit records into the f1gstate.sav or a .rec file, but the new times are not included.

Remarks

If you are using your gpaedia and/or menu files in a mod and encounter displays departing from your gps or str file data, please bear in mind that there are other parts of the mod which may effect the appearence of the displays:

- The performance text file of GPxPatch is responsible for driver selection display.
- Track ini files used by TSM and/or the GPxPatch program will affect the track selection and other track related displays. In addition this depends on the chosen language and the setting of the "Catch menu strings" in GPxPatch/GPxTrack.
- If menu str files are used together witch "Catch menu strings“ checked in GPxTrack you may encounter confused race menu title displays .

A typical failure scenario occurs when the tracknames specified in CSMs "globalvars.ini“ are not consistent with the names of the performance files (=> wrong driver selection display) or the names of trackspecific ini files in the teamart folder (=> wrong driver pictures).
Several parts of the active gps and str file are changed using values from the tracks settings.ini specifications `Name=`, `Country=` and `Adjective=` if they are set:
In the gps file the tracklist (rows behind labels "`21 [NAK]`" to "`43 [NAK]`") with trackname / country is overridden and therefore the values from the ini files are displayed in the track selection menu.
In the str file the tracknames and the adjectives are overridden.

To avoid the grand_prix_name be changed with the adjective from the settings.ini you should delete the value in the ini file.

You may omit the str files and use the above settings.ini file values instead.

Driver 9/10 bug

Be aware of a bug in the original GP4: in the gpaedia drivers 9 and 10 and their gpi files (driver and helmet pictures) are in wrong order. If you generate a mods gpaedia with drivers 9 and 10 in correct order, the gpi images of the drivers may now be in the wrong order in the driver select menu or in drivers gpaedia, depending on your the GP4 version.

a) If you are running GP4 1.00 without the 9.6 patch installed, you can get the correct order of the driver pictures by just exchanging their gpi files, for example by editing the mods INI file(s) in the teamart folder by exchanging the File9 and File10 entries:

```
[Drivers]
  ...
File9=drivers\%driver10%.gpi
File10=drivers\%driver9%.gpi
  ...
```

If in the drivers gpaedia menu you are using the default menu background video, to get the correct helmet display you have to switch helmets gpi also, e.g. by editing the INI file(s) in the teamart folder like

```
[Helmets]
  ...
File9=helmets\%driver10%.gpi
File10=helmets\%driver9%.gpi
  ...
```

b) In GP4 1.02 with 9.6 patch installed the assignment of the driver 9/10 pictures has been changed in the driver select menu, so the above way is not suitable now. Unfortunately then - if you generate the gpaedia with correct driver order - the picture assignment in the drivers gpaedia has not been changed and will now be wrong for drivers 9/10.

In a mod, where the driver selection menu is controlled by performance files, you may exchange drivers 9/10 in the gpaedia using the driver switches

```
sw_driver9 = 10
sw_driver10 = 9 .
```

The pictures would then show up correct in drivers gpaedia, but in the driver selection menu choosing gpaedia of driver 9 or 10 with the right mouse click will now show the wrong display.
Anyway, this seems to be the best method to handle the bug.
The teamart helmets 9/10 should then be swapped as shown for the gp4.exe 1.00 version.

* * *

Driver pictures, track layout maps, track background maps and grid cars are used with gpaedia and/or menu display and should be implemented correctly. See the following list of gpi numbers.

A list of gpi numbers relevant for gpaedia

This list is given as a reference. In a CSM mod you can use drivers and gridcars gpi with (driver resp. team) names and without fiddling with their numbers.

```
--- driver gpi ---
22 driver pictures
gpi: 33, 55-56, 60-64,
66, 68, 70, 72, 74-75,
193, 268-273, 304

22 helmet pictures
(for driver gpaedia)
gpi: 195, 199-218, 336
```

driver	pic	helmet
1	33	201
2	61	217
3	193	213
4	63	195
5	72	218
6	272	207
7	66	203
8	62	206
9	55	210
10	56	215
11	60	205
12	74	208
13	75	209
14	270	199
15	68	214
16	273	211
17	70	200
18	64	216
19	269	202
20	268	336
21	304	204
22	271	212

```
-----------------
Driver numbers 1 to 22 are
related to the drivers
position in the GP4 driver
select menu, not to real
driver numbers.
Column "helmet" means teamart
helmet gpi.
```

```
--- team gpi ---
for grid cars
and team logos
11 logo gpi 49-51, 142-149
11 grid gpi 178-179,182-190
```

team	grid	logo
01	178	51
02	179	144
03	182	148
04	184	50
05	183	49
06	186	143
07	190	149
08	189	147
09	185	142
10	187	145
11	188	146

```
--------------
```

```
--- track gpi ---
17 trackmaps    gpi 14-30
17 countrymaps gpi 275-292
```

track	trackmap	countrymap
1	14	275
2	26	287
3	17	278
4	28	289
5	29	290
6	15	276
7	27	288
8	19	279
9	20	280
10	21	281
11	18	283
12	22	282
13	23	284
14	16	277
15	24	285
16	30	292
17	25	286

```
--------------------------
```

```
-- corner markers --
20 circled numbers
as trackmap corner markers
with tooltips
gpi 155-174
-----------------------------

gpi 155 to 174 (corner numbers
1 to 20) should eventually be
blank to avoid the display of
the numbers in a mod with
different trackmaps.
```

GPaedia and menu displays

with information about associated gpi. Rows with yellow background are relevant for GPaediaMaker.

display	category	value	*	gpi
driver select	gpxpatch	driver\team	*1)	driver, team logo
quicklap driver select	gpxpatch	driver | team	*1)	
circuit select	gpaedia gps	circuit \ country	*2)	trackmap
track information, track information corner data	gpaedia gps	circuit \ country	*2)	trackmap, trackmap + numbers
gpaedia driver list	gpaedia gps	drivers		
gpaedia driver	gpaedia gps	driver, driver information		driver, helmet
gpaedia team	gpaedia gps	team list, team information		
gpaedia track list	gpaedia gps	country	*2)	
gpaedia track	gpaedia gps	country, track information	*2)	countrymap
track records (workshop)	gpaedia gps, menu str f1gstate.sav	circuit \ country (selection list) country – gp location records, circuit length	*2) *5) *6)	
performance data (circuit select)	gpaedia gps	circuit \ country	*2)	
race options	gpaedia gps	season (from INI file spec)		
the grid	menu str	gp location	*3)	gridcars
quicklaps	menu str	grand prix	*3)	
quickrace	menu str	grand prix	*3)	
qualifying	menu str	grand prix	*3)	
qualifying results	menu str	grand prix	*3)	
race	menu str	grand prix	*3)	
current race positions	menu str	grand prix	*3)	
accelerated time	menu str	grand prix	*3)	
results	menu str	grand prix	*3)	
circuit records	menu str f1gstate.sav	grand prix, gp location records, laps	*3) *6)	
race results	menu str	gp location	*3)	
lap chart	menu str	grand prix	*3)	
driver best laps	menu str	gp location	*3)	
driver race points	menu str	gp location	*3)	
constructor race points	menu str	gp location	*3)	
championship, season's results	gpxpatch gpaedia	driver country	*3) *4)	
drivers championship table	gpxpatch	driver	*3)	
constructors championship table	gpxpatch	team	*3)	
next grand prix	gpxpatch	driver	*3)	
season's results	gpaedia gps	country	*3)	
select car setup	gpaedia gps	circuit \ country	*2)	

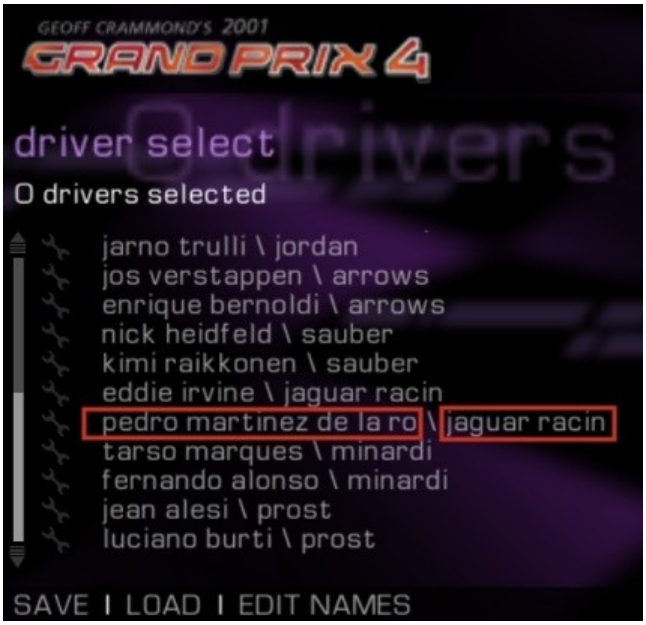
```
Example:   circuit = Melbourne, country = Australia,
           grand prix = Australian Grand Prix, gp location = Albert Park
```

*1) taken from gp4.exe; in a mod from memory written by GPxPatch, controlled by performance file(s)
*2) taken from gps file circuit/country list (label 21-37); in a mod it is controlled by GPxPatch using the tracks ini file or by GP4Info data in the tracks .dat file;

*3) racing menus; driver and team names are originally taken from the gp4.exe; in a mod the names are taken from memory set by GPxPatch, controlled by the performance file(s);
the circuits name (gp location), the country and the grandprix name (adjective) are taken from the tracks ini file instead of the menu str, if "Catch menu strings" is checked in GPxTrack;
the gp location (circuit name) is always taken from the track ini file if the main language is active;
*4) this country name is taken from the linkdata value, not from the circuit/country list;
*5) taken from the track ini file if "Catch menu strings" is checked in GPxTrack;
*6) circuits records (qualifying and race lap record, laps, circuit length) are not part of gps or str files;
the actual circuit records are taken from the f1gstate.sav file; they can be saved into a .rec file;
the original values may be restored from the gp4.exe; circuit records can be changed by GPxPatch (GPxTrack) using the tracks ini file (see chapter "*Changing track records*").

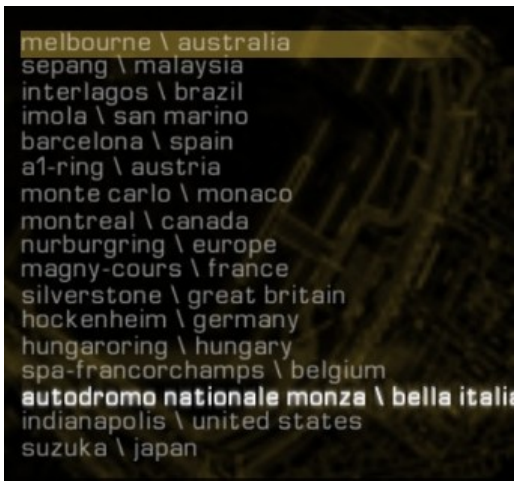
Name length limits

How long can the name of a driver, team, track or country be without making problems in a GP4 display?



a) driver and team names:
The example above shows, that the maximum name length for driver is 23 and for team 12. Together they should be not longer than 29 characters. Actually it's best not to go to maximum lengths.

b) circuit and country names:



Circuit names should not be longer than 19. Country name lengths have a maximum from 13 to 16 as already discussed in chapter Language supporting files.

Convertcsv

The program **convertcsv.exe** is a tool for GPaediaMaker that can
a) convert CSV files from older releases into the new free format of version 1.00
b) create new CSV files without data.
It's action is controlled by an INI file that can be given as a command parameter (default: "convert.ini“).

Copy the **exe** and the **ini** file into the program directory of your choice. No installation is needed.
Because convertcsv uses data and paths of the new GPaediaMaker release, this should already exist.
Processing information including possible errors is shown at the console and in the log file "convert.log“ in the ***out_path*** directory defined in convert.ini.
The directory for the converted CSV files is created if it does not already exist.
Into a new CSV file a header line containing the season value and column numbers will be included.

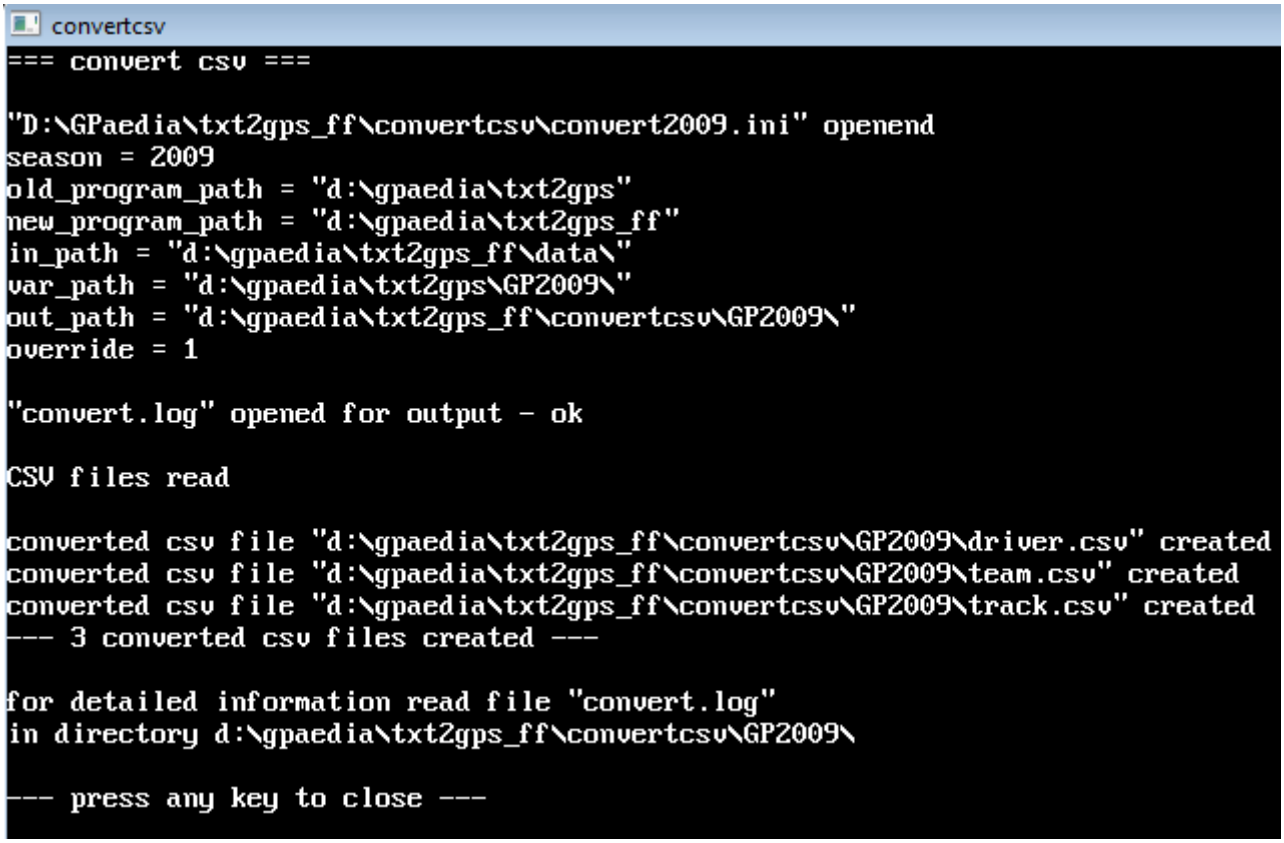
Convert old files

```
;--- convert2009.ini ---
;--- GPaediaMaker Tool "convercsv" configuration file ---

; define season variable %season%
season=2009

; program path of old txt2gps.exe
old_program_path = d:\gpaedia\txt2gps

; program path of new txt2gps_ff.exe
new_program_path = d:\gpaedia\txt2gps_ff

; input path for CSV format files
in_path = %new_program_path%\data\

; input path for CSV files to be converted
var_path = %old_program_path%\GP%season%\

; output path for new converted CSV files
out_path = %new_program_path%\convertcsv\GP%season%\

; override existing old CSV files (default=0: no override)
;override=1
```

The following example shows an INI file to convert old 2009 season CSV files into new free format.The ***season*** specification defines the variable %season%**.**. As with txt2gps_ff you may create a different INI file for every season (e.g. convert2012.ini, convert2013.ini) and drag and drop it unto convertcsv.exe to run the program with this settings.

- ***old_program_path*** and ***new_program_path*** are setting the variables %old_program_path% and %new_program_path%
- The ***in_path*** folder contains the CSV files with the standard free format that tell the program how to arrange the rows of the new CSV data files.
- The ***var_path*** folder contains the old CSV data files to be converted.
- The ***out_path*** folder defines the folder were the converted files are put in.
- The ***override*** switch is meant as a possible protection of existing CSV files.With default 0 old CSV files "driver.csv“, "team.csv“ and "track.csv“ are not overridden but new files "driver_new.csv“, "team_new.csv“ and "track_new.csv“ created instead (which will then be overridden with the next try). With override=1 you do without that safety measure.

Example 1 – season 2009 files converted



Create new season files without data

If you deactivate the ***var_path***, convertcsv does not look for old CSV data files but leaves the data cells empty instead. That way you can create a new CSV data file corresponding to the format CSV file found in the in_path folder.
Example to create new season 2012 files with standard format:

```
;--- newcsv2012.ini ---
;--- GPaediaMaker Tool "convercsv" configuration file ---

; define season variable %season%
season=2012

; program path of new txt2gps_ff.exe
new_program_path = d:\gpaedia\txt2gps_ff

; input path for CSV format files
in_path = %new_program_path%\data\

; output path for new CSV files
out_path = %new_program_path%\GP%season%\

; override existing old CSV files (default=0: no override)
;override=1
```

The following lines were omitted:

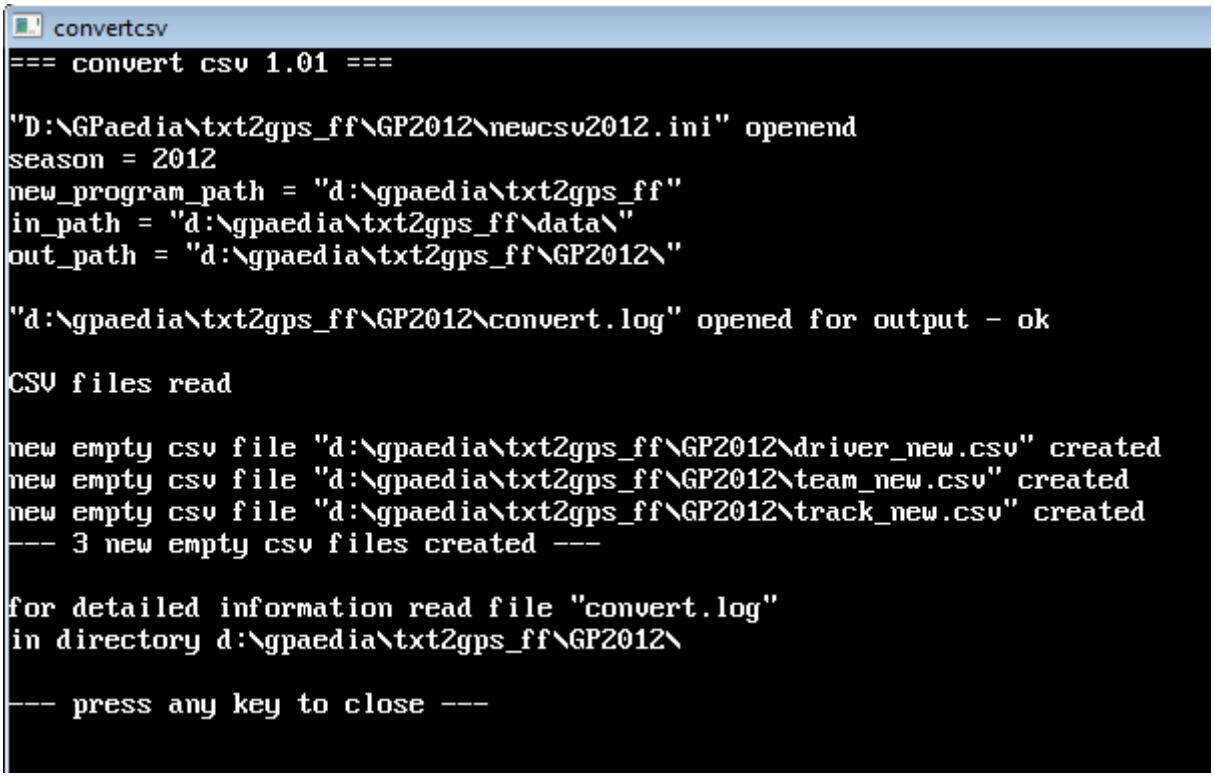
```
; program path of old txt2gps.exe
;old_program_path = d:\gpaedia\txt2gps
; input path for CSV files to be converted
;var_path = %old_program_path%\GP%season%\
```

If you have userformat CSV files in your GP%season% folder, you can specify the in_path differently as

```
in_path = %new_program_path%\GP%season%\
```

to get your CSV data files in userformat.

Season 2012 empy files



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	2012	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
2	Name																						
3	Team																						
4	Date of Birth																						
5	Nationality																						
6	Residence																						
7	Grand Prix Starts																						
8	Points Scored																						
9	Best Result																						
10	Best Qualifying																						
11	First GP																						
12	%lastyear% Championship Position																						
13	Previous Teams																						
14	driver replace																						
15	1																						
16	2																						
17	3																						
18	4																						
19	5																						
20	6																						
21	7																						
22	8																						
23	9																						
24	10																						
25	11																						
26	12																						
27	13																						
28	14																						
29	15																						
30	16																						
31	17																						

Empty season 2012 "driver_new.csv“ created by convertcsv:

The file "convert.log“ in the ***out_path*** directory specified in "convert.ini“ contains detailed information

about the conversion process.

If one of your old CSV data files is corrupt, conversion may still take place but the result will be faulty. If the error is not easy to recognize, examine the log file.

The log file contains two parts:

- **Reading log:** After reading a CSV file, the first two columns of the generated table are put into the log, so you can see wether everything is read the right way.
- **Creating log:** After creating a new table, the identifiers of the new and the old table are written side by side. The log for the track.csv includes both row indexes (beginning with row 0) and stops after the "corner data“ row.