GIR Titan-Hyperion

Vehicles/drivers import

www.gir.fr/en info@gir.fr

Version 1.0-14, march 2012

Table des matières

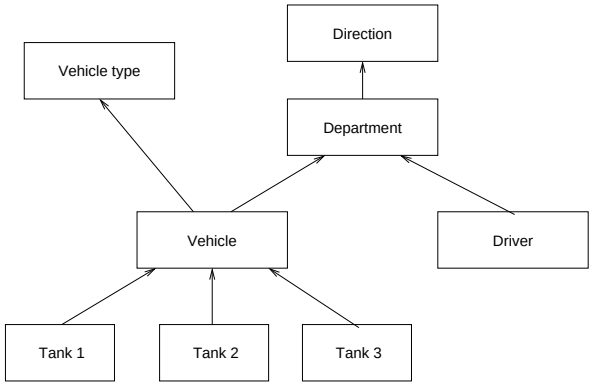
Chapitre 1

General points

1.1 Database model

Vehicles/drivers import synchronize the data of GIR Titan-Hyperion with the data from other applications. Usually, the two main tables which will be synchronized are vehicles and drivers.

In GIR Titan-Hyperion, the vehicles and drivers use the following database model :



- Vehicles are attached to a type of vehicle and a department, wich is attached to a direction. A vehicle is also linked to a list of tanks, which define the products authorized for this vehicle.
- Drivers are attached to a department.

1.2 Principle

Data synchronization with GIR Titan-Hyperion is performed by creating a file with a predefined name in a dedicated folder. GIR Titan-Hyperion reads regularly this folder, and when a new file is detected, it is imported.

Different import modes are available :

- Full import : the datafile contains a heading line and multiple lines of data. The import does full synchronization between the files and the database of GIR Titan-Hyperion :
  - Records present in the file and not in the database are created.
  - Records present in the file and in the database are updated.
  - Records present in the database but not in the file are marked as 'To delete' and forbidden.
- Incremental import : The datafile contains a heading line and one line of data. The record present in the file is created in the database, or updated if it already exists.
- Deletion : The datafile contains a heading line and one line of data. The record present in the file is marked as 'To delete' and forbidden in the database.

It is recommended to schedule a full import at least once a day (for example at nights), in order to guarantee an autorepairing of the system in case of failure or individual synchronization.

1.3 Creation of file

File creation is done in two steps :
- Creation of the datafile in the folder. `data\dti`. The file name have a strict syntax, defined later in this document.
- Creation of the confirmation file in the same directory. It is an empty file with the same base name of the datafile, apart from the extension which should be ".ok". This file should not be created until the datafile has been completely written in order to avoid the import of an incomplete file.

The name of the datafile must verify this syntax :
`hypimp-<format>-<mode>-<id>.txt`
where :
- `format`: name of the format. The name of each format is specified in its description, in the appendix of this documents. Examples : IGV3, IGD3, ...
- `mode`: import mode.
  - `all`: Full import
  - `inc-add`: Incremental import
  - `inc-del`: Deletion
- `id`: Identifier. It must be unique for every new file. Recommended identifiers :
  - For a full import, date of generation of the file with second precision using the format `YYYYMMDDHHNNSS`.
  - For the incremental import or deletion, `<date>-<key>` where `date` is the date of creation of the file using the format `YYYYMMDDHHNNSS` and `key` is the reference to the synchronized element. (ex : driver's identification number), wich can be found in the 'key' column of the file.

GIR Titan-Hyperion reads regularly the directory `data\dti`. When a file whose name verifies the syntax showed above is found, and is accompanied by

the corresponding confirmation file (".ok"), it will be imported. Once the import has finished, the given file and the confirmation file will be deleted.

1.3.1 Example

A synchronization from an application with a list of drivers and GIR Titan-Hyperion is required.

IGD3 format will be used to import the drivers.

The files to generate in `data\dti\` will be the following :

- Every night at 1am, generation of a file `hypimp-igd3-all-YYYYMMDDHHNNSS.txt`, and as confirmation file `hypimp-igd3-all-YYYYMMDDHHNNSS.ok`.
- In case of creation or modification of a driver in the client's application, generation of a file `hypimp-igd3-inc-add-YYYYMMDDHHNNSS-<key>.txt`, and its confirmation file `hypimp-igd3-inc-add-YYYYMMDDHHNNSS-<key>.ok`, where `key` is the reference key of the driver.
- In case of a driver deletion in the client's application, generation of a file `hypimp-igd3-inc-del-YYYYMMDDHHNNSS-<key>.txt`, and its confirmation file `hypimp-igd3-inc-del-YYYYMMDDHHNNSS-<key>.ok`.

Chapitre 2

Import

2.1 Import files

The import files are text files in CSV format. Default separators are the semicolon (';', ascii 0x3B) as field separator and double quotes ('"', ascii 0x22) as text separator.

In general, a CSV file corresponds to all or part of a table in GIR Titan-Hyperion. The expected data during the reading of the CSV file is different depending on the position in the file. If it is the first line, the data expected is a heading defining the content of the rest of the file. Following lines contain the data to import.

2.1.1 Heading line

First line of the file defines the relationship between the CSV fields and the fields of the data of GIR Titan-Hyperion. The format of each CSV field on the first line is :

`<name>%<options>%<comments>`

name : Name identifying the field in an unique way.

options : Options affecting the treatment which will be applied on the field

comment : String of any character, ignored during the import.

The character '%' will be used as delimiter between the different data in a field.

The presence of the options and comment is optional. In case of abscense of separator, the whole CSV field is interpreted as name of field.

2.1.2 Following lines

The line two and followings define the data. Their format depends on the type of the fields. The accepted formats are text, numerical and date-hour.

text : String of characters.

numerical : Group of characters between 0 and 9, preceded by le sign '-' if neccesary. All numeric fields are reduced to integers. Decimal format does not exist (par example, a volume is expressed in centilitres, 12.5 litres are written as '1250').

date : `dd/mm/yyyy`.

date-hour : `dd/mm/yyyy␣hh:nn` or `dd/mm/yyyy␣hh:nn:ss`. The year and hour are separated by an space.

For each format, valid field names as well as the datatypes and options associated are presented in the appendix of this document.

2.2 Import types

Synchronized imports perform a synchronization between the data defined in the CSV file and the database of GIR Titan-Hyperion. CSV file must contain the list of vehicles or the definition of the drivers. GIR Titan-Hyperion determines from this list which operations must be done in order the database to be synchronized with that file :

Existing records are updated, new records are created, and records that are not referenced anymore are marked 'to delete'

2.2.1 Matching

Synchronized imports need to do a matching between one line of the data in the CSV file and a record in the database of GIR Titan-Hyperion. The field used to do a matching always has the name 'key'. In order the synchronization to work correctly, this key should uniquely identify a record in the table.

It is possible to use different fields of the vehicles and drivers tables as well as the key.

- Vehicles : Registration, Code or Ref.
- Drivers : Name, Code or Ref.

The matching to be used is specified in the options of the 'key' field. The options indicating a matching method are accumulating, their order determines in wich order the matching will be attempted.

- `key%N`: Matching by name (Registration of the vehicles).
- `key%R`: Matching by ref
- `key%RN`: Matching by ref, then by name

When no matching method is defined, approximation by name ('N') is used by default.

In a similar way; it is necessary to use a matching when references to depencences of vehicle or driver are done. Examples of those dependences :

- For the vehicles : Types of Vehicles, Departments and Directions. Badges types, ...
- For the drivers : Departments, Directions and Badges types, ...

As for the 'key' field, the method of matching is specified in the options for the field for which the matching should be done. The different matchings accepted for each field are detailed in the import formats as appendix in this document.

The different matching methods are :

- by Name ('N') : uses the fields Abbreviated name and Complete name in the referenced table. Search is first done by the abbreviated name, then by the complete name if the first search was not succesfully completed;

- by Ref. ('R') : uses the field Ref. in the referenced table.
- by Code ('C') : uses the field Code in the referenced table.
- by N° of registration ('E') : uses the internal record number of GIR Titan-Hyperion. Numbering starts from Zero.

The three first methods create a new record if no matching was found (Except for the configuration tables, see below).

On the other hand, matching by registration number requires the element to be already defined in the database.

In case the string of data defined in a matching field is empty, two meanings are possible :

- If the field is optional in the database of GIR Titan-Hyperion (ex : Type de badge), an empty string means *(N/A)*.
- If the field is compulsory (ex : Department, Type of vehicle), the empty string leaves existing records unmodified and uses a value by default (see below) for new records.

Some elements cannot be automatically created during the import : These elements are the products and the badges types. This is the configuration information that must be created manually. If a type of badge or product not known is referenced, it will be ignored with a warning.

Examples :

- `service%NR`: Search a department by name, then by reference.
- `idbadge%C`: Search a type of badge by code.

The following file (format IGV3) uses the Ref. field to do the matching, defines the registration fields, type and department, and leaves the other fields unchanged (or empty if it is a new vehicle). Vehicle types are searched by name, departments are searched by Ref.

```
key%R;     immat;   type%N; service%R;
 A120;  1234AB69;  voiture;       S01;
 A121;  7451AB69;   camion;       S02;
 A122;  9834AB69;  voiture;       S02;
```

For the definition of tanks (format IGV3) or the local zones (formats IGV3-IGD3), the matching method is slightly different. To appoint, in this case, the value to be used for the matching should be specified on the first line of the CSV file, with the 'V' option. In the case of the tanks, as well as a column defining the tanks of Diesel, another column defining the tanks of Super will be presented...

Examples :

- `tk1%NV'GOI'`: Do a matching by name on the product GOI.
- `tk1%RNV'GOI'`: Do a matching by reference, then by name on the product GOI.
- `lz1%NV'Lyon'`: Do a matching on the Lyon area by name.

2.2.2 Synchronized import

The synchronized import performs a synchronization between the data in the CSV file and the information in the database. It may creates new data, not only in the vehicules and drivers tables, but also in their dependences (type of vehicles, departments, tanks, local zones...).

The rest of this section details the functioning of synchronization :

Vehicles and drivers tables have, each of them, a field for options dedicated to the synchronization with a managment of vehicles fleet. These fields contains two options : "Synchronization" and "To delete".

For each line of the CSV file, the matching field ('key') is read, and one element which this matching is searched. Multiple cases are possible :

- The element hasn't been found : It is created with the option "Synchronized".
- The element has been found and has the option "Synchronized", it is updated.
- The element has been found but it does not have the "Synchronized" option, it is ignored with a warning.
- The element has been found multiple times, it is ignored with a warning.

In the case of a full import, the elements with the option "Synchronized" that have not been found during the import, are marked to be deleted and forbidden. Actual deletion of these elements is performed manually using a management operation (See 2.3, page 14).

For an incremental import, the elements that were not found stay unchanged.

In the case of a creation, all the fields given in the import file are used to fill up the fields of the new record. The vehicles and drivers tables have a certain number of compulsory fields. If some of this compulsory fields are not defined in the CSV file, a default value will be used. The table below sums up the default values of the different compulsory fields of the vehicles and drivers tables.

Table	Field	Default value
Vehicles	Registration	Same value that the one used for the matching (code ou ref.)
Vehicles	Type	Type 'Imported'
Vehicles	Department	Department and/or direction 'Imported' in accordance with the defined fields
Drivers	Name	Same value that the one used for the matching (code ou ref.)
Drivers	Department	Department and/or direction 'Imported' in accordance with the defined fields

In the case of an update, for all the fields defined in the import file, the supplied value is compared with the existing value in the database. If they are different, the value in the database is updated. The fields that are not defined in the import file are ignored. The update can lead to the creation or deletion of dependencies.

– Creation of directions and departements.
– Creation of vehicle types.

If a department or a type of vehicle is not referenced, it remains in the database and should be deleted manually.

Sub-tables

During an import, the following sub-tables may be modified.

Tanks (Only vehicles) :

The matching is performed by product. If a product is not refered in the import file, the matching elements will remain unchanged. If it is refered, the matching elements will be added, deleted or updated depending on the value of the field in the CSV file.

Example : The vehicle '1234ZZ69' has a Gasoil tank (GOI), without meter prompt. The following examples show what modifications will be done on the tanks of the vehicle according to the content of the import file.

Unchanged tanks (no columns with V'GOI') :
`key%N; code;`
`1234ZZ69; 1234;`

Prompt for meter added on the Gasoil tank :
`key%N; tk1%NV'GOI';`
`1234ZZ69; AK;`

Oil tank added :
`key%N; tk1%NV'GOI'; tk2%NV'HUI';`
`1234ZZ69; A; A;`

Gasoil tank deletion :
`key%N; tk1%NV'GOI';`
`1234ZZ69; ;`

Locales Zones (vehicles and drivers) :

The matching is performed by site. The principle is identical to the one used for the tanks.

Authorized Vehicles (only drivers) :

A driver can be authorized either for certain specific vehicles, or for a set of vehicles. Depending on the configuration, the set of vehicles can be a department, a direction or a group of departments.

If a set that does not match the configuration is specified in the import file (for example, if the import file defines authorizations per department while the software is configured for authorizations per management), the column is ignored.

If there is at least one column auth_veh, all the authorizations for an specific vehicle will be updated by the ones defined in the import file.

Also, there is at least one column auth_svc (respectively auth_dir or auth_grp depending on the configuration), all the authorizations per department (respectively per management or group of department) will be updated.

Examples :

Authorization for two specific vehicles and authorization for unchanged department
`key%R; auth_veh1%R; auth_veh2%R;`
`1234; 1111ZZ69; 1112ZZ69;`

Authorization for a department and no authorization for vehicle
`key%R; auth_svc1%R; auth_veh1%R;`
`1234; svc01; ;`

2.2.3 String of options

The Options and Zones fields in the vehicles and drivers tables are option-type fiels. They are displayed in GIR Titan-Hyperion under the form of a set of boxes to check. In the import files, these fields are defined by a string of options. It is a string made up of alphanumeric characters, where each letter or figure matchs an option in GIR Titan-Hyperion (ex : 'I' to forbid, 'A' to zone A ...). The string can contain also characters '+' and '-'.

A string of options is interpreted in the following way :

```
<initial options>+<options to add>-<options to delete>
```

The part of 'initial options' is used to affect the new records. If the string of options does not have the characters '+' or '-', the part 'initial options' is also used to overwrite the existing values.

If the characters '+' or '-' are presented. they determine respectively the options to add or remove in the existing records.

Examples :

`ABC`: Place the options A, B and C and delete the other options, at the same time, in the new and in the existing records.

`ABC+B`: Place the options A, B, and C in the new records, place the option B in the existing records and leave the other options unchanged.

`-C`: Do not place any option in the new records and delete the option C in the existing records.

`A+`: Place option A dans new recors and leave the existing records unchanged.

2.3 Purge

While the imports are synchronized, the elements that are not referenced anymore in the import file are marked 'to delete' (See 2.2.2, page 12) and forbidden, so they are prevented to take fuel.

Nevertheless, they remain defined in the database, which allows to keep trace of their transactions, or still manage them manually (deactivating the options "Synchronized" and "To delete").

Real purgue of these elements can be done manually, in ▷ Management, Park import. The system asks for a date and search all the elements that have been marked as 'to delete' before the introduced data. Then, the system do the purgue after a confirmation.

If there are still transactions referencing deleted elements, they will not be affected. The purgue of transactions is done using a different operation, depending on the data specification. (See user manual)

2.4 Example of import

Problem : We have a fleet of vehicles identified by a code, which is identified by a badge in the terminal. The badge is not known in the application of management of vehicles, and it will taken from GIR Titan-Hyperion. We want to use the code to identify the vehicles during a transaction in a station.

Possible solutions : The vehicles are identified by their code. This value will be used to do the matching ('key' field). It is also required to determine in which field this code will be imported. It is also used to be specified in the options of 'key' field.

Given the vehicles to be identified in the terminal by the code, the field Code will necessarily contain this value. This field will be the one to be used to perform the matching, which will allow to leave the Immatriculation and Ref. fields availables to fill in with other information. Hence, the option to be specified in 'key' field is 'C'.

There are two possible ways to add information :

- Defining a fix value (for example, Immatriculation 'field' of the vehicle with the code 1234 will be always '1234AB69').
- Defining a default value for new vehicles, which will be allowed to be freely modified in GIR Titan-Hyperion.

This behavior is parameterized by 'I' option in the supported fields. In our case, we are going to use it to supply Code badge field a default value in order to prevent new created vehicles from being identified alone in the terminal with their code. We are going to specify a nonexistent badge (for example, 'FFFFFFFF'), wich will be manually remplaced by the GIR Titan-Hyperion administrator in order to allow a vehicle to identify itself per badge.

The registration, departments and products authorized are completely specified during the import. If these values are modified in GIR Titan-Hyperion, they will be resetted to the specified value at each import.

Other fields like the vehicle type are not specified in the file. They will be created with a default value ('Imported' in the case of type), and freely editable afterwards in GIR Titan-Hyperion.

The file will be the following format :

```
"key%C";    "regi"; "cbadge%I"; "service%N"; "zone"; "tk1%NV'GOI'"; "tk1vol"
 "1234"; "1234AB69"; "FFFFFFFF";     "Admin";    "A";          "AK";      50
 "1235"; "7531CD51"; "FFFFFFFF";      "Tech";    "A";          "AK";      50
 "1236"; "7532CD51"; "FFFFFFFF";      "Tech";    "A";          "AK";      50
 "1237"; "4084EJ75"; "FFFFFFFF";     "Admin";    "A";          "AK";      50
```

Annexe A

Import formats

This section describes the format of import files.

A.1 Format IGV3 - Vehicles

Field	Format	Options	Comment
key	Text	CNR	Compulsory field, used by the matching. C : Matching by code N : Matching by registration R : Matching by ref.
immat	Text	DI	Registration
ref	Text	DI	Ref.
nbadge	Text	DIP	Badge number
idbadge	Text	DICEN	Badge type C : Search by code or badge type E : Search by record number or badge type N : Search by name or badge type
cbadge	Text	DI	Badge code
code	Text	DI	Code
scode	Num.	DI	Secret code
type	Text	DIENR	Vehicle type E : Search by record number of vehicle type N : Search by the name of vehicle type R : Search by the ref. of vehicle type
type.nick	Text	DI	Short name of vehicle type (since version 1.4.50)
type.name	Text	DI	Full name of vehicle type
type.ref	Text	DI	Ref of vehicle type
model	Text	DIENR	Vehicle model (Same options as in the field type)
model.nick	Text	DI	Short Name of vehicle model
model.name	Text	DI	Full name of vehicle model
model.ref	Text	DI	Ref of vehicle model
division	Text	DIENR	Division (Same options as in the field type)
division.nick	Text	DI	Short name of division
division.name	Text	DI	Full name of division
division.ref	Text	DI	Ref of division
direction	Text	DIENR	Direction (same options as in the field type)
direction.nick	Text	DI	Short name of the direction
direction.name	Text	DI	Nom complet of the direction
direction.ref	Text	DI	Ref of the direction
service	Text	DIENR	Department (same options as in the field type)
service.nick	Text	DI	Short name of the department
service.name	Text	DI	Full name of the department
service.ref	Text	DI	Ref of the department

krange	Num.	DI	Range km (in km)
hrange	Num.	DI	Range horaire (in hours)
nhours	Num.	DI	Number of hours for capacity credit (since version 1.4.15)
expiry	Date	DI	Expiry
tipp_stat	Text	DI	TIPP status (since version 1.4.26). Possible values : Empty : Any P : Owner L : Tenant SL : Sub-tenant
tipp_expiry	Date	DI	TIPP expiry date (since version 1.4.26)
msg1	Text	D	First line of the message assigned to the vehicle. (since version 1.5.13)
msg2	Text	D	Second line of the message (ignored if msg1 is not defined)
kmeter	Num.	DF	Odometer (in km) (since version 1.5.13) If 'F' option is not present, the meter is directly modified. If not, it is imported by a fuel transaction with a null volume.
hmeter	Num.	DF	Hour meter (in hundredth of hour) (since version 1.5.13) The behaviour of the 'F' option is identical to the odometer.

options	Text	D	String of options among the characters : I : Forbidden B : Identification by badge C : Identification by code Y : Badge initialization Z : Code by filling-station attendant x : Generic criterion 1 y : Generic criterion 2 z : Generic criterion 3 A : Activity of Code S : Specific code 2 : 2nd identification (Driver Id.) P : Attendant M : Attendant Menu T : Ticket U : Mutualized X : Inhib. ext. vehicle R : Disable consumption computation. K : Forceable meter Q : Show consumption. L : Rental
zone	Text	D	String of options among [A-Z]
tk1	Text	DIENRV	Tank 1, String of options among : A : Authorized product K : Prompt for odometer H : Prompt for hour meter F : Fuel option C : Capacity credit (since version 1.4.15) S : Department credit (since version 1.5.14) Matching of the product : E : Matching by record number of product N : Matching by name of product R : Matching by reference of product V : Value used for the product matching. The following V character is used as delimiter : `V'GOI'` and `V#GOI#` are equivalent
tk1vol	Num.	DI	Capacity of tank 1. This field is ignored if the tank 1 is not defined.

tk1cons_kref	Num.	DI	Reference km consumption for tank 1 (in L/100km * 100)
tk1cons_kmin	Num.	DI	Min. threshold for the km consumption of tank 1 (in L/100km * 100)
tk1cons_kmax	Num.	DI	Max. threshold for the km consumption of tank 1 (in L/100km * 100)
tk1cons_href	Num.	DI	Reference hour consumption for tank 1 (in L/H * 100)
tk1cons_hmin	Num.	DI	Min. threshold for hour consumption of tank 1 (in L/H * 100)
tk1cons_hmax	Num.	DI	Max threshold for hour consumption of tank 1 (in L/H * 100)
...	...	...	...
tk16	Text	DIENRV	Tank 16. Same options as for the tank 1
...	...	...	...
lz1	Text	DIENRV	Local zone 1, string of characters among [A-Z] Matching for the site : E : Matching by record number of site N : Matching by name of site R : Matching by reference of site V : Value used for the matching of sites The following V character is used as delimiter : `V'Lyon'` et `V#Lyon#` are equivalent
...	...	...	...
lz99	Text	DIENRV	Local zone 99 (16 max. for versions below 1.4.15)
vc1	Text	DIENV	Characteristic 1 (since version 1.4.32) String of options among : A : Defined value Matching for the characteristic : E : Matching by record number of characteristic N : Matching by name of characteristic V : Value used for the matching of characteristics. The following V character is used as delimiter : `V'Colour'` et `V#Colour#` are equivalent

vc1value	Text	DI	Value of the characteristic 1. This field is ignored if the characteristic 1 is not defined. The type of this field can be text, numeric or date, depending on the type of the characteristic. The characteristic of boolean type are interpreted as numeric values, where 0 means “false” and all other value means “true”.
...	...	...	...
vc16	Text	DIENV	Characteristic 16 (since version 1.4.32). Same options as for the characteristic 1.
...	...	...	...

The 'D' option allows to deactivate a field. Every column will be ignored during the import.

The 'I' option allows to define an active field only if the record has been created by the import. If the record already exists, the value given in the file is ignored and the field remains unchanged in the database.

The 'P' option searchs the number of badge specified in the proxi files and updates the Code badge field if any correspondance is found. If a badge code is specified in the import file, it will be used when the badge number has not been found in the proxi files. If the badge code is not specified, it will be ignored.

The matching for departments is different depending on the presence or not of the direction field :

- If no direction is specified, search is done over all departments defined in the database. The new departments are created in the direction 'Imported'.
- If a direction is specified, search is done among all departments of the given direction. If the direction does not exist, it will be created.

Character 'B' in the 'option' field allows to force the identification type to 'Badge'. This option is only useful when the 'nbadge' and 'cbadge' fields are empty or not specified. By default, identification type (badge or code) is determined automatically according to all defined fields.

A.2 Format IGD3 - Drivers

Field	Format	Options	Comments
key	Text	CNR	Compulsory field, used by the matching. C : Matching by code N : Matching by name R : Matching by ref.
name	Text	DI	Surname
fstname	Text	DI	Name
nick	Text	DI	Initials
ref	Text	DI	Ref.
nbadge	Text	DIP	Badge number
idbadge	Text	DICEN	Badge type C : Matching by code or badge type E : Matching by the record number of the badge type N : Matching by the name of the badge type
cbadge	Text	DI	Badge code
code	Text	DI	Code
scode	Num.	DI	Secret Code
type	Text	DIENR	Driver type (from version 1.4.50) E : Matching by record number of the driver type N : Matching by name of driver type R : Matching by ref. of driver type
type.nick	Text	DI	Short name of driver type (from version 1.4.50)
type.name	Text	DI	Full name of driver type
type.ref	Text	DI	Ref of driver type
division	Text	DIENR	Division (Same options as in field)
division.nick	Text	DI	Short name of the division
division.name	Text	DI	Full name of the division
division.ref	Text	DI	Ref of the division
direction	Text	DIENR	Direction (Same options as in field)
direction.nick	Text	DI	Short name of the direction
direction.name	Text	DI	Full name of the direction
direction.ref	Text	DI	Ref of the direction
service	Text	DIENR	Department (Same options as in field)
service.nick	Text	DI	Short name of department
service.name	Text	DI	Full name of department
service.ref	Text	DI	Ref of department
expiry	Date	DI	Expiry

msg1	Text	D	First line of message assigned to the driver (since version 1.5.13)
msg2	Text	D	Second line of message (ignored if msg1 is not defined)
options	Text	D	String of options among the characters : I : Forbidden B : Identification by badge C : Identification by code Y : Badge initialization Z : Code by filling-station attendant x : Generic criterion 1 y : Generic criterion 2 z : Generic criterion 3 A : Activity of code S : Specific code 2 : 2nd identification (Id. vehicle) P : Attendant M : Attendant menu U : Allow on mutualized vehicles X : Vehicles ext. inhibit
zone	Text	D	String of options [A-Z]
tacho_dl	Num.		Frequency of unloading for tachograph cards, in days. (since version 1.4.10)
cpu_lang	Text		Language in the terminal. (since version 1.4.10) Languages availables : Empty : Language by default fr : French en : English es : Spanish nl : Dutch de : German it : Italian
auth_veh1	Text	DIENR	Authorized vehicle 1 E : Matching by vehicle record number N : Matching by registration R : Matching by reference of vehicle
...	...	...	...
auth_veh3	Text	DIENR	Authorized vehicle 3
auth_svc1	Text	DIENR	Authorized department 1 (same matching as in service field)
auth_svc2	Text	DIENR	Authorized department 2
auth_dir1	Text	DIENR	Authorized direction 1 (from version 1.2.71, same matching as in direction field)

auth_dir2	Text	DIENR	Authorized direction 2 (from version 1.2.71)
auth_grp1	Text	DIENRS	Authorized group 1 (from version 1.2.71) E : Matching by the record number of the group of departments N : Matching by name of the group of departments R : Matching by the ref. of the group of departments S : Matching by departments (from version 1.2.80). This option should be used together with E, N ou R. Ex : "SR" : Matching by ref. of department.
auth_grp2	Text	DIENRS	Authorized group 2 (from version 1.2.71)
lz1	Text	DIENRV	Local zone 1, string of characters [A-Z] Matching by site : E : Matching by record number of the site N : Matching by the name of the site R : Matching by the reference of the site V : Value to be used for the matching of the sites. The following V character is used as delimiter : `V'Lyon'` et `V#Lyon#` are equivalent
...	...	...	...
lz99	Text	DIENRV	Local zone 99 (16 max. in versions under 1.4.15)
...	...	...	...

See the format IGV3 for the dependencies between the department fields and the direction, and the meaning of the options 'D', 'I' and 'P'.