GIR Titan-Hyperion
User manual

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

Version 1.2-2, may 2006

2

Contents

Introduction

Overview

GIR Titan-Hyperion is a fleet management software. It covers fuel delivery, assignment of vehicles to drivers, vehicles maintenances, access to the station or to buildings, vehicles keys delivery.

Part I

Instructions for use

Chapter 1

Data entry

1.1 Vehicles

▷ Vehicles

1.1.1 General information

Num.: number automatically assigned to each vehicle by GIR Titan-Hyperion. This number is unique to each vehicle. When a ticket is printed after a fueling, this number may be used to identify the vehicle (See 2.7, page 50).

Registration: registration of the vehicle. It is used to refer to a vehicle in the software, and must be unique to each vehicle.

Ref.: Value used to refer to a vehicle in specific exports.

1.1.2 Identification

Fields: Badge num., Badge type, Badge code, Code, Identification
A vehicle may be identified on the CPU by three different ways:

- Identification by code:

Identification	*Code*
Badge num.	Optional
Badge type	*(N/A)*
Badge code	Empty
Code	Filled in

- Identification by a GIR proximity badge:

Identification	*Badge*
Badge num.	Number written on the proximity badge
Badge type	*(N/A)*
Badge code	Chip code of the proximity badge
Code	Advised

- Identification by a specific badge:

Identification	Badge
Badge num.	Optional
Badge type	Specific badge type
Badge code	Chip code of the specific badge
Code	Advised

When identifying by badge, entering a code is not neccessary, nonetheless, it is advised. If no code is defined, the vehicle will only be identified by its badge. If a code comes along with a badge, it may be used to identify the vehicle through the help of the pump attendant (See 2.3, page 45). This can be useful if someone has lost or forgotten his badge. The code is also used to identify a vehicle in the printed transactions from the live printer.

Each vehicle should be identified with a unique code, hence it is forbidden to define several vehicles with the same code. A code can be numerical or alphanumerical.

1.1.3   Entry of a GIR proximity badge
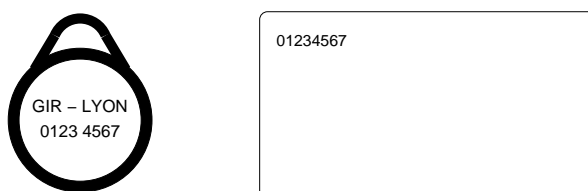
The above picture shows two examples of GIR proximity badges which number is 01234567. Several methods are available to facilitate the entry of such badges:

Via a proximity file

The initialization of GIR proximity badges may automatically occur from proximity files, supplied during the installation. Two methods exist:

- Enter the number written on the badge in the Badge num. field while creating or modifying a vehicle. After the validation, GIR Titan-Hyperion automatically looks for an associated chip code. If such code is encountered, a message allowing to automatically update the Badge code field appears. A click on Yes validates the update.

  This procedure can be made totally transparent by enabling the *Strict proxi file* and *Hide id+badge code* options in the Manager config..

- Click on Read proxi. file in the General tab. Then, enter the number which is written and validate.

For GIR proximity badges delivered since 2005, the number to enter in the Badge num. field is composed of the 8 numbers written on the badge and begins with 2 or 3 (ex: 23450000).

It is possible to consult the available proximity files as well as their content in ▷ Config., Verifications, Check proxi. file. These are text files containing on each line a badge number and the associated chip code.

They must be placed in the `data\dti` directory.

On the CPU

The initialization of GIR proximity badges may be directly realized on the CPU:

- during the first fueling of the vehicle.

- via the attendant menu.

The first method allows to initialize badges gradually, on their first use, whereas the second initializes multiple badges at the same time. In both cases, the vehicle must have the following parameters:

Badge num.	Optional
Badge type	*(N/A)*
Badge code	Empty
Code	Filled in
Identification	*Proxi. badge init.*

The vehicle Badge code field in GIR Titan-Hyperion will automatically be assigned during the first transactions retrieval which will follow the initialization. This can involve difficulties when several CPUs are being used at the same time (See 3.1.1, page 51).

Initialization at the first fueling

1. `ID VEHICLE`: Enter the vehicle code.

2. `Badge initialization`: Present your badge.

3. Continue the fueling normally (See 2.2, page 42).

Initialization by the attendant menu

1. Access to the attendant menu (See 2.4, page 46)

2. Select the sub-menu badges (key G or 6)

3. `Vehicle`: Enter the vehicle code. U (or 2) allows to initialize a driver badge. V (or 1) returns to a vehicle badge.

4. `Badge initialization`: Present your badge. The possible messages are:

    - `BADGE PROGRAMMED`: The badge of this vehicle has already been initialized.

    - `INEXISTENT CODE`: The code entered does not correspond to any vehicle.

    - `BADGE ALR. ASSIGNED`: The badge has already been assigned to another vehicle.

- **BADGE ACCEPTED**: The badge has been correctly initialized.

It is then possible to enter a new vehicle code, to intialize another badge.

1.1.4   Meters

Each vehicle possesses a milometer and an hour meter. It is not possible to directly modify the value of a meter, a fuel transaction must be carried out.

There is a range of tolerance for each meter, which allows to check the value entered by someone while fueling (For more details, see 4.1, page 59).

The meters and ranges defined for a vehicle are displayed under the General tab.

1.1.5   Options

- *Forbidden*: The vehicle cannot refuel.

- *Secret code*: The entry of a secret code is required after the identification. This code is defined in the Code field. This option is not available if *Code + Secret code vehicles* is enabled in Installer config.. See below for more details about this option.

- *Id. driver*: A driver identification is required after having identified the vehicle.

- *Attendant*: The vehicle is an attendant (See 2.3, page 45).

- *Attendant menu*: The vehicle can access the attendant menu (See 2.4, page 46).

- *Activity code*: An activity code is asked after the identification (See 1.4, page 23).

- *Specific code*: A specific code is asked after the identification (See 1.5, page 23).

- *Ticket*: Activates the printing of a ticket at the end of this vehicle's transactions (only with a ticket printer).

- *Disable cons. comput.*: Disables covered miles and consumption computing for this vehicle.

- *Mutualized*: When vehicles authorization by driver is enabled (See 1.9, page 31), this option sets the vehicle as "Mutualized". This makes the vehicle available to all drivers with the *Auth. on mutu.* option, in addition to standard authorizations.

- *Inhib. ext. vehic.*: When the *Vehicle ext.* feature is enabled in the Manager config., this option cancels it for this vehicle.

Option *Code + Secret code*

By default, vehicles only have one Code field, which is used to identify the vehicle (directly or via the attendant), and to serve as a secret code if the Secret code option described above is activated.

Should we wish that the vehicles could be identified by code, then confirm their identification with an additional secret code, the *Code + Secret code vehicles* option in the Installer config. table should be activated. This option can only be activated by GIR Titan-Hyperion installer.

When this option is enabled, a new Secret code field appears in the vehicle modification form, and the existing Code field length is reduced by 2. The Secret code is numerical and composed of 4 figures. The zero value means "no secret code".

The following table sums up the different available identification modes, as well as the field which will be used as a secret code. The identification method of a vehicle may be consulted in the vehicle sheet General tab.

Identification	Without the C+Sc option	With the C+Sc option
Badge	Yes	Yes
Code	Yes (8 characters)	Yes (6 characters)
Badge + Secret code	Yes (Code field)	Yes (Secret code field)
Code + Secret code	No	Yes (Secret code field)

1.1.6 Messages

Only one message can appear on the terminal for each vehicle. If a vehicle has several messages, the one with top priority will appear, following this priority:

1. Maintenance message

2. Message attached to the vehicle (Message field)

3. Message attached to the vehicle's department (only if the vehicle is authorized on the CPU)

The attachment of a message to a vehicle is done by clicking on Modif. in the vehicle sheet General tab.

1.1.7 Authorization

In addition to the *Forbidden* option, two criteria determine a vehicle access rights.

Zone defines the zones in which the vehicle is authorized (See 1.6, page 24).

Expiry allows to define a time limited authorization: if a vehicle's expiry is set to *31/12/2004*, it won't be able to use the CPU from January 1 2005 at midnight.

The vehicle sheet General tab allows to quickly visualize the access rights:

Green square: The vehicle is authorized, a summary of the authorized zones and sets is displayed.

Red square: The vehicle is forbidden, and the reason is displayed. A vehicle
may be forbidden for the following reasons:

- *Forbidden*: the *Forbidden* option is enabled.
- *Expired*: the deadline has expired.
- *Blocked: maintenance not done*: one of the vehicle's maintenance has
  reached the blocking limit (See 9, page 83).
- *No zone*: the vehicle is not authorized on any zone.
- *Insufficient credit*: the vehicle no longer has any credit.

1.1.8 Tanks

Generalities

Tanks specify which products a vehicle is entitled for. They may be consulted
via the vehicle sheet's Tanks tab. The Capacity field specifies the maximum fuel
quantity that can be taken during a fueling. If the Capacity value is *(N/A)*, this
quantity is not limited.

A tank Options specify which meters are asked at the CPU when a vehicle
is fueling: *Meter (miles)* to enter a milometer, *Meter (hours)* to enter an hour
meter.

The *Fuel* option is only useful for vehicles with several fuel products (Super,
GPL, . . .). In this case, we would tick the fuel option for fuel products, and
would keep it disabled for other products (oil, windscreen, . . .). The *Fuel* option
is useful when computing the consumption and the covered miles during the
processing of a fuel transaction).

The Initial credit, Credit, Period, and Credit options fields will be explained in
the credits chapter (10, page 89). If credits are not managed, leave these fields
with *(N/A)*.

Consumption management

Consumption criteria can be defined:

- Specifically for each vehicle, under the Tanks tab in the vehicle sheet.

- For each vehicle model, in ▷ Misc, Vehicles models, when this feature has
  been enabled in the Manager config..

The Cons. (miles) and Cons. (hours) fields represent the reference consump-
tions for a given product. They are used to determine if a vehicle over or
underconsumes. If those values are *(N/A)*, there won't be any over or under-
consumption indicator. The tolerance limit for the reference consumptions is
defined in the Manager config. (See 1.9, page 31).

It is also possible to define customized values for these limits, via the Min.
cons. (miles) and Max. cons. (miles) fields (repectively Min. cons. (hours) and
Max. cons. (hours) for hour meters). If those two fields are *(N/A)*, the limit
defined in the Manager config. is used. If at least one of those two is defined,
customized values are used. Leaving one of those two limits to *(N/A)* allows to
disable the over or underconsumptions for a given vehicle.

Tolerance ranges are displayed in the tanks list.

The table below emphasizes the different possibilities. "Reference", "Min." and "Max." respectively refer to the reference consumption and the customized limit values, defined in the vehicle's tank. "Undercons. tol." and "Overcons. tol." refer to the limits defined in the Manager config.:

Reference	Min.	Max.	Undercons. tol.	Overcons. tol.	Range
(N/A)	ignored	ignored	ignored	ignored	*(N/A)*
10.00	*(N/A)*	*(N/A)*	30.00 %	20.00 %	7.00 - 12.00
10.00	*(N/A)*	*(N/A)*	*(N/A)*	25.00 %	*(N/A)* - 12.50
10.00	5.00	14.00	ignored	ignored	5.00 - 14.00
10.00	*(N/A)*	14.00	ignored	ignored	*(N/A)* - 14.00

1.1.9   Other information

The Maintenances and Last maintenances tabs concern maintenances and will be detailed in the corresponding chapter (Chapter 9, page 83).

The Last transactions tab allows to access the last 10 transactions of the vehicle.

If vehicles authorization by driver is enabled (See 1.9, page 31), the Auth. drivers tab shows the drivers who are entitled to use this vehicle.

1.1.10   Scrapping a vehicle

This paragraph describes how to scrap a vehicle in GIR Titan-Hyperion.

1. Edit the vehicle by clicking on Modif.

2. In the Badge num. field, add a '-' just before the number.

3. In the Badge code and Code fields, delete the current value and leave the fields empty.

4. Tick the *Forbidden* option.

5. Save the changes by clicking on Validate

6. Go to the Maintenances tab and tick the *Disabled* option for each maintenance of the vehicle.

The scrapped vehicle is no longer recognized on the CPU and its badge may be assigned to another vehicle.

This method allows to not erase the scrapped vehicle, in order to keep a trace of its transactions. If we sum up a period with both the transactions of the scrapped vehicle and the transactions of the new vehicle, both vehicles will appear on separate lines.

1.2   Drivers

▷ Drivers

1.2.1   General information

Num.: number automatically assigned to each driver by GIR Titan-Hyperion. This number is unique to each driver. When a ticket is printed after a fueling, this number may be used to identify the driver (See 2.7, page 50).

Name, Firstname: Name and firstname of a driver. It is used to refer to a driver in the software.

Ref: Value used to refer to a driver in specific exports.

1.2.2   Identification

Fields: Badge num., Badge type, Badge code, Code, Identification

The definition of identification settings for a driver is the same as the one described for vehicles (See 1.1.2, page 13).

1.2.3   Options

- *Forbidden*: The driver cannot refuel.

- *Secret code*: The entry of a secret code is required after the identification. This code is defined in the Code field. This option is not available if *Code + Secret code drivers* is enabled in Installer config.. See 1.1.5, page 16 for more details about this option.

- *Id. vehicle*: A vehicle identification is required after the driver identification.

- *Attendant*: The driver is an attendant (See 2.3, page 45).

- *Attendant menu*: The driver can access the attendant menu (See 2.4, page 46).

- *Activity code*: An activity code is asked after the identification (See 1.4, page 23).

- *Specific code*: A specific code is asked after the identification (See 1.5, page 23).

- *Auth. on mutu.*: When vehicles authorization by driver is enabled (See 1.9, page 31), this option allows the driver to use "mutualized" vehicles.

- *Inhib. ext. vehic.*: When the *Vehicle ext.* feature is enabled in the Manager config., this option cancels it for this driver.

1.2.4   Messages

Only one message can appear on the CPU for each driver. If a driver has several messages, the one with top priority will appear, following this priority:

1. Message attached to the driver (Message field)

2. Message attached to the driver's department (only if the driver is authorized on the CPU).

To attach a message to a driver, click on Mod. in the user sheet's General tab.

1.2.5 Authorization

In addition to the *Forbidden* option, two criteria determine a user access rights.

Zone defines zones in which the user is authorized (See 1.6, page 24).

Expiry allows to define a time limited authorization: if a driver's expiry is set to *31/12/2004*, it won't be able to use the CPU from January 1 2005 at midnight.

The driver sheet General tab allows to quickly visualize the access rights:

Green square: The driver is authorized, a summary of the authorized zones and sets is displayed.

Red square: The driver is forbidden, and the reason is displayed. A driver may be forbidden for the following reasons:

- *Forbidden*: the *Forbidden* option is enabled.
- *Expired*: the deadline has expired.
- *No zone*: the driver is not authorized on any zone.

1.2.6 Other information

If vehicles authorization by driver is enabled (See 1.9, page 31), the Auth. vehicles tab allows to define which vehicles the driver is authorized to use.

By default, a driver is entitled to use all vehicles. Authorizations can be defined:

- By department, direction or department group, depending on the configuration.

- On particular vehicles.

The Last transactions tab allows to access the last 10 transactions of the driver.

1.3 Assignment of vehicles and drivers

1.3.1 Directions and departments

▷ Misc, Directions
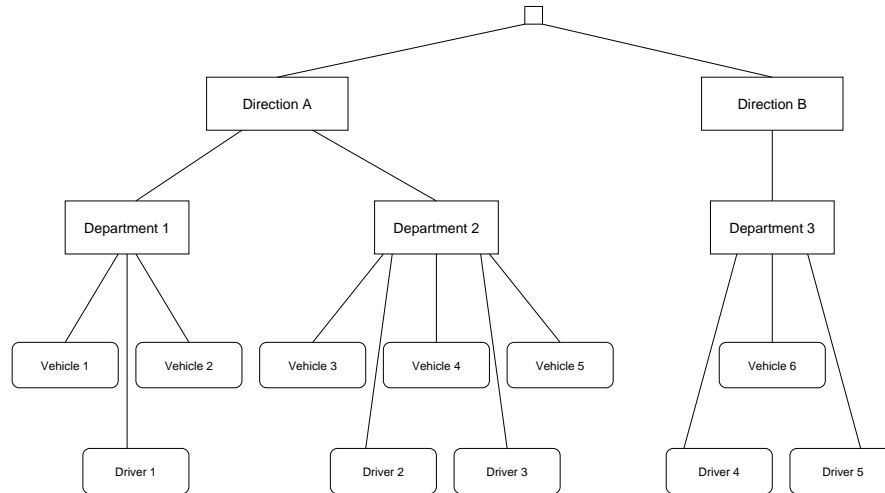
The field Department in vehicles and drivers refers to the department to which they are assigned. Each department is itself connected to a direction. The connection by direction and department is used to make sums by Direction/Department, Department/Vehicle and Department/Driver (See 5.1.2, page 64).

It is possible assign a given message to all the vehicles or drivers of a department, by selecting it in the Message to drivers or Message to vehicles fields of the desired department. This message may not appear on the CPU if there is a message with a superior priority (See 1.1.6, page 17 and 1.2.4, page 20).

Assigning vehicles and users to departments may be used to define credits by department (See 11, page 99)

Departments may also be used to define department-wide credits (See 10, page 89).

Transactions closure

Transaction closure can be enabled by defining a Closure day in ▷ Misc, Manager config., then ticking the *Closure* option in the desired directions.

Once this configuration has been done, the ▷ Management, Closures menu allows to close completed months for all departments in the selected directions. When a period is closed on a given department, it becomes impossible to modify or to create new transactions in this department during this period.

Transaction closure is available to standard users who have the closure right enabled (see 1.8, page 27, advanced mode). The main manager can also reopen a period that has been closed.

When a period is not closed and that the closure day has been reached, an alert is generated. Example: the closure day is set to 10. From the first to the tenth day of the month, a "closure to do" message is displayed on the dashboard, as long a the closure is not done. If it is still not done on the 10th, an alert is generated and the dashboard message is updated to "closure not done".

Note: a period can only be closed when all transactions have been retrieved. Thus, closing April will only be possible if a transaction retrieval has been done after May 1st at midnight.

1.3.2   Types

▷ Misc, Vehicles types ▷ Misc, Drivers types
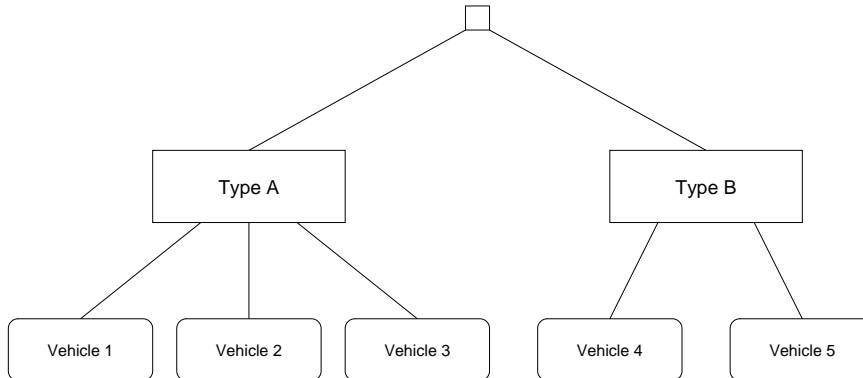
The Type field in a vehicle refers to the assignment type of this vehicle. It is used to make sums by Type and Type/Vehicle (See 5.1.2, page 64).

An assignment type is also available for drivers.

1.4   Activities

▷ Misc, Activities

Code: Activity code entered on the CPU.

Short name, Full name: Name of the activity, used to refer to an activity in the software.

Ref.: Value used to refer to an activity in specific exports.

Options:

- *Forbidden*: Disables the entry of this activity on the CPU.
- *Add-on*: Sets the *Add-on* option on the transactions for which this activity is declared.

The activities allow to make connections between fuel transactions and may be used for the credit management (See 10, page 89).

1.5   Specific codes

▷ Misc, Specific codes

Code: Specific code entered on the CPU.

Name: Name of the specific code, used to refer to a specific code in the software.

The specific codes table is automatically filled in by GIR Titan-Hyperion when the *Group spec. codes* option is enabled in the Manager config. (See 1.9, page 31). Grouping specific codes allows to make connections between transactions, particularly in sums (See 5.1.2, page 64).

1.6   Zones and time slots

▷ Misc, Zones

1.6.1   Overview

GIR Titan-Hyperion allows to define up to 26 zones, where each one is coded by a letter (from A to Z).

Zone: Zone code

Name: Zone name, used with the code to refer to a zone in the software

Options:

- *Watch*: This option allows to be notified when a driver is making a transaction in the zone. The notification must have been configured in e-mails settings.

Each zone is associated to a set of time slots which defines when the zone is active. The time slots fields are:

Begin hour: Hour when the validity begins

End hour: Hour when the validity ends (*24:00* means "Valid until the end of the day")

Days: Days of validity

Validity calendar: Allows to limit the time slot validity to the specified days.

Exception calendar: Allows to exclude the specified days from the time slot.

Options:

- *Vigilance*: When this option is enabled in a time slot, all the transactions made in this slot have the *Vigilance* option. The transactions containing this option will be included in the noteworthy transactions list (See 5.1.1, page 63), which allows to easily obtain the list of the transactions made in a given period, by night for instance.

Calendars are defined in the ▷ Misc, Calendars menu. They are made of a days list (if only the Date field is filled), or of a periods list (if both the Date and Final date fields are filled).

On TIP CPUs, the calendar is limited to the current week: it is then necessary to make a synchronization between the PC and the CPU at least every week if calendars are defined.

The Calendar tab in a zone sheet displays a summary of the authorizations. When only one month appears, a click on the day allows to have the detail of the time slots concerning this day. A click on the year shows a summary of all the months of this year.

Examples:

- To define a zone which is valid at any time, we would create the following time slot:

Begin hour	End hour	Days
00:00	*24:00*	*Mon, Tue, Wed, Thu, Fri, Sat, Sun*

→ **Conventions:**

- Name of a zone which is valid at any time: `All 24/24`.

• To define a zone which is valid from 6a.m. to 8p.m. during the working days, and from 6a.m. to 12a.m. on Saturday, we would create two time slots:

Begin hour	End hour	Days
06:00	*20:00*	*Mon, Tue, Wed, Thu, Fri*
06:00	*12:00*	*Sat*

• To exclude public holidays from this zone, we would define a calendar called `Public holidays`, containing the days to exclude, and select it as an Exception calendar for both time slots of this zone.

• To create a zone in which access is limited to the month of January 2005 (for instance for a temporary visitor), we would define a calendar with a `01/01/2005 - 31/01/2005` period, and select it as a Validity calendar in the time slots that are to be limited.

A vehicle or driver is authorized on a set (Fuel or Access) at the T time when the following condition is true:

There is at least one active zone at T, which is present both in the vehicle (resp. driver) and in the fuel set (resp. access set).

Access and fuel sets zones are editable in the ▷ Config., CPU, tanks tresholds, ref. and zones management menu (See 1.11, page 37).

A global summary of the different zones may be consulted in the ▷ Misc, Zones report menu. Moreover, zones for which a user is authorized are displayed in his sheet.

Configuration example:

We consider a petrol station defined with a fuel set. We want some of the vehicles to be able to refuel only during working days, and the other part to refuel at any time.

1. In ▷ Misc, Zones, we define the two following zones:

Zone code	Zone name	Time slots
A	All 24/24	00:00 à 24:00, every day
B	Week	06:00 à 20:00, from Monday to Friday

2. In the ▷ Config., then CPU, tanks tresholds, ref. and zones management, then Fuel sets, we tick *A* and *B* zones in the Zone field of the appropriate fuel set.

3. In ▷ Vehicles, we tick the *A* zone for all vehicles that should be able to refuel at any time, and the *B* zone for all vehicles that should only refuel during working days.

   Note: In this case, ticking both *A* and *B* for a vehicle is strictly equivalent to tick only *A*, since the *A* zone contains the *B* zone.

We consider a two access sets, the *Barrier* set which controls the main entry, and the *Washing* set which controls access to the washing station area. We want to limit the access to the station to some vehicles.

1. In ▷ Misc, Zones, we define both following zones:

Zone code	Zone name	Time slots
A	Barrier	00:00 à 24:00, every day
B	Washing	00:00 à 24:00, every day

2. In the menu ▷ Config., then CPU, tanks tresholds, ref. and zones management, then Access set, we tick the *A* zone in the Zone field of the access set *Barrier*, and the *B* zone in the Zone field of the access set *Washing*.

3. In ▷ Vehicles, we tick the *A* zone for all the vehicles, and the *B* zone for those we wish to authorize on the washing area.

1.6.2   Advanced management

For a large scale zones management in a multi-sites context, GIR Titan-Hyperion allows to define zones that are local to a site. Local zones are enabled by GIR Titan-Hyperion installer in the Installer config.. The authorization condition for a vehicle or a driver to a fuel or access set is the same as the one described above: an authorization is valid when an active zone is present both in the user and in the set.

The assignment of zones to drivers can be limited to the managers of the site (See 1.8, page 27).

1.7   Automatic programmings

▷ Misc, Auto. prog.

Automatic programmings allow to plan operations so that they could be automatically runned by GIR Titan-Hyperion at a given day and hour.

Name: Programming name, use to refer to a programming in the software.

Operation:

- *Retrieve.-Send* ,*Retrieve.-Send-Gauge*,. . . : Performs a synchronization dialogue with the different programmed CPUs (See 3, page 51).

- *Backup*: Performs a backup of the database in a `TWB` file.

- *Real-time suspension*: Disables all the real time communications with CPUs during the time range specified by the beginning and ending hours.

Begin hour: Hour when the operation will be launched.

End hour: For the *Retr./Send* operation, it keeps the connection on until the indicated hour. For the *Real-time suspension* operation, it indicates when communications will start again.

Days: Days when the operation should be launched.

Options:

- *Disabled*: Allows to temporarily disable an automatic operation execution.

As for *Retr.-Send* automatic operations type, it is necessary to define the CPUs with which the dialogue must be carried out. This is done in the Programmed CPUs subtable:

CPU: CPU with which the dialogue must be launched.

Retry on error: If an error occurs on this CPU during the dialogue, it allows to define the time in seconds at the end of which a new dialogue will be attempted. This parameter is only valid if an ending hour is specified. The *(N/A)* value indicates to not realize a new dialogue in case of error.

Options:

- *Disabled*: Allows to temporarily disable a CPU in the automatic dialogues. Caution: the CPU will no longer be updated.

- *Modules test*: Performs a modules test during the operation. This may generate an alert depending on the settings defined in the CPU. (TIP CPUs only, see 1.11, page 37).

1.8 Users

▷ Misc, Users

It is possible to define several users to control access to GIR Titan-Hyperion. A user is defined by a name (User field) and a Password to enter on the connection screen.

According to the application configuration (e.g. for an intranet integration), users may also connect without having to enter their password. In this case, an Options field is also available, and allows to specify if a user's identification is to be done automatically or not.

Each user can be limited to specific operations, according to the value defined in the Level field and the criteria in the Authorizations tab.

Two definition modes are available for authorizations: classic and advanced. The mode can be chosen in the Manager Config. (See 1.9, page 31).

1.8.1   Classic mode

Each user is given an access Level. The different access levels are described
below, from the most restrictive to the most permissive. Each access level gives
a set of operations, as well as operations authorized at inferior levels (e.g. a
Manager can make all the operations authorized to *Standard* and *Consultation*
levels).

The following sums up which operations are authorized on access levels:

- *Consultation*

  - Data tables reading

- *Standard*

  - Manual transactions and supplies entry.
  - Maintenance declaration (for maintenances already defined)
  - Fuel transactions correction (vehicle, driver, meters, activity and specific code)
  - Messages attachment to vehicles and drivers
  - Dialogues: Link test, Retrieve-Send, Gauges and totalizers, Supervision

- *Workshop*

  - Maintenances editing

- *Local manager*

  - Configuration tables reading
  - Vehicles and drivers tables editing
  - Dialogues: all operations

- *Manager*

  - Users definition
  - Price and volume correction on fuel transactions
  - Data tables editing (menu ▷ Misc)
  - Management operations (menu ▷ Config.)
  - Data imports/exports

- *Installer*

  - Configuration tables editing
  - Transactions reprocessing

Manager and *Installer* levels can access the whole database. For other levels, it is possible to restrict access to some sites, directions or departments. This can be done using the Authorizations tab in each user's sheet.

For each user, a list of sites, directions and departments can be defined, conforming to the following rules:

- If no site is defined, the user has access to all sites.

- If at least one site is defined, the user has access to data related to the defined sites (e.g. dialogue with a site's CPUs, site tanks state, transactions realized on a site...).

- If no direction nor any department is defined, the user has access to all directions.

- If at least one direction or department is defined, the user has access to data related to the defined directions and/or departments. (e.g. vehicles and drivers from a department, transactions realized by drivers of a direction...).

The value *(N/A)* for the Access field means that the element is only defined to configure e-mails sending (See 6, page 71), and has no effect on authorizations. Therefore, to define a user that has no access limits, and who receives alerts for a given site, we would declare this site with a *(N/A)* access.

For departments and directions, access may be defined as reading or writing. With a reading access, the user is limited to data consulting. With a writing access, he may realize different operations on data. The table below indicates the different operations categories, and their authorization status for each access level.

(1): Vehicles and drivers entry.

(2): Maintenances definition.

(3): Manual transactions, transactions modification, maintenance declaration, message attachment

Level	Access	(1)	(2)	(3)
Consultation	Reading	-	-	-
	Writing	-	-	-
Standard	Reading	-	-	-
	Writing	-	-	Yes
Workshop	Reading	-	-	-
	Writing	-	Yes	Yes
Local manager	Reading	-	-	-
	Writing	Yes	Yes	Yes

A summary of the access rights for each user is available in the user sheet Summary tab.

1.8.2   Advanced mode

To make the configuration easier, there are only three access levels:

- *Installer*: Full access to the application. Only one installer account can be defined. The installer mode application connection is protected by a dongle.

- *Manager*: Full access to the application, except for hardware configuration which is only available for reading.

- *Standard*: Restricted access to the application, according to the authorizations specified in the Authorizations tab.

A standard account can allow from a limited consultation to authorizations close to those of a manager.

Authorizations are defined by specifying explicitly which operations are allowed to a user, on sites and departments. Departments can be merged into directions, and sites can be merged into regions (See 1.9, page 31).

Authorizations on sites and regions

The operations allowed are:

- *Consultation*: Access to the consultation states (history, sums. . .) for transactions and supplies realized on the site.

- *Dialogues 1*: Basic dialogue operations on the site's CPUs (Link test liaison, Retr.-Send, Supervision).

- *Dialogues 2*: Advanced dialogue operations on the site's CPUs.

- *Manual trans.*: Entry of manual transactions on the site's CPUs.

- *Zones*: (Only with local zones management) Local zones assignment to vehicles and drivers.

- *Tank supply*: Entry of supplies on the site's tanks.

Autorizations on departments and directions

The operations allowed are:

- *Consultation*: Access to the consultation states (history, sums. . .) for transactions involving a department vehicle or driver.

- *Add/Mod.*: Modification of all the department vehicles and drivers, and addition of vehicles or drivers to that department.

- *Mod. id.*: Modification of identification data (badge, code), on the department vehicles and drivers.

- *Add/Mod. departments*: (Directions only) Creation and modification of departments inside the direction.

- *Maint. mod.*: Modification of maintenances defined on the department vehicles.

- *Maint. entry*: Declaration of maintenances done for the department vehicles.

- *Manual trans.*: Manual transaction entry for the department vehicles and drivers.

- *Access int. tr. mod.*: Allows to modify internal transactions done by the department vehicles or drivers.

- *Access ext. tr. mod.*: Allows to modify external transactions done by the department vehicles or drivers.

- *Assign tr. mod.*: Allows to reassign the department vehicles or drivers when modifying a transaction.

- *Remote trans.*: Remote transactions control for the department vehicles and drivers.

- *Closure*: Department closure (See 1.3.1, page 22)

Notes

There are default authorizations and specific authorizations for sites, as well as for directions and departments. The default authorization applies for all the sites (resp. directions/departments) which are not explicitely mentionned in the specific authorizations.

Example: There are 3 sites A, B and C. We would like to define a user that could enter supplies on the B site and that could only consult the supplies of the A and C sites. To do so, we would tick Consultation in the default authorizations and then add the B site in the specific authorizations, ticking Consultation and Tank supplies.

A user is authorized to consult a state when he has at least one element for which consultation is enabled. For instance, a transaction will be visible in a user's history if consultation is enabled for the vehicle, the user or the site.

The other operations that realize data modifications are only possible when all the elements are authorized. For example, a user will be allowed to enter a manual transaction related to a user, a vehicle, and a site only if he is authorized to do the *Manual trans.* operation for those three elements.

As for the classic mode, the Summary tab allows to view the authorizations applied by the system for a given user.

1.9 Manager config.

▷ Misc, Manager config.
   This table defines the general settings of the application.

1.9.1 Identification

Type: Default identification method when creating a new vehicle or driver .

Identifier: Default identifier when creating a new vehicle or driver .

Options:

- *Strict proxi file*: Automatically assigns badges using the proximity files, without prompting for confirmation.
- *Hide id+badge code*: Hides the fields Identifier and Badge code in the vehicle sheet (resp. driver). This option can be used with *Strict proxi file* to handle badges assignement totally automatically.
- *Set vehicle id.* (resp. driver): Sets the *Vehicle id.* (resp. *Driver id.*) option when creating a new driver (resp. vehicle).
- *Hide vehicle id.* (resp. driver): Hides the *Vehicle id.* (resp. *Driver id.*) option in the driver creation form (resp. vehicle).
- *Hide badge num.*: Hides the Badge num. field in the drivers and vehicles list.
- *Hide code*: Hides the Code field in the drivers and vehicles list. The code will still be visible in the detailled sheet (General tab) and in the modification form.
- *Conceal Code*, *Conceal Secret code*: Activates the concealed-mode entry for codes (resp. secret codes). In this modes, codes are never displayed clearly on the screen.
- *Mandatory expiry*: Makes the Expiry field entry mandatory in the vehicle (resp. driver) sheet.

Display, Display opt.: Defines the fields to use to display a vehicle or driver.

Code prefix, Nb. char. code autogen.: Defines the value of the Code field when creating a new vehicle or driver. The code is made of a common prefix, followed by N automatically generated digits to create a unique code.

1.9.2 Fuel

Overcons. tolerance: Defines the consumption beyond which a transaction is considered as overconsuming.
For instance, if the reference consumption is 10.00 gal/100miles, and that the tolerance is 20%, a transaction will be considered as overconsuming if its instantaneous consumption exceeds 12.00 gal/100miles.

Undercons. tolerance: Defines the consumption below which a transaction is considered as underconsuming.

Min covered distance for cons.: Defines the mininum covered mileage that should have a transaction to compute the instantaneous consumption. This allows to avoid generating overconsuming transactions when the covered distance is too low to compute a relevant consumption.

Min covered time for cons.: Same thing for the number of covered hours.

Merge consecutive meters: When two identical fuel transactions are made within the defined time, meters and consumptions are assigned to the most recent one.

Fuel trans. dept.: When processing a fuel transaction, this setting defines to which department the transaction will be assigned. It may be either the vehicle department, or the driver department. The department assigned to a transaction is used in the sums (See 5.1.2, page 64).

Imported trans. meter: Defines how to handle milometers and hour meters when importing transactions (See 12.1, page 97).

- *Always*: The transaction meter is used to update the vehicle.
- *Except add-on*: The transaction meter is used to update the vehicle, except for add-on transactions.
- *Never*: The transaction meter is never used to update the vehicle.

Fuel options:

- *Set 'Undercons.' on negative covered*: When this option is set, fuel transactions having a covered mileage equal to or under zero are indicated as underconsuming. This allows to include those transactions among the noteworthy transactions(See 5.1.1, page 63).

1.9.3 Miscellaneous settings

CSV section: Defines the format of downloadable CSV files.

Spreadsheet	Date format	Decimal delimiter	Field delimiter
Excel	mm-jj-aaaa	Dot	Semicolon

HTTP proxy section: Defines the proxy server settings for the internet access used by software support.

E-Mails section: Defines how and when to send e-mails (See 6, page 71).

Regional parameters section: Defines regional parameters to use: dates format, decimal delimiter, units. Dates format and decimal delimiter are used on both display and data entry controls.

1.9.4 Users authorizations

Auth. settings : Defines the mode used to set users authorizations (See 1.8, page 27).

- *Classic*: Historical setting, grouped with e-mails definition.
- *Advanced*: This setting brings a greater flexibility when defining which operations are permitted to each user. It is separated for e-mails management, and more adapted to a large-scale application.
- *Advanced (detailled)*: This setting is similar to the advanced mode, but offers the additionnal possibility to define independant authorizations on drivers and on vehicles.

Options :

- *Prevent force . . . manual tr.*: Prevents entry of manual transaction for forbidden vehicles or drivers.
- *Prevent force . . . modif.*: Prevents assignment of forbidden vehicles or drivers in transaction modification.
- *Prevent force . . . remote tr.*: Prevents launching remote transactions for forbidden vehicles or drivers.

1.9.5   General parameters

Features: Those options enable or hide specific features in the application.

- *Gauges manual entry*: Enables the entry of manual gaugings in ▷ Tanks state (See 5.2.1, page 66).
- *Totalizers manual entry*: Enables the entry of manual totalizers in ▷ Tanks state (See 5.2.1, page 66).
- *Temperature corr.*: Enables temperature correction. When products have been configured (See 1.11, page 39), this allows to automatically correct the volume according to the measured temperature during a tank supply (See 7.2, page 77).
- *Set 'U.P. at Tref'*: When temperature correction is enabled, this option allows to interpret by default the unit price entered as price at the reference temperature, instead of price at the supply temperature (See 7.2, page 77). *Hide 'U.P. at Tref'* hides the checkbox in the tank supply form.
- *Temperature mandatory*: Makes the temperature entry mandatory in tank supplies.
- *Prod. providers*: Enables entry of a provider and a comment in tank supplies(See 7.2, page 77).
- *Provider mandatory*: Makes the provider entry mandatory in tank supplies.
- *Fuel activity code*, *Fuel spec. code*: Shows activity and specific codes in the fuel transactions history.
- *Vehicles auth. by driver*: Enables per-driver vehicles authorizations. This involves an additionnal tab in the vehicle sheet (showing whom drivers are authorized for this vehicle), and in the driver sheet (allowing to define which vehicles are authorized to this driver).
- *Ext. vehicle*: Sets the corresponding option on the fuel transactions for which the vehicle department differs from the driver department, except when the vehicle or driver has the *Inhib ext. vehic.* option.
- *Ask spec. code on ext. tr.*: Enables specific code entry for external transactions.
- *Monetary credits*: Enables monetary credits management (See 10, page 89).

- *Vehicles models*: Enables the Model field in vehicles, as well as the Vehicles models table in the ▷ Misc menu. Vehicles models are similar to vehicles types: they allow to group vehicle on an additionnal criterion, that can then be used in sums.

  Vehicles models can also be used to define general consumption settings (See 1.1.8, page 18).

- *Hide Ids*: Hides the Id. field in the multiple histories.

- *Show vehicle/driver num.*: Shows the Num. field for vehicles and drivers. This number is automatically assigned by GIR Titan-Hyperion, and is unique for each vehicle (resp. driver). This number may also be used to identify a vehicle or driver on the ticket printed after a transaction (See 2.7, page 50).

- *Manual trans. regul. for all*: Allows all users to enter regularization transactions (See 7.1, page 75) .

- *Hide maintenances*: Hide all data relayed to maintenances management in the user interface.

- *Hide credits*: Hide all data related to credits management in the user interface.

- *Hist. menu*: Displays a link to the fuel transactions history in the main menu.

- *Regions*: Enables assignment of sites to regions.

- *Explicit zones*: Simplifies zones management by only displaying the defined zones in the modification forms.

- *Charts on 100 elts*: Increases the number of elements displayed in graphical sums (See 5.1.3, page 65).

Date fields: Defines how dates should be entered: text input or dropdown lists.

Options:

- *Group spec. codes*: If this option is set, when a specific code has been entered in a transaction, GIR Titan-Hyperion searches this code in the Specific codes table. If the code is found, the transaction is attached to this code. Otherwise, the code is created. When this option is not set, specific codes are only stored as strings in the transactions, and the Specific codes table is not used. Grouping specific codes allows to compute sums on the transactions sharing an identical code.

- *Conceal users pwd*: Enables the entry of users passwords in concealed mode.

Closure day: Enables fuel transactions closure. When this field is zero, the feature is disabled. Otherwise, it defines the day in the month after which transactions made the month before must be closed (See 1.3.1, page 22).

Protected zones : Defines the zones that only the main manager is entitled to attach to vehicles and drivers. By default, the vehicles and drivers zones can be modified by local managers.

1.9.6   Traclient

This section defines automatic traclient exports configuration (See 12.2, page 99).

Fuel options:

- *Manual tr.*, *Imported tr.*, *Tank supplies*: Defines which transactions should be used in live export. By default, only transactions retrieved from CPUs are exported. By setting those options, it is also possible to export manual and imported transactions, as well as tank supplies.

- *Disable mod. tr. export*: When this option is set, only the original transaction is exported live. If modifications are made after that, they won't be exported.

Generation day : For sum-based exports, defines the day in the month when they should be generated (e.g. if this field is set to 10, the sum concerning March 2005 will be generated on April 10th). This allows to leave a small time range to regularize transactions.

Generation hour: For sums-based exports, defines the hour in the day when they should be generated. This value should be set so that all the transactions done the day before are retrieved when generating the export.

Vehicle name: Defines the field to use as registration number for vehicles in all traclient formats.

Fuel, Access, Fuel sums: Defines the format of the live-generated traclient files (See 12.2, page 99). Leave to *(N/A)* to generate no traclient file.

1.9.7   Invoices

This section defines general invoicing configuration (See 11, page 93).

Geometric coordinates: Those numerical values allow to configure the header of generated invoices, by defining the header total height, and the coordinates of the rectangles containing customer and provider addresses.

Traclient format: Defines the traclient format to use when generating downloadable invoicing files.

Transcoding format: Defines the transcoding format to use when generating downloadable invoicing files.

Options:

- *Multiple profiles*: Enables multiple invoicing profiles (A,B,C,...). Each profile has its own configuration.

1.10   Profiles

▷ Misc, Profiles

The users profiles allow to adapt some settings according to the connected user.

Priority, User, IP address mask: Those fields define the attachment of a profile to one or more users. A profile is associated to a user when both User and IP address mask criteria are compatible to the connected user. The User field is compatible if it is *(N/A)* or if it is equal to the user. The IP address mask field is compatible if it is empty or if the address from where the user is connected matches this mask. It is possible to specify a DNS name or an IP address, with the ? and * joker characters (e.g. *.example.com, 192.168.1.*, . . .). The Priority field defines the order in which the profiles will be tested. 0 is the highest priority, and 254 is the lowest.

Dates format, Decimal delimiter, Field delimiter: Defines the format of downloadable CSV files. If a parameter is set to *(N/A)*, the value defined in Manager config. is used.

Reader address, Reader type: Address of a badge reader in a host:port format. If this field is filled in, a Read badge button will be available in the vehicles and users sheets for users having this profile. This button allows to directly read a badge on the defined reader, and to assign it to the selected vehicle or driver.

Example: We would like to use CSV parameters different from those defined in the Manager config. for all the users who connect from the example.com domain, except for the A user. To do so, we would then define the following profiles:

Priority	User	IP address mask	CSV parameters
0	A	Empty	All *(N/A)* (to use Manager config. settings)
1	*(N/A)*	*.example.com	Specific settings

1.11 Configuration

The following tables define CPUs configuration. They are set by GIR Titan-Hyperion's installer and available to the manager for reading only, under the ▷ Config. menu.

- Installer config.

- Products

- Sites

- Tanks

- CPUs

- Badges formats

- Badges types

- Access types

Some of these tables values are editable by the manager, using the ▷ Config.,
CPU, tanks tresholds, ref. and zones management   menu. They are classified
into several categories:

CPUs (settings) :

State: CPU state (*Enabled* or *Disabled*). Enabling or disabling a CPU
is done by clicking on the corresponding button. When a cpu is
disabled, it is no longer usable in automatic or manual dialogues.
Entering manual transactions on this CPU pumps remains possible.
Disabling a CPU can be useful in two cases:

- Managing a fictive CPU
- Temporarily disabling dialogues during a CPU failure

Link type: Type of the link between the CPU and the PC: network (tcp),
serial, or modem.

Link param.: Link parameters. Their parsing by GIR Titan-Hyperion
depends on the link type:

- *TCP*: `<host>:<port>`. The host name may be an IP address or
a DNS name (e.g. `192.168.0.123:6001`).
- *Serial*: `<comX>` where X is a serial port number (e.g. `com1`).
- *Modem*: `<comX>:<tel>[:<atdt>[:<at&f>]]` where X is the se-
rial port on which the modem is connected, and `tel` is the phone
number to dial (e.g. `com2:0478741234`). The successive com-
mands sent to the modem are:
  1. `atz`
  2. Command specified in the fourth parameter (`at&f` by de-
fault)
  3. Command specified in the third parameter followed by the
phone number (`atdt` by default)

Mode: Synchronisation mode between GIR Titan-Hyperion and the CPU
(TIP CPUs only).

- *Autonomous*: Connection with the CPU is temporarily estab-
lished during dialogues, which can be manually or automatically
launched (See 3, page 51).
- *Real time*: Connection with the CPU is permanent. Transactions
are retrieved in real time by the software, and data is regularly
synchronized.
- *Autonomous without auto. conn.*: Identical to autonomous mode,
except for the supervision screen where the user must explicitely
ask for the connection to the CPU.

Dialogue alert: For autonomous CPU, it defines how many consecutive
dialogue errors are allowed before an alert is issued. Only errors
occuring during an automatic dialogue are counted.

Generally, this limit will be defined according to the number of auto-
matic retrievals scheduled. Thus, if two retrievals are scheduled each
day, and that we would like to be warned after two days without any
retrieval, a limit of 4 should be define.

When this limit is zero, alerts on dialogue errors are never issued.

Modules alert: (TIP CPUs only)

> TIP CPUs are made of multiple modules (terminal, badge reader, pump ...), connected to a management unit (MU). The dialogue alert, described above, allows to detect connection issues between the MU and the PC. Based on the same principle, modules alert allows to detect connections issues between the MU and modules.
>
> This field defines how many MU/modules link errors are allowed before an alert is issued. Only the errors occuring during an automatic dialogue for which the *Modules test* option has been selected are counted.
>
> When this limit is zero, alerts on modules link errors are never issued.

Real time alert: For real-time CPUs, it defines the maximum interruption length (in minutes) after which an alert is issued. When this limit is zero, alerts on real-time connections are never issued.

Products (settings) :

- Sale unit: volume value and name of the sale unit for this product. When these fields are set, an additionnal column is displayed in the tanks state (See 5.2.1, page 66).

- Options:

  Type: Product type (*Fuel*, *Washing*, *Other*), used to filter transactions in the history.

  Options:
  - *Disable traclient*: Ignores transactions with this product during traclient exports.

- Temperature: (Only when the temperature correction feature has been enabled, see 1.9, page 31).

  These two fields are optional: when one of them is not set, the temperature correction is disabled for the associated product. This allows to limit temperature correction to specific products.

  Reference temp.: Reference temperature (T_{ref}) in Celsius degrees. The coefficient at this temperature is 1.

  Coef.: Coefficient to apply when the temperature is T_{ref} + 10℃.

  Temperature correction is computed using a linear interpolation with the two specified values: if the reference temperature is 15℃, and that the coefficient at 25℃ is 0.9, the coefficient applied at 20℃ will be 0.95.

Tanks (tresholds) :

Blocking treshold: Stock below which the pumps using this tank are automatically blocked. Leave this field empty to never block any pump whatever the tank stock is.

Alert treshold: Stock below which an alert is issued.

Capacity: Tank capacity. When using multiple gauged compartments, it is also possible to modify the capacity of each compartment.

Tanker: When the selected vehicle takes fuel in another tank, a supply with the same volume as the transaction is automatically created on the current tank.

Variation +: Generates an alert when the "theoretical stock - real stock" difference exceeds this limit. For a compartmented tank, the real stock is the sum of the fuel volumes in each compartment.

Variation -: Generates an alert when the "real stock - theoretical stock" difference exceeds this limit.

Max. water: Water volume beyond which an alert is issued. For a compartmented tank, the water volume used is the sum of water volumes in each compartment.

Alert on error: Number of consecutive gauging errors beyond which an alert is issued.

Options:

- *Gauge disabled*: Allows to temporarily disable gauging management for this tank.
- *Temperature corr.*: Enables temperature correction on automatic gaugings for this tank.
- *Alert on manual gauging*: Enables alerts generation when entering manual gaugings for this tank.

The fields Variation +, Variation -, Max. water, and Alert on error are only relevant for tanks with gauging management. Each of them may remain at zero to never generate any alert.

An alert generation creates an event (See 5.3, page 68), displays a message on the home page, and optionally sends an e-mail (See 6, page 71).

Pumps (tresholds) :

Tot. options:

- *Totalizer reading*: Enables automatic report of the internal totalizer for this pump module.

Tot. tolerance: Tolerance limit beyond which alerts are generated on totalizer reports (See 5.2.5, page 68). If this field is zero, no alert is issued.

Ref. : The Ref. fields are used to refer to an element in the specific exports. This menu allows to define the Ref. field as well a the name for some configuration elements (sites, CPUs, tanks, products, . . .).

Zones : Zones define the vehicles and users authorizations on the CPUs (See 1.6, page 24). This menu allows to configure zones on the fuel and access management sets. It is possible to define up to 8 zones per CPU.

Chapter 2

CPU usage

2.1 States

When a CPU is not used for data entry, it may display various messages according to its state:

```
Mo 11/10/04 10:30:00
> SEL.PUMP 1 2 3
```
Several pumps are available to refuel. Pumps for which numbers are not displayed are either in use or unavailable.

```
Mo 11/10/04 10:30:00
PUMP 2 AVAILABLE
```
Only one pump is available. The others are either in use or unavailable.

```
Mo 11/10/04 10:30:00
DELIVERY IN PROGRESS
```
All available pumps are currently in use.

```
Mo 11/10/04 10:30:00
     OUT OF ZONE
```
No pump is available according to the defined time slots.

```
Mo 11/10/04 10:30:00
    OUT OF ORDER
```
No pump is available.

A detailed pump state can be viewed in GIR Titan-Hyperion using the supervision mode (See 3.3, page 53), or on the CPU using the attendant menu (See 2.4, page 46).

2.2   Fueling

Fueling steps on the CPU:

```
┌─────────────────┐
│   Pump choice   │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  Identification │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    Entry of     │
│     meters      │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    Entry of     │
│  optional data  │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│   Distribution  │
└─────────────────┘
```
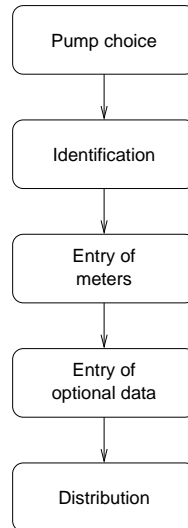
The transaction may be cancelled by the EFF key at any time during data entry. The TRANSACTION CANCELLED message appears and the CPU returns to the waiting state.

2.2.1   Pump selection

```
Mo 11/10/04 10:30:00          Mo 11/10/04 10:30:00
> SEL.PUMP 1 2          or    PUMP 1 AVAILABLE
```

Select the pump number you wish.

```
Pump 1 GOI
Validate your choice
```

The pump and the product name appear, validate with VAL.

Single pump

When only a single pump is defined on the CPU, pump selection is skipped and identification is directly asked.

```
Mo 11/10/04 10:30:00          Mo 11/10/04 10:30:00
ID VEHICLE              or    ID DRIVER
```

2.2.2 Identification

- When identifying the vehicle first:

```
Pump 1 GOI
ID VEHICLE
```

- When identifying the driver first:

```
Pump 1 GOI
ID DRIVER
```

Identification is processed by presenting the badge to the reader, or by entering the code and validating with VAL.

For a double identification, (vehicle with *Driver id.* option, or driver with *Vehicle id.* option), both identifications will be successively requested.

Error messages:

- BADGE NOT RECOGN., UNKNOWN BADGE: The badge is not defined.

- UNKOWN CODE: The code is not defined.

- VEHICLE FORBIDDEN: The identified vehicle is forbidden.

- DRIVER FORBIDDEN: The identified driver is forbidden.

- PRODUCT NOT AUTHOR.: The pump product is not authorized to this vehicle.

It is possible to ask for a secret code in addition to the badge identification (Secret code option for vehicles and drivers). The entry of such a code is asked by the following message:

```
Pump 1 GOI
PIN CODE
```

Enter the code and validate with VAL.

2.2.3 Meters entry

Meters are asked by the following messages:

```
Pump 1 GOI
Meter mi
```

Enter the milometer and validate with VAL.

```
Pump 1 GOI
Mtr 1/100 H
```

Enter the hour meter, and validate with VAL (Example: for 120.75 hours,

enter 12075).

Error message:

- INCORRECT VALUE: The meter entered is out of the validity range. See 4.2, page 60 for problems related to meters entry.

2.2.4 Additional entries

Some additional entries may be asked, depending on the vehicle and driver configurations:

```
Pump 1 GOI
Activity
```

Enter the activity code and validate with VAL.

```
Pump 1 GOI
Spec code
```

Enter the specific code and validate with VAL.
Error message:

- UNKNOWN CODE: Undefined activity code.

2.2.5 Delivery

Once everything has been entered, the delivery starts and the CPU displays READY FOR DELIVERY.

The fueling then depends on the pump configuration, based on the following variables:

T_{start}: Beginning of the delivery time

T_{end}: End of the delivery time

T_{after}: Post-delivery period

T_{start} and T_{end} are defined in the pump configuration, and can be viewed in ▷ Tanks state by clicking on Detail for the selected pump. T_{after} is fixed to 2 seconds.

A fueling conforms to the following rules:

- If nothing is delivered for T_{start} seconds, the delivery stops.

- Once a first output has occured, if no other ouput is made for T_{end} seconds, the delivery stops.

- If a hangup is managed, the delivery stops when the pump nozzle is hanged up (the two above time periods remain valid).

- Once the delivery has been stopped, the volume output is still counted for T_{after} seconds. This allows to take into account compressing phenomena in the delivery pipe.

Null volumes

Two other values are defined in the pump configuration and may be consulted in the tanks state screen:

T_0: mininum transaction length to count it as a null volume

N_0: number of consecutive null volumes before blocking the pump

A fueling is considered as null when it doesn't carry out any debit and that its length is longer than T_0. When N_0 unsucessful fuelings successively occur on the same pump, it is automatically blocked.

2.3 Attendant

The attendant is a particular vehicle or driver, who is allowed to enter transactions for other vehicles or drivers. It allows to:

Vehicle attendant:

- identify a vehicle without its badge
- force the fueling for a forbidden vehicle
- force a meter value

Driver attendant:

- identify a driver without its badge
- force the fueling for a forbidden driver

The identification process for an attendant is the same as for a normal vehicle or driver (See 2.2.2, page 43). Once the attendant is identified, the code of the vehicle or driver to which the transaction must be assigned is asked:

```
Pump 1 GOI
CODE VEHIC.
```
or
```
Pump 1 GOI
CODE DRIVER
```

Enter the code and validate with VAL.

The transaction continues as if the vehicle or driver was normally identified, with two additional possibilities:

- If the vehicle or driver is forbidden, it is still possible to make the transaction:

```
VEHICULE FORBIDDEN
OVERRIDE
```
or
```
DRIVER FORBIDDEN
OVERRIDE
```

Validate with VAL to bypass the ban.

- If a range control encounters an incorrect meter value, it is possible to force the value:

```
Pump 1 GOI
force Mi to :
```
or
```
Pump 1 GOI
force H to :
```

Validate with VAL to force the meter.

2.4   Attendant menu access

To access the attendant menu, press the 0 key when the CPU is idle.

If only one pump is defined:

```
Mo 11/10/04 10:30:00
ID VEHICLE
```
Press 0 then VAL.

If several pumps are defined:

```
Mo 11/10/04 10:30:00
> SEL.PUMP 1 2
```
Press 0.

The identification of the vehicle or driver (depending on the configuration) accessing the attendant menu is then asked:

```
Mo 11/10/04 10:30:00
ID VEHICLE
```
or
```
Mo 11/10/04 10:30:00
ID DRIVER
```

A vehicle or driver is authorized to accesss the attendant menu when the option *Attendant menu* is enabled (See 1.1.5, page 16 and 1.2.3, page 20). Otherwise, the MENU ACCESS DENIED message appears.

The attendant menu content depends on the CPU type.

2.5   Attendant menu - TIP1-Pabbay CPUs

The attendant menu on TIP1-Pabbay CPUs appears as follows:

```
Tanks Pumps Clock
Gauges Instal badGes
```

A submenu is accessed by pressing the corresponding number:

Tanks (key 1): Consultation of tank's theoretical stocks, entry of supplies.

Pumps (key 2): Consultation of totalizers, pump blocking or unblocking.

Clock (key 3): Modification of the CPU date and time.

Gauges (key 4): Consultation of gauging information.

Instal (key 5): *(N/A)*

Badges (key 6): Proximity badges initialization (See 1.1.3, page 15).

2.5.1 Tanks submenu

```
C1 GOI          1500 g
Supply   Upd    (+/-)
```

First line:

- Tank number (from 1 to 16). Use the + (or 9) and - (or 7) keys to go from one tank to another.
- Product name
- Tank theoretical stock.

Second line:

- Supply (key 1): Tank supply entry
- Upd (key 2): Tank stock update (regulation)

Supply entry

```
1 Suppl:        gal
Price  :        Euros
```

Enter the volume of the supply and validate with VAL, then enter total price in euros (it is possible to enter a decimal number, for instace `150.85`), and validate with VAL.

Stock update

```
C1 GOI          1500 g
Corr:           gal
```

Enter the volume to add to the stock, and validate with VAL. It is possible to decrease the tank stock by entering a negative volume (e.g. `-10` to remove 10 gallons).

2.5.2 Pumps submenu

```
 1 GOI       T:002902
On              (+/-)
```

First line:

- Pump number (from 1 to 16). Use the + (or 9) and - (or 7) keys to go from one tank to another.
- Product name
- Value of the pump module internal totalizer

Second line: Pump state. The following states are possible:

On: Pump available.

Blocked/CPU: Pump blocked by the attendant via the attendant menu.

Blocked/PC: Pump blocked by the manager via the supervision screen (See 3.3, page 53).

Counting Error: Pump automatically blocked after a counting error.

Volume error: Pump automatically blocked after several consecutive unsuccessful fuelings (See 2.2.5, page 45).

Module error: Pump automatically blocked because of a module failure.

Config error: Pump automatically blocked because of a configuration error.

Blocking or unblocking a pump is done by pressing the keys B, M, D or 0.

2.5.3   Gauges submenu

A gauging is automatically done when accessing this menu.

```
 1 GOI          1500 g
T  14 H20: 1300 D   -
```

First line:

- Tank number (from 1 to 16). Use the + (or 9) and - (or 7) keys to go from one tank to another.
- Product name
- Tank stock measured by gauging.

Second line:

- T: temperature in Celsius degrees.
- H20: water volume.
- D: gauging status (0 indicates a correct gauging, - or any other value indicates a gauging error).

2.6   Attendant menu - TIP2-Vatersay CPUs

The attendant menu on TIP2-Vatersay CPUs appears as follows:

```
1:PUMPS 2:TANKS
3:MODE
```

A submenu is accessed by pressing the corresponding number. Depending on the configuration, some elements might not be available.

Pumps (key 1): Consultation of totalizers, pump blocking or unblocking.

Tanks (key 2): Consultation of tanks theoretical stocks, entry of supplies, gauging.

Mode (key 3): Restricted mode configuration.

In all this section, any scrolling using the - and + keys is also available with the 7 and 9 keys on numerical keyboards.

2.6.1 Pumps submenu

This menu shows the pumps status:

```
Pump 1 GOI      ./+
ENABLED        [58]
```

The screen contains the following elements:

- On the top left, the pump and product names.

- On the top right, the navigation symbol. It indicates whether there are pumps before or after the current one. When applicable, the + and - keys allow to scroll between the different pumps.

- On the bottom left, the pump state: ENABLED when the pump is available, or DISABLED followed by the blocking cause when it is blocked.

- On the bottom right, the list of available operations, between brackets:

  - Key 5: blocks or unblocks the pump. When pressing this key, a confirmation message shows up (ENABLE? or DISABLE?). The VAL key confirms the operation, EFF cancels it.
  - Key 8: shows the totalizer value. Press 8 again to go back to the first screen.

2.6.2 Tanks submenu

This menu shows the tanks status:

```
Tank 1 GOI      ./+
T:18006L       [568]
```

The screen contains the following elements:

- On the top left, the tank and product names.

- On the top right, the navigation symbol (same usage as for the pumps submenu).

- On the bottom left, the theoretical stock value.

- On the bottom right, the list of available operations, between brackets.

    - Keys 5 and 6: Stock regulation by entering a positive (key 5) or negative (key 6) supply. Confirm the operation with VAL, then enter the supply volume and validate.

    - Key 8: When a gauge is defined on this tank, it is automatically gauged while this screen is displayed. The 8 key scrolls between the gauging values, which are identified by the letters on the bottom left of the screen:

      Jvc: Fuel volume

      Jhc: Fuel height

      Jve: Water volume

      Jhe: Water height

      Jt: Temperature

      The (*) symbol means that an error occured during the last gauging.

2.6.3   Mode submenu

This menu shows the fuel set operating mode:

```
RESTRICTED MODE  ./.
DISABLED         [5]
```

Press 5 to enable or disable the restricted mode. When this mode is enabled, a R is displayed next to the hour. Depending on the configuration, the restricted mode allows to forbid some pumps, or to limit transactions maximum duration.

2.7   Printing

2.7.1   Live printing

Live printing allows to keep on paper an history of all the transactions registered on a CPU.

2.7.2   Ticket printing

Ticket printing allows to deliver a receipt at the end of each fueling.

Chapter 3

CPU communication

3.1 Principle

Each CPU operates in an autonomous way: it possesses a copy of the vehicles and users data in memory, and registers the transactions. It is then necessary to regularly synchronize data between GIR Titan-Hyperion and the CPUs. This synchronization operates in two distinct steps:

1. Retrieval and processing in GIR Titan-Hyperion of the transactions stored on the CPUs. TIP CPUs can store up to 2500 transactions.

2. Update of the CPUs data from GIR Titan-Hyperion database (meters, blocking, etc.)

Any change realized in GIR Titan-Hyperion (message assignment, ban, ...) will really be active on the CPU only after having made a dialogue with it.

3.1.1 Multiple CPUs

Synchronization between GIR Titan-Hyperion and the different CPUs is especially important when multiple CPUs are being used. Indeed, each CPU possesses its own copy of the database. When entering a new meter, this copy is updated with the new meter and the other CPUs remain with the previous value.

Example with a two-CPUs system: The vehicle V has a milometer which value is 50000 mi. Its tolerance range (See 4.1, page 59) is 1000 mi. This information stored by GIR Titan-Hyperion and sent to CPUs B_1 and B_2.

GIR Titan-Hyperion	CPU B_1	CPU B_2
Vehicle V, 50000 mi	Vehicle V, 50000 mi	Vehicle V, 50000 mi

A fueling is made on the CPU B_1 for the vehicle V, and the meter entered is 50800 mi. The situation is now the following:

GIR Titan-Hyperion	CPU B_1	CPU B_2
Vehicle V, 50000 mi	Vehicle V, 50800 mi	Vehicle V, 50000 mi

If the vehicle V later comes to the B_2 CPU, with a meter of 51500 mi, and that no dialogue was made to synchronise data, the new meter will be refused. Indeed, on the B_2 CPU, the last known meter for this vehicle is still 50000 mi. Given the defined range, the maximum meter value is 51000 mi.

If a dialogue occurs between the two fuelings of the vehicle V, the situation would be the following:

GIR Titan-Hyperion	CPU B_1	CPU B_2
Vehicle V, 50800 mi	Vehicle V, 50800 mi	Vehicle V, 50800 mi

The new 51500 milometer would then be accepted on both CPUs.

To avoid such issues, it is recommended to regularly synchronize data, so that there is always at least one synchronization between two fuelings of the same vehicle. The best solution is to define an automatic dialogue (See 1.7, page 26). Generally, scheduling a dialogue once every 24 hours is sufficient (this should be adapted if the same vehicle is likely to make several fuelings per day).

3.2   Manual dialogue

The ▷ CPU comm. menu allows to launch a communication between GIR Titan-Hyperion and one or several CPUs.

To start a dialogue, click on Launch after selecting the CPUs, the operation and the desired options. The dialogue report screen is then displayed. It allows to control the operation progress on the selected CPUs.

Operation:

- *Retr.-Send*: This is the most common operation. It allows to synchronize data between GIR Titan-Hyperion and the CPUs.

- *Retr.-Send-Gauge*: Makes a synchronization, then performs a gauging to retrieve the tanks stocks (only with a gauge module). Gauging data is stored in a history.

- *Retr.-Send-Totalizers*: Makes a synchronization, then retrieves the internal totalizers of the pump modules for which a report is enabled.

- *Retr.-Send-Totalizers-Gauge*: Makes a synchronization, retrieves totalizers, and performs a gauging.

- *Retr.-Config.-Send*: Retrieves transactions, sends configuration, and performs a full data sending (*).

- *Retr.-Init.-Config.-Send*: Initializes a CPU with a new sequence, after retrieving the transactions (*).

- *Init.-Config.-Send*: Initializes a CPU with a new sequence. Caution: this operation erases the transactions stored on the CPU (*).

- *Retr.-Wait-Init.-Config.-Send*: Initializes a CPU with a new sequence, after making sure that the CPU is unused. If fuelings or entries are in progress, GIR Titan-Hyperion waits for them to be finished before initializing the CPU.

- *Memory dump*: Copies the CPU memory into a file.

- *Reboot*: Reboots the CPU (*).
- *Link test*: Checks the link between the CPU and GIR Titan-Hyperion, and shows a report on the modules status.
- *Modules update - Link test*: Resets all modules then makes a link test as above.

(*): These operations cause a temporary interruption on the CPU and must not be launched if a transaction is pending.

Options:

- *Detailed log*: Generates a detailed file about the exchanges between the CPU an GIR Titan-Hyperion. These files can be very large.
- *Force retr.*: Allows to force transactions retrieval when it fails indicating a 'Modified module' error.

Sending: Sending mode selection. *Delta* only sends the necessary data to the terminal, *Complete* sends all the data. Thus, should we possess a synchronized base, and should we attach a message to a vehicle, the *Delta* sending will only send one vehicle to the CPU whereas the complete sending will send 3000 vehicles.

Address: Leave to *Normal*.

The default settings (disabled *Detailed log* and *Force retr.*, *Delta* sending and *Normal* address) fit for a normal use of the software and should not be modified unless there is a problem.

Disconnect: By default, the connection to a CPU is established at the beginning of an operation, and closed at the end of the operation (*After operation*). However, it may be interesting to keep the connection during a certain period, should we wish to carry out several successive operations on the same CPU. This is particularly true when the opening phase takes a long time, for instance for a modem connection.

Two disconnection modes are available for this purpose:

- *Interactive*: The connection is closed after 6 minutes of inactivity.
- *Stay connected until*: The connection is closed at the specified hour.

It is possible to launch a dialogue in one of those two modes only when a single CPU is selected.

3.3 Supervision

▷ Supervision

In addition to the dialogue operations seen above, GIR Titan-Hyperion allows to interact directly with a CPU, thanks to the supervision mode. The supervision screen shows the tanks relating to the selected CPU, and the pumps associated to each tank. It allows to:

- view tanks stocks

- perform tanks gauging

- view pumps states and totalizers

- view pending transactions

- start a remote transaction

Click on Update data to launch a connection on CPU and update the displayed data.

CPU state displays the CPU status, allowing to detect a potential system failure. The Last update date indicates when the displayed information has been retrieved.

The connection to the CPU is realized using the interactive mode (See Disconnect, page 53).

Tanks data:

- Theo. stock: Tank theoretical stock, updated according to supplies and transactions.

- Real stock: Tank stock read by gauging (only with a gauge module). Detailed gauging information may be consulted by clicking on Real stock. A gauging is automatically launched for each fueling. A gauging can be launched on all tanks by clicking on the Force gauging button. The gauging information retrieved using this method is intended for temporary consultation only. To carry out a gauging which result is kept in an history, use a *Retr.-Send-Gauge* dialogue.

Pumps data:

- Pump state.

    - *Manual mode*: Pump in manual mode
    - *Blocked*: Pump blocked, the cause of the blocking is displayed just after:
        * *manager*: Pump blocked by the manager via the supervision screen.
        * *attendant*: Pump blocked by the manager via the attendant menu.
        * *counting error*: Pump automatically blocked after a counting error
        * *N null volumes*: Pump automatically blocked after several consecutive unsuccessful fuelings (See 2.2.5, page 45)
        * *module error*: Pump automatically blocked after a module failure
        * *config. error*: Pump automatically blocked because of a configuration error.
    - *Comm. error*: Pump module communication error.
    - *Delivery*: Fuel delivery in progress.
    - *Entry in progress*: A driver is entering a transaction on this pump.

– *Tank blocked*: Pump blocked because the tank stock is below the blocking stock.

– *Out of zone*: Pump out of zone.

– *Available*: Pump available.

Detailed information on a possible pending transaction, as well as the totalizer value are available by clicking on Detail below the selected pump. The totalizer displayed is the internal totalizer of the pump module (and not the totalizer of another counting device).

3.3.1 Pump blocking

Depending on the pump state, one of the following operations is available:

- Block

- Unblock

- Cancel blocking

- Cancel unblocking

Blocking or unblocking a pump requires to launch a *Retr.-Send* operation on the CPU. Hence, after clicking on Block or Unblock, a confirmation screen appears and allows either to launch the operation immediately or to postpone it. When the operation is postponed, the request can be cancelled by Cancel blocking or Cancel unblocking.

As the *Retr.-Send* operation may be quite long, postponing it may be interesting when several pumps must be changed at the same time. In this case, we would chose Later for all the pumps and Immediate for the last one.

3.3.2 Remote transaction

When a pump is available, it is possible to remotely start a transaction by clicking on Transaction. GIR Titan-Hyperion then tries to reserve the pump.

On success, the message "Pump <pump name> reserved" appears. When a pump is reserved, it is no longer available for a fueling on the CPU. The reservation can be cancelled at any time by clicking on Cancel reservation.

Enter a transaction allows to define the various transaction parameters (vehicle, driver, meters...) in a form similar to the one used for manual transaction (See 7.1, page 75).

After entering all the required information, the transaction is started on the CPU and the fueling may begin.

When a pump is remotely reserved but that no transaction is entered, it is automatically released after a couple of minutes.

As for an attendant transaction (See 2.3, page 45), it is possible to bypass certain limitations on a remote transaction. In this case, a warning will appear with the nature of the forced limitation. The following controls can be bypassed:

- Explicit ban

- Expiry date

- Zone-based authorization

- Blocking because of a maintenance not done

- Blocking because of an insufficient credit

3.4   Resolving errors

In addition to simple communication errors, there are also various dialogue errors which require an human intervention to be solved.

3.4.1   Modified module

Cause: Each time configuration is sent, GIR Titan-Hyperion generates a cookie (random number) which is stored on both the CPU and the PC. During each dialogue, the cookie is read from the CPU and compared to the one on the PC. It they are different, the dialogue fails with a *Modified module* error.

Solution 1:   If the CPU contains transactions that should be retrieved but that the cookie differs (for instance because an old backup was restored), perform a *Retr.-Config.-Send* dialogue ticking the *Force retr. (modified module)* option.

Solution 2:   If the CPU comes from a standard exchange and that the data it may contain can be discarded, perform a *Init.-Config.-Send* dialogue.

3.4.2   Modified config.

Cause: After a change in a CPU configuration, GIR Titan-Hyperion refuses to dialogue with it until the new configuration has not been sent (*Retr.-Config.-Send* or *Retr.-Init.-Config.-Send*).

   Caution: a CPU can be set to *Modified config.* in two other cases:

1. After an application update, when there is a change in the way data is sent.

2. After restoring a `TWB` backup, since the CPU data may be completely incompatible.

Solution: Perform a *Retr.-Config.-Send* or *Retr.-Init.-Config.-Send* dialogue.

3.4.3   Invalid state

Cause: When the CPU contains inconsistent information about the size of transactions stored, the retrieval fails with an *Invalid state* error. For security purpose, GIR Titan-Hyperion doesn't automatically retrieve data when a CPU is in such a state.

Solution: Perform a *Retr.-Send* dialogue, ticking the *Force retr. (invalid state)* option.

3.4.4 Config. in progress

Cause: Dialogue with TIP2-Vatersay CPUs is not possible while the configuration application is being used.

Solution: Exit the configuration application on CPU to perform a dialogue.

Chapter 4

Meters

A vehicle possesses two meters, one for the miles and another for the hours. The prompt for a meter during a fueling can be configured for each product, through the Tanks tab in the vehicle sheet (See 1.1.8, page 18).

4.1 Tolerance ranges

Tolerance ranges allow to check the consistency of meters entered during a fueling. When a meter is entered, if the associated range is higher than zero, the entered value must be between the last known value and the last known value added to the range.

When a meter is equal to zero, no control is made on the entered value.

Examples:

Meter	Range	Authorized values
76000 mi	2000 mi	From 76000 to 78000 mi
182000 mi	15000 mi	From 182000 to 197000 mi
49000 mi	*(N/A)*	No control
(N/A)	5000 mi	No control the first time

When a vehicle is created, its meter is set to zero. The first meter entry defines its current meter. It is possible to initialize a meter:

- On CPU, by typing the meter value as for normal fueling.

- In GIR Titan-Hyperion, by creating a manual transaction with zero as volume (See 7.1, page 75).

It is not advised to define a null range when the associated meter is manually entered. Indeed, no control is then made and a small error when entering the number (e.g. 5500 instead of 55000) will not be detected. The values of covered distances and consumptions will then be wrong and require a manual correction (See 7.3, page 79). It is better to always define a range, even very big (e.g. 10000 mi). An exception can be made for emergency intervention vehicles, for which a null range may be defined so that they won't be blocked when fueling.

When several CPUs are used simultaneously, the meters value must be updated on the various CPUs so that the range control operates correctly (See

3.1.1, page 51).

4.2   Problem when entering a meter

This part describes what to do when a meter is refused with an INCORRECT VALUE message, whereas the entered value really corresponds to the vehicle meter.

1. Make sure that all CPUs have been synchronized since the vehicle's last fueling. This is particularly important when using several CPUs (See 3.1.1, page 51).

2. Check the value of the meter and the associated range in GIR Titan-Hyperion. Various cases are possible:

   - The defined range is too low: the range value must be higher than the maximum distance that a vehicle may cover with a fueling. In this case, fix the range value and perform a synchronization dialogue with the CPUs.

   - The vehicle has made fuelings that have not been entered: enter these fuelings with a manual transaction (See 7.1, page 75) then launch a dialogue, or force the meter using method (1) or (2) below.

   - The meter entered during the last transaction is wrong: use one of the three methods below.

   Three methods are available to perform a fueling when the value of a meter is invalid:
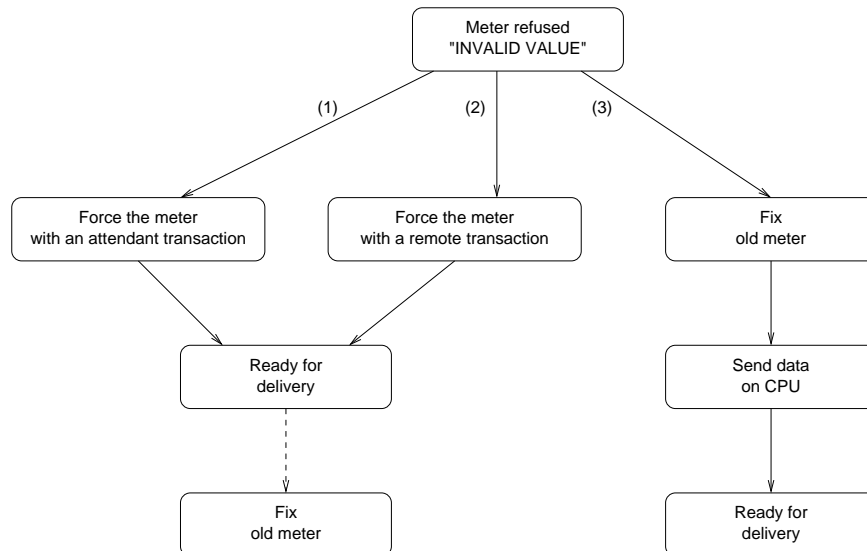
1. Force the new meter with an attendant transaction (See 2.3, page 45).

2. Force the new meter with a remote transaction (See 3.3.2, page 55).

3. Fix the invalid meter in the last transaction (See 7.3, page 79), send fixed data to the CPU, then continue the transaction normally.

Methods (1) and (2) allow to perform the transaction immediately, without having to send data back to the CPU. But when the problem comes from an invalid meter in a previous transaction, they must be completed by the correction of this meter. Indeed, without this correction, covered distances and consumptions will be erroneous.

Example:

Vehicle V possesses a milometer with a 2000 mi. tolerance range. The meter entered during its last transaction was 11500 mi.

It makes a first fueling on 12/05/2004, and enters by error a meter of 13000 mi instead of 12000. As this value is in the tolerance range, it is accepted and the following transaction is created:

	Date	Vehicle	Meter	Covered
T_1	12/05/2004	V	13000 mi	1500 mi

The day after, the vehicle comes back to the CPU and enters its new meter: 12500 mi. The message `INCORRECT VALUE` appears since the last known meter is 13000 mi. Once the error is notified, the meter is forced to 12500 mi using the method (1) or (2). A new transaction is created:

	Date	Vehicle	Meter	Covered
T_2	13/05/2004	V	12500 mi	-500 mi
T_1	12/05/2004	V	13000 mi	1500 mi

The value of covered distance, and the associated consumption is wrong for transactions T_1 and T_2. It is necessary to fix the meter in transaction T_1, so that GIR Titan-Hyperion recomputes covered distance and consumptions:

	Date	Vehicle	Meter	Covered
T_2	13/05/2004	V	12500 mi	500 mi
$T_1^{'}$	12/05/2004	V	12000 mi	500 mi

The transactions stored in history are then correct. Method (3) allows to obtain exactly the same result without having to force a meter, but it requires to fix the erroneous meter and to launch a dialogue before the beginning of the fueling.

If the meter real value is not known, it is possible to erase it. In this case, the transaction $T_1^{'}$ would have meter and covered fields to *(N/A)*, and its volume will be reported on transaction T_2 consumption.

Chapter 5

States consultation

5.1 Fuel transactions

5.1.1 History

▷ Management, Fuel transactions hist.

This screen allows to visualize all fuel transactions stored in GIR Titan-Hyperion. By default, transactions are ordered by decreasing date (antichronological). Two links in the ▷ Management menu allow to access directly to transactions of the current and the previous month.

The Options field notably allows to find the origin of a transaction:

- *Manual*: Transaction manually entered in GIR Titan-Hyperion (See 7.1, page 75).

- *Remote*: Transaction remotely launched on CPU from a PC (See 3.3.2, page 55).

- *Imported*: Transaction imported from oil disks data (See 12.1, page 97).

- *Install.*: Test transaction done during a CPU installation.

If there is no pump on a manual transaction, it means that an off-site fueling has been realized. The *Attendant* option indicates that the transaction was done via an attendant. The attendant identity can be viewed in the transaction sheet.

When a filter is enabled, a sum appears at the bottom of the page. It is computed on all the transactions matching the current filter.

Noteworthy transactions

The (Not.) link in the ▷ Management menu allows to directly view an history filtered on noteworthy transactions. The following transactions are considered as noteworthy:

- Overconsuming or underconsuming transactions

- Remote and attendant transactions

- Manual transactions

- Transactions realized in a vigilance zone (See 1.6, page 24)

A periodical report of noteworthy transactions can be sent by e-mail (See 6, page 71).

5.1.2   Sums

▷ Management/Fuel transactions sums.

This screen allows to compute sums of fuel transactions according to various criteria.

The first criteria group defines the result presentation:

Sum type: Sums types have a section/element form. For instance, a *Department/Vehicle* sum will show, for each department, a table which elements are the vehicles of this department.

Options:

- *Page break between tables*: Allows to show each section of the sum on a separate page when printing.

- *Show last known meter*: When the sum is computed on vehicles, this option displays an additional column containing the last meter of each vehicle on the selected period.

- *Show hour meters*: Displays the number of covered hours and hourly consumption

- *Hide prices*: Hide price-related data (Total and unit price)

- *Show elements with no transactions*: This option allows to display all the elements on which the sum is realized (e.g. all the vehicles) even those that made no fueling during the selected period.

- *Show transac. count*: Diplays a column with the transaction count for each element of the sum.

- *Split by month*: Shows a separated sum for each month in the selected period.

Product detail:

- *No*: Displays only one line per element, summing the values for each product.

- *Ext/Int together*: Displays one line by product for each element, summing on-site and off-site fuelings for the same product.

- *Ext/Int distinct*: Displays one or two lines by product for each element, separating the on-site and off-site fuelings. Off-site fuelings are labelled with *(Ext.)* after the product name.

The filter criteria allow to limit the sum to a time range, a product (with on-site and/or off-site fuelings), a site, etc. The selection of a period is mandatory.

Once all the parameters are selected, click on Compute to display the sum result. Selected parameters are shown in the upper part of the screen. The Modify button allows to change selected parameters to make a new sum.
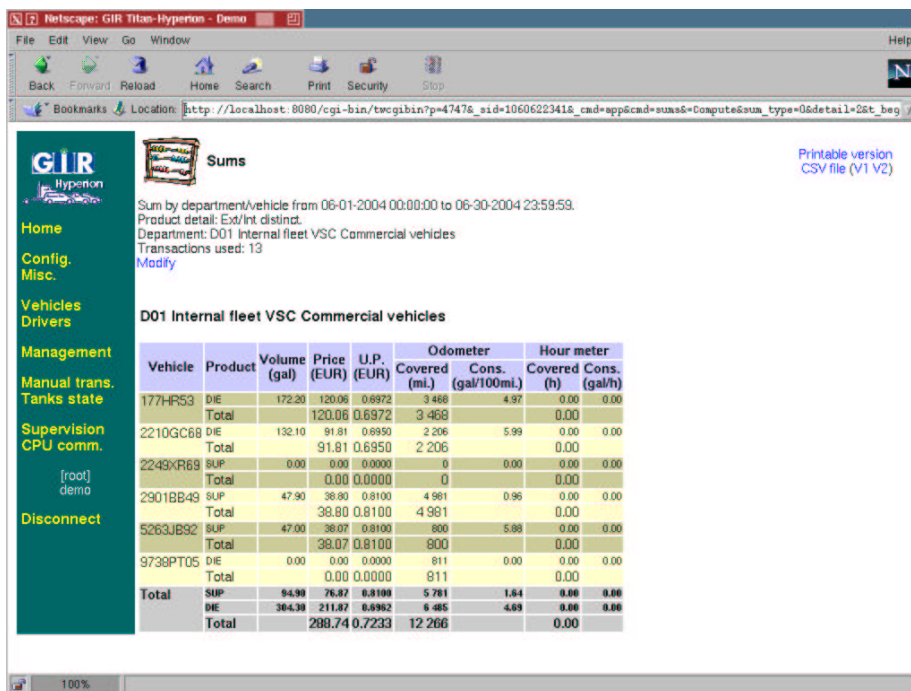
Average consumption computing

There may be a difference between the displayed average consumption and the one obtained from the volume and the covered distance. This is normal and the result of a precision concern, as shown in the example below:

We consider a vehicle which realized 3 transactions, with an unknown meter for the most recent transaction:

Date	Volume	Meter	Covered
15/04/2004	57,00 gal	*(N/A)*	*(N/A)*
08/04/2004	55,00 gal	73300 mi	500 mi
02/04/2004	59,00 gal	72800 mi	600 mi

Should we calculate a sum of these three transactions, we would obtain a 171 gal volume and a 1100 mi covered distance. Computing consumption from these two values by saying that 171 gal were used to cover 1100 mi is wrong: the last transaction doesn't have a meter, and its 57 gal volume must be ignored to compute the average consumption. The correct value is then 114 gal for 1100 mi, i.e. 10.36 gal/100mi.

5.1.3 Charts

▷ Management/Fuel transaction charts.

This screen allows to display fuel transaction sums as histograms.

Orientation: Vertical or horizontal histogram.

Type:

- *History*: Transactions history on the selected period.

- *Average per day*, *Average per week*: Average usage of the station during the selected period, reduced to a day or a week.

- *Dispersion*: Dispersion of the station usage according to the selected criteria. For instance, *Dispersion per vehicle* with the *Volume* value will display the 10 vehicles which took the most fuel during the selected period.

Date Hour: Period selection. When the chart is displayed, navigation buttons allow to move to the previous and next periods, or to change the scale. The available scales are day, week, month and year.

Filtering elements: Type of the elements available in the Value fields.

Value 1 to 3: Type of value to display (transactions count, volume, . . .). Il is also possible to filter on a specific element. For instance, to compare the evolution of two departments consumptions on a year, select *Departments* as the filtering element, then the two desired departments as Value 1 and Value 2.

5.2   Tanks stocks management

5.2.1   Tanks state

▷ Tanks state

This screen allows to graphically visualize the tanks state of a given site, as well as pumps related to each tank.

Information on tanks:

- Current stock indicates the tank theoretical stock, according to the transactions and supplies processed by GIR Titan-Hyperion. It appears in red if it is below the blocking limit defined for this tank, and in orange if it is below the alert limit.

- Detail allows to view detailed information on a tank: stock price, limits, capacity, last supplies . . .

- Tank supply allows to enter a new supply for this tank.

- Manual gauging allows to enter a gauging, which will be stored in an history (▷ Management, Gaugings history). This feature is only available when the corresponding option has been enabled in the Manager config..

Information on pumps:

- Detail allows to view detailed information on a pump: time delays, hangup type, last transactions. . .

- Totalizer allows to enter a new totalizer report, which will be stored in an history (▷ Management, Totalizers hist.). This feature is only available when the corresponding option has been enabled in the Manager config..
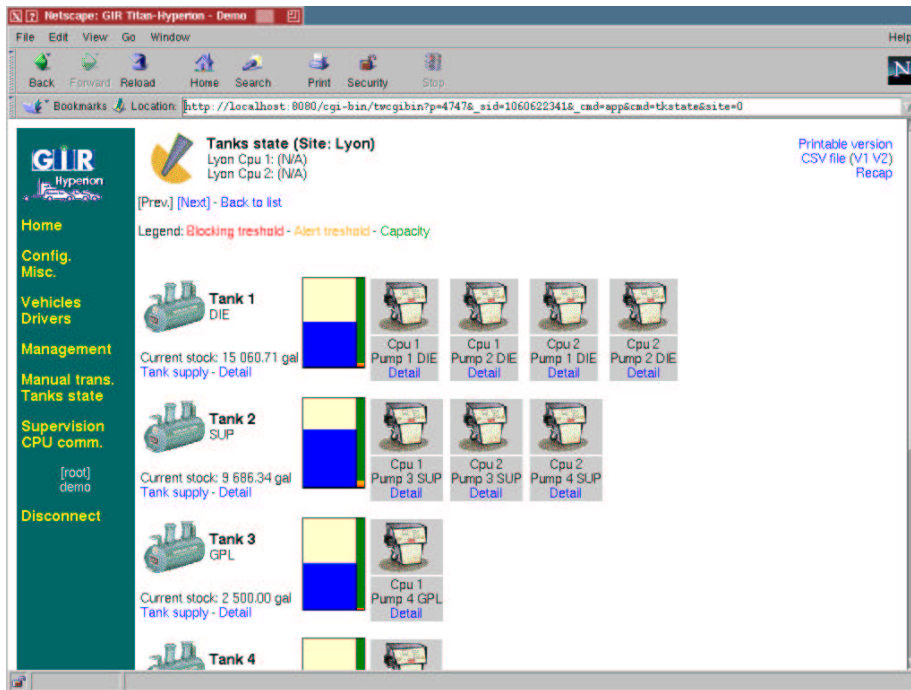
5.2.2 Previous tanks state

▷ Management, Previous tanks state

This screen allows to compute a tank theoretical stock at a given date. This calculation is done using the current tank stock, fuel transactions history and tank supplies history.

Enter the selected date in Date Hour, then click on View. A list of tanks and their theoretical stocks at the selected date is then displayed.

5.2.3 Stock prediction

▷ Management, Tanks stocks prediction

This screen allows to estimate when a tank will need to be supplied. The estimation is based on all the transactions made on each tank during the selected reference period. The displayed time limits show the number of days left before the tank reaches alert and blocking limits.

This estimation is done by computing the average volume removed from a tank during the reference period, for each day of the week. This notably allows to take into account the consumption difference between working days and week-ends.

Example: The average consumption on a tank is 500 gallons per day during working days, and nothing on Saturday and Sunday. A stock prediction is launched on Friday evening with 800 gallons left before reaching the alert treshold. Then the estimated date to reach the alert will be the next Tuesday.

The correct choice of the reference period is very important to obtain an accurate estimation. It is advised to select a period long enough during which

the station activity has been significant. Moreover, all transactions realized during the period must have been retrieved.

5.2.4   Gauging

When gauges are installed, GIR Titan-Hyperion allows to perform a gauging and to consult the obtained information. A gauging can be done through two ways:

- Via the supervision screen (See 3.3, page 53), to consult the gauging values from time to time.

- Via a *Retr.-Send-Gauge* dialogue, to store gauging values in an history (▷ Management, ▷ Gaugings history). This dialogue can be launched either manually (▷ CPU comm.) or automatically (See 1.7, page 26). The latter allows to regularly fill the gauging history, so that it is possible to generate warnings on the measured values.

  In the gauging sheet, the Correction supply button allows to enter a supply where the difference between the theoretical stock and the real stock is already set.

5.2.5   Totalizers

Pump modules have an internal totalizer, which increases when a fueling is detected. GIR Titan-Hyperion allows to automatically read this totalizer, in order to check its consistency compared to the stored transactions. To enable the totalizer report on a pump, tick the *Totalizer report* option in the selected pump module (▷ Config., CPU, tanks tresholds, ref. and zones management)

When a *Retr.-Send-Totalizers* dialogue is launched, the totalizers of the selected pump modules are stored in the Totalizers history table. GIR Titan-Hyperion computes the difference with the previous report (Difference (tot.) field in the history), as well as the total volume of transactions that have been retrieved between the two reports (Difference (trans.) field in the history). The difference between these two values is displayed in the Variation field. It must be close to zero.

5.3   Events

▷ Management, Events

The events table keeps an history of messages generated by GIR Titan-Hyperion. The various elements types are:

- *Warning*: Miscellaneous warnings

- *Invalid badge*: An unknown or invalid badge was read

- *Badge init.*: Transaction created when initializing a proximity badge.

- *CPU*: Transaction created when rebooting a CPU or installing pump modules.

- *Login*: User connection to GIR Titan-Hyperion

- *Maintenances*: A maintenance reached the alert or blocking limit

- *Expiry*: Vehicles or drivers authorization expiry, and credit depletion. An event is generated for credit depletion only when the corresponding option is checked in the credit period (See 10.3, page 90).

- *Overcons.*: Vehicle overconsumption

- *Tank stock*: Tank stock warning

- *Transaction*: Warnings generated during a transaction processing. The transaction can be identified by the CPU and the Id number between brackets in the warning text.

- *Transac. not processed*: Transaction data could not be processed

5.3.1 CPU reboot events

The following events allow to understand the electrical supply of the terminal:

- *Start With Retr.*: Startup with RAM restoration

- *Start No Retr.*: Startup without RAM restoration. The meters stored on CPU are then to zero, which disables the controls during the first meter entry (See 4.1, page 59). It is advised to launch a synchronization between CPU and PC after such an event.

- *Network supplied*: Module supply

- *Reboot Sequence*: Sequence startup

- *Register Retr.*: RAM backup during a tension drop

Appearance order of these events in various real cases:

- on power cut:

    - without any battery:
        1. *Start No Retr.* when the supply comes back
        2. *Network supplied* some seconds later
    - which period is lower than the battery's autonomy:
        1. *Register Retr.* when the cut occurs
        2. *Network supplied* some seconds later
    - which period is bigger than the battery's autonomy:
        1. *Register Retr.* when the cut occurs
        2. *Start With Retr.* when the supply comes back
        3. *Network supplied* some seconds later

- on a reboot, requested from the PC:

    1. *Register Retr.* when processing the request

    2. *Start With Retr.* 30 seconds later

    3. *Network supplied* some seconds later

- on initialization, or sequence crash

    1. *Reboot Sequence* at the sequence reboot date

5.4   Transactions Id.

▷ Management, Transactions Id

The Transactions Id. table allows to control transaction retrieval on CPUs. Each time a transaction is created on a CPU, it is assigned an Id number which is then increased.

When GIR Titan-Hyperion processes a transaction, it is stored in both the dedicated history (fuel, access, event, . . .) and in the Transactions Id. table. Further consultation of this table can ensure that no transaction has been lost.

A record in the Transactions Id. table where the Id number is zero and for which no transaction type is set indicates a CPU initialization.

5.5   Other states

These histories may be consulted via the ▷ Management menu.

Tank supplies hist.: History of supplies made on tanks.

Pumps blockings history: History of pumps' state changes.

Chapter 6

E-mails

GIR Titan-Hyperion can send e-mails, either as alerts (ex: pump blocked, maintenance to do, overconsuming vehicle...), or as periodical reports (daily, weekly or monthly).

Configuring e-mails in GIR Titan-Hyperion is done through 3 steps:

- Global parameters configuration

- Definition of the message to send

- Definition of addresses

6.1 Global parameters

Global parameters for e-mail sending are defined in ▷ Misc, Manager config.:

Sending hour: Hour in the day when periodical reports are sent.

Days: Days in the week when the daily and weekly periodical reports are sent. For weekly reports, the sending day is the first day activated in this field, from Monday.

Sending day: Day from when monthly periodical reports are sent.

SMTP server: SMTP server address. If this field is not set, e-mail sending is disabled.

Sender address: Address used in the sender field for messages sent by GIR Titan-Hyperion. It notably allows to receive a failure notification when a message could not be sent.

The Sending hour, Days, and Sending day fields allow to adjust sending so that data is up to date. Indeed, because of the transactions retrieval mechanism (See 3, page 51), the database may not be up to date when a period finishes.

Example:

We consider a CPU on which transactions are retrieved every morning at 6 a.m. by an automatic dialogue. We want to receive an e-mail every day, with all the transactions made during the day. If this e-mail is sent between midnight

and 6 a.m., transactions retrieval won't be done yet, so the database won't be up to date and transactions history will be incomplete. To avoid this, we can define 8 a.m. as Sending hour, so that all transactions will be retrieved before sending the message.

On the same principle, if automatic retrievals are scheduled on an every day basis but Sunday, Saturday should be unchecked in the Days field to avoid sending an incomplete report on Sunday morning.

For monthly reports, the Sending day field allows to delay sending at the beginning of each month, for instance to regulate some transactions.

When sending a report, GIR Titan-Hyperion checks whereas all the transactions have been retrieved on the various CPUs. If not, the message will contain a warning, as well as the "Incomplete" label in the subject field. Moreover, the message will be sent again once all retrievals are done.

6.2   Messages to send

Messages to send to a user can be defined either in the Authorizations tab or in the E-mails tab in the user sheet, depending on the configuration (See 1.8, page 27). It allows to define which alerts and periodical reports will be sent to a user.

6.2.1   Alerts

Any alert sent as e-mail may also be consulted in the Events and Pumps blockings hist. tables.

Alerts available for sites:

- *Tanks-pumps*: Alerts on tanks levels, pumps states and totalizers reports.

- *Tanks only*: Alerts on tanks levels and totalizers reports.

- *Pumps only*: Alerts on pumps states.

- *Watch*: Alerts when a vehicle or driver carries out a transaction in a 'watched' zone.

Alerts available for directions and departments:

- *Maintenances*: Alert whens a maintenance reaches the alert or blocking limit.

- *Overcons.*: Alerts when a vehicle is overconsumming.

- *Expiries*: Alerts when a vehicle or driver authorization expires, or when a credit is depleted. The latter only appears when an associated event is generated (See 10.3, page 90).

- *CPU*: Alerts on CPU link error (see 1.11, page 37 for detailed conditions to trigger this alert).

- *Invalid badge*: Alerts0 when an unknown badge is used on a CPU.

6.2.2 Periodical reports

Each report may be sent daily, weekly, or monthly. When no period is chosen, report sending is disabled. For some reports, it is possible to chose the content type: fuel, access, or both information types.

For transactions histories, it is possible to receive either a complete history, or an history with only noteworthy transactions (See 5.1.1, page 63) .

Reports available for sites :

- *Stocks*: Tanks stocks, sum by CPU/pump.

- *Transactions*: Supplies and transactions made on the site.

Fuel	Tank supplies history
	Fuel transactions history
Access	Access transactions history

Reports available for directions and departments :

- *Transactions*: Transactions made by members of the direction or department.

Fuel	Fuel transactions history
Access	Access transactions history

- *Sums*: Sums on transactions of the direction or department.

Fuel	Fuel transactions sum by department/vehicle

6.3 Addresses

A user's addresses are defined in the E-Mails tab of the user sheet (▷ Misc, Users). When a user must be informed, GIR Titan-Hyperion sends a message with the same content to each of these addresses.

Three formats are available:

- *HTML*: HTML format, similar to the one used in the application.

- *Printable HTML*: HTML format intended for black and white printing.

- *Text*: Simple text format, only for alerts. This format is intended for an automatic process.

E-mail sending and addresses validity can be tested by clicking on Test in the E-Mails tab. GIR Titan-Hyperion will then send a text format message to the selected address. This test tool is also available in ▷ Config., Verifications, Test e-mail sending.

To send e-mails to a user without allowing him to connect to GIR Titan-Hyperion, just define a user account with a blank password. The definition of messages to send and addresses is then the same as for a normal user.

Chapter 7

Operations entry

7.1 Manual transaction

7.1.1 Manual entry and regulation

A manual transaction can be created for the following cases:

- Entering an off-site fueling

- Totally manual management, with a "fictive" CPU

- Regulation after a failure in automatic management

The two first points can be used under normal system behaviour. They are available via the ▷ Manual trans. menu. The *Pump* field contains a list with the various products available for off-site fuelings, as well as pumps declared on a fictive CPU.

The third case is intended to overcome a temporary problem, for instance an hardware defect which requires the delivery to operate manually for some days. It can be accessed by ▷ Misc, Manual regul. trans. (fuel).

By default, only the manager is entitled to enter regulation transactions, in order to avoid innoportune manual transactions on automatically managed pumps. This can be made available to all users by ticking the corresponding option in *Manager config*. (See 1.9, page 31).

Both entry forms are strictly identical, except for available pumps. A pump is displayed in the menu ▷ Manual trans. if the associated CPU is disabled, otherwise it is displayed in the regulation menu. For more details on enabling and disabling CPUs, see 1.11, page 37.

7.1.2 Entry form

- First screen:

    Date Hour: Date and time of the fueling

    Vehicle or Driver: Depending on the first configured identifier.

    Pump: Pump on which the fueling was realized, or *Product (Ext.)* for an off-site fueling

Volume: Volume of the transaction (with 2 decimal digits)

- Second screen (depends on config.):

  Vehicle or Driver: If a second identifier is required

  U.P., Price: Unit or total price of the transaction (off-site fuelings only).

- Third screen (depends on config.):

  Meter (mi): Milometer

  Meter (H): Hour meter (in hours, with 2 decimal digits)

  Activity: Activity code

  Spec. code: Specific code

  Options:

  - *Renewed meter*: Indicates a meter renewal transaction. This disables the computing of covered distance and consumption for this transaction. Generally, such a transaction will be entered with a null volume.
  - *Add-on*: Indicates that the transaction is a fuel add-on and not a total fueling. This disables the computing of covered distance and consumption for this transaction.

At the end of the entry, all the transaction information is summed up on a confirmation page. Any validy control on meters entered on CPU is also realized when entering a manual transaction: an invalid meter appears in red with the *out of range* label on the confirmation page. This is only a warning, and does not prevent from validating the transaction. It is then possible to force meter values via a manual transaction.

It is also possible to enter a manual transaction with a forbidden vehicle or driver. A warning message will appear after having selected this particular vehicle or driver.

Finally, any message that would appear on CPU for the driver or the vehicle is also displayed during a manual transaction.

7.1.3   Fuel suction

When a vehicle is removed from the fleet, a fuel suction can be done to put its fuel back into a tank. The best way to declare such an operation in GIR Titan-Hyperion is to enter a manual transaction for the concerned vehicle, on a pump related to the tank where the fuel is put back.

The volume of this transaction should be negative (e.g. `-40` if 40 gallons of fuel were retrieved), to remove this volume from the vehicle consumption and add it back to the tank stock.

7.1.4   Remote transaction

When a remote transaction is launched (See 3.3.2, page 55), parameters are entered in a form similar to the manual transaction form, with pump and date already set.

7.2 Tank supply entry

To enter a tank supply, click on Tank supply near the selected tank in the ▷ Tanks state menu.

- First screen:

  Date Hour: Date and time of the supply

  Tank: Tank on which the supply is made

- Second screen:

  Volume: Supply volume (with two decimal digits)

  Purchasing U.P.: Purchasing unit price

  Options:

  - *Regulation*: Indicates that the supply is a regulation. This sets the *Regulation* option in the supply, which will allow to distinguish it from real supplies in the history.

- With the "temperature correction" feature (See 1.9, page 31), a Temperature field is also available. When it is set, the volume entered is considered as the volume at the measured temperature, and the software automatically computes the volume at the reference temperature, using the defined settings (See 1.11, page 39). This corrected volume will be the one added to the tank stock. Its value can be checked on the supply confirmation page.

  In this case, the Purchasing U.P. applies to the entered volume. To enter directly the volume at the reference temperature, tick the option *U.P. at 15℃*.

- With the "Product providers" feature (See 1.9, page 31), two additional fields can be set during the entry:

  Provider: Provider of the supply.

  Commentary: Free text field, to enter various notes about the supply.

All the supply information is summed up at the end of the entry, on a confirmation page. A warning appears if the tank stock after the supply is not between the mininum limit and the tank capacity.

Tank supplies must be entered in chronological order.

7.2.1 New unit price computing

The unit price after a supply is computed using the "weighted average price" method:

$$p = (P_a * V_a + P_c * V_c)/(V_a + V_c)$$

with:

P_a: supply unit price

V_a: supply volume

P_c: tank stock unit price

V_c: tank stock at the supply date

V_c is computed by adding to the current stock, the volume of the transactions realized between the supply date and the current date. The result of this calculation may be viewed in the Previous tanks state page.

For this computing to be valid, all the transactions carried out before the supply must have been retrieved in GIR Titan-Hyperion

7.2.2   Tank supply cancellation

To cancel a supply, click on Cancel in the supply sheet. Only the last supply of each tank can be cancelled.

7.2.3   Transactions valuation

When a fuel transaction is retrieved from a CPU by GIR Titan-Hyperion, it is valuated with the tank stock unit price at the time of the processing. Hence, to correctly valuate all transactions, it is important to keep a correct chronology when entering tank supplies.

When entering a supply, all the transactions occuring before this supply must have been retrieved. If not, the stock used to compute the new unit price is wrong, and this price is wrong too.

A correct transaction valuation requires to conform to those two rules:

- Do a *Retr.-Send* dialogue before entering a supply.

- Enter a supply just after it has been made; i.e. before retrieving any transaction realized after.

If the second rule is not respected and that transactions made after the supply are processed before its entry, those transactions will be valued with the tank stock unit price that was before the supply.

Transactions revaluation

Transactions valuation can be changed after their creation. This can be useful for the following cases:

- To fix an erroneous supply

- To change the unit price of a tank stock

Transactions revaluation can be done via ▷ Config., Transactions management - Recomputing, Transactions revaluation. Select a tank, a unit price and a time range. All the transactions done during the selected time range on the selected tank will then be revaluated with the indicated price.

This form is preset with the current unit price of the stock, and a time range going from the last supply to the current date.

Example: Fixing an erroneous meter.

On 10/10/2004, a 0.8 euro/gal, 500 gal supply was entered for tank 1, whereas the real value was 5000 gal. The error is detected three days later. All the transactions retrieved between the 10/10/2004 and the 13/10/2004 have been valuated with an invalid unit price, computed from an erroneous supply. The correction requires two steps:

1. Cancel the erroneous meter, then enter a new one with the correct values. The unit price of the tank stock is now correct.

2. Click on Transactions revaluation, and select tank 1. The default values correspond to the required revaluation: all the transactions retrieved since the 10/10/2004 will be revaluated with the new unit price of tank 1 stock.

7.3 Fuel transaction modification

Fuel transaction modification allows to fix a possible error on a transaction. This screen can by accessed by clicking on Modify in a transaction sheet.

The following data can be modified:

- Volume, U.P. (Manager only)

- Vehicle, Driver, Milometer, Hour meter (Manager and standard user)

To modify a value:

1. Select the field to modify by clicking on % near this field

2. Enter the new value

3. Click on Validate

The transaction modification being an exceptional operation, only one field may be changed at a time. Moreover, a trace of the change is always kept in the transactions history: the original transaction appears with the *Cancelled* option.

When a meter value is not known, it is possible to delete this meter by letting the entry zone empty. The meter will then be set to *(N/A)*, and consumption computing will be transfered to the next transaction.

Covered distances, consumptions and meters are automatically recomputed when a modification is made.

Example: modifying a meter

We want to change the transaction T_1 to set its meter to 72600 mi:

	Date	Vehicle	Volume	Meter	Covered	Options
T_1	02/04/2004	1234ZZ69	59,00 gal	72800 mi	1100 mi	

1. Search transaction T_1 in the transactions history

2. Click on % to access the transaction sheet

3. Click on Modify

4. Click on % near the Meter (mi) field

5. Enter the new meter (in our example 72600)

6. Click on Validate

7. GIR Titan-Hyperion does the following processing:

   - Transaction T_1 is *Cancelled*
   - Covered distances, consumptions and meters are recomputed to create a transaction $T_1^{'}$
   - Transaction $T_1^{'}$ is added to the history

8. The sheet of the fixed transaction $T_1^{'}$ is displayed

The transactions history now contains an additional transaction:

	Date	Vehicle	Volume	Meter	Covered	Options
$T_1^{'}$	02/04/2004	1234ZZ69	59,00 gal	72600 mi	900 mi	
T_1	02/04/2004	1234ZZ69	59,00 gal	72800 mi	1100 mi	Cancelled

Chapter 8

Access management

8.1 Supervision

8.1.1 Punctual and forced control

Access points may be controlled from two different ways:

- Punctually: the access point is opened for a few seconds after having identified a driver.

- Under forced control: the access point is opened for a specified period, at a fix hour. The definition of the time slots during which the access will be opened is specified by forced zones.

Forced zones and access point control mode are defined by the installer of GIR Titan-Hyperion.

8.1.2 Supervision screen

The ▷ Supervision menu allows to visualize the state of access management CPUs. This screen displays the various access points defined on the CPU, and their state at the time of the supervision:

- *Available*: The access point is available for vehicles or drivers identification.

- *Out of zone*: The access point is currently in no valid zone and is inactive.

- *Forced*: The access point is currently in a forced zone, access is opened.

- *Forced (manager)*: The access point is opened because of a remote transaction from GIR Titan-Hyperion.

- *Forced stop (manager)*: The access point is in a forced zone, but opening has been stopped by a remote transaction from GIR Titan-Hyperion. Depending on the active zones, access remains available for the vehicles or drivers identification.

According to the configuration defined by the installer of GIR Titan-Hyperion, it may be possible to remotely control the access points using the software. When remote control is enabled, a Remote button appears near the access point picture.

Clicking on this button shows the operation selection screen. The various operations are *Punctual*, *Forced*, *Forced stop* (operations availability may differ according to the configuration).

Any remote control by GIR Titan-Hyperion generates a transaction, which can be assigned on this screen to a vehicle or driver. The Validate button launches the remote transaction, and Cancel goes back to the supervision screen.

8.2   States

Access management states are available by the ▷ Management menu:

Access transactions hist.:   History of all access transactions (allowed or denied) retrieved after a dialogue.

The (Not.) link allows to access directly to an history filtered on noteworthy transactions.

The following access transactions are considered as noteworthy:

- Access denied
- Access remotely launched
- Remotely forced stop
- Access realized in a vigilance zone (See 1.6, page 24).

Access transactions charts: Graphical visualization of access transactions, shown as histograms. This screen is similar to the one available for fuel transactions (See 5.1.3, page 65).

Chapter 9

Maintenances

9.1 Maintenance definition

GIR Titan-Hyperion allows to manage vehicles maintenances using three parameters:

- Milometer value

- Hour meter value

- Date

For each parameter, two limits can be defined:

- Alert limit: a message appears on the terminal when the vehicle is fueling.

- Blocking limit: the vehicle can't refuel as long as the maintenance has not been done. Blocking only occurs on fuel delivery CPUs: the vehicle is still authorized on access management CPUs. According to the configuration, the vehicle may also remain authorized on maintenance CPUs (e.g. oil delivery).

GIR Titan-Hyperion generates an event when a maintenance reaches the alert or blocking limit.

Before creating a maintenance, it is required to define its type in ▷ Misc, Maintenance types:

Short name, Full name: used to refer to a maintenance type in the software.

Alert message: Message displayed on the terminal when this maintenance type reaches its alert limit.

Blocking message: Message displayed on the terminal when this maintenance type reaches its blocking limit.

Nb. reminders: Number of reminders on the terminal before blocking the vehicle.

Once a maintenance type has been defined, a maintenance can be created under the Maintenances tab in the vehicle sheet. A maintenance is composed of four parameters for each criteria (milometer, hour meter, date):

Next: Next maintenance term.

Frequency: Maintenance theoretical frequency.

Alert delta: Difference between the value defined in Next and the alert limit.

Blocking delta: Difference between the value defined in Next and the blocking limit.

Alert delta and Blocking delta may be positive or negative. When a delta is negative, the associated limit will be situated *before* the term. When a delta is positive, the associated limit will be situated *after* the term. The picture below shows a maintenance with a negative Alert delta and a positive Blocking delta.
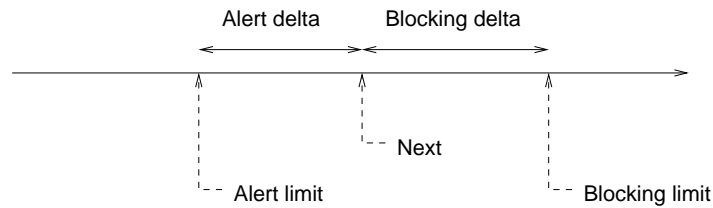
A maintenance state may be consulted via the Maintenances tab of the vehicle sheet. The various possible states are:

- *Alert*: The maintenance has reached its alert limit.

- *Blocking*: The maintenance has reached its blocking limit.

- *Invalid*: The maintenance is invalid, no alert or blocking limit has been defined.

- *Waiting*: The maintenance is valid, and has not reached any limit yet. This is the most common maintenance state.

- *Disabled*: The maintenance is disabled.

Alert, blocking and invalid maintenances appear in red in the General tab of the vehicle sheet.

9.1.1   Examples

- On 01/09/2004, vehicle *V* meter is 54 000 mi and it has to be drained off every 10 000 mi. Its last oil change was done on 01/06/2004 at 52 000 mi. We would like to be warned 500 mi before the term, but not to block the vehicle if the maintenance has not been made in time. We would declare the following maintenance:

Next meter	*62 000*
Frequency (mi)	*10 000*
Alert delta (mi)	*-500*
Blocking delta (mi)	*(Empty)*

Should we wish, in addition to the mileage criteria, that the oil change would be done at leat every year, we would add the following parameters:

Next date	01/06/2005
Frequency (month)	12
Alert delta (month)	-0,5
Blocking delta (month)	2

The Alert delta of -0,5 means that we want to receive an alert about 2 weeks (0.5 month) before the term, and the Blocking delta of 2 means that the vehicle must be blocked if the oil change has not been made two months after the term.

- For a new vehicle bought on 15/07/2004, we would like to define a MOT test, with a reminder one month before and a blocking at the term. The MOT test must be realized the first time 4 years after the purchasing date, then every two years. We would then choose 15/07/2008 as Next date, and 24 as frequency:

Next date	15/07/2008
Frequency (month)	24
Alert delta (month)	1
Blocking delta (month)	0

- It is possible to define a maintenance to be realized only once, by ticking the *Only once* option. For instance, to define the 50 000 mi next major service, with a reminder 1000 mi before the term and a blocking 2000 mi after, we would declare the following maintenance, ticking the *Only once* option:

Next meter	50000
Frequency (mi)	(Vide)
Alert delta (mi)	-1000
Blocking delta (mi)	2000

9.2 Maintenance entry

In GIR Titan-Hyperion, "maintenance entry" is the act of declaring that a maintenance has been realized. To enter a maintenance, click on Enter in the Maintenances tab of the vehicle sheet. According to the maintenance, up to three parameters may be required:

Date: Date to which the maintenance has been made.

Meter (mi): Value of the vehicle milometer when the maintenance is made (only if distance criteria are defined for this maintenance).

Meter (h): Value of the vehicle hour meter when the maintenance is made (only if hour criteria are defined for this maintenance). The value to enter is in hours, with a two-digits decimal precision.

When a new maintenance is entered, the Next (mi), Next (H) and Next date values are automatically recomputed from the defined frequencies.

The Last maintenances tab allows to view the last 10 maintenances of a vehicle. A complete history of the maintenances carried out may be consulted in ▷ Management, Maintenances hist..

9.3   Estimation of maintenances to do

The maintenances of a given vehicle may easily be consulted in the various tabs of the vehicle sheet. GIR Titan-Hyperion also allows to visualize all maintenances to be done on the fleet, in the menu ▷ Management, Maintenances to do.

This screen allows to view the maintenances to do from three different ways:

Currently: Maintenances to do currently.

Prediction (estimation): Maintenances to do before a given date, using an estimation of the vehicle activity during a reference period.

GIR Titan-Hyperion estimates what will be the future meter of a vehicle by the following formula:

$$K_{future} = \Delta_{ref} * (T_{future} - T_{current})/(T_2 - T_1) + K_{current}$$

with:

T_{future}: date to which the maintenances are predicted

$T_{current}$: current date

Δ_{ref}: distance covered during the reference period

T_1, T_2: dates defining the reference period

$K_{current}$: current meter value

The result of this calculation appears in the Predicted column, below the current meter value.

The Order by term option displays the date to which the vehicle will reach the alert or blocking limit, according to the estimation above. Maintenances are then sorted using this date.

Note: the accuracy of the prediction greatly depends on the choice of the reference period. It is advised to choose a period long enough (at least one month), during which the vehicle activity has been significant.

Prediction (range): Maintenance reaching their term in a given range.

Several filters are available. The State filter allows to display only alert, blocking, or other maintenances. When all boxes are checked, all the maintenances defined in the software are displayed. Other filters allow to limit the result to a maintenance type, to a department or to a vehicle.

The result screen shows vehicles for which a maintenance to do has been found, with the alert and blocking limits values.

Netscape: GIR Titan-Hyperion - Demo

File   Edit   View   Go   Window   Help

Back   Forward   Reload   Home   Search   Print   Security   Stop

Bookmarks   Location: http://localhost:8080/cgi-bin/twcgibin?p=4747&_sid=1060622341&_cmd=app&cmd=mtn_todo&=View&page=3&date=05-20-2005&r

GIR Hyperion

Home

Config.
Misc.

Vehicles
Drivers

Management

Manual trans.
Tanks state

Supervision
CPU comm.

[root]
demo

Disconnect

Maintenances to do

Prediction (valuation)
To do before: 05-20-2005
Reference period: 06-01-2004 - 06-30-2004
State: Alert, Blocking
Modify

Printable version
CSV file (V1 V2)

D01 Internal fleet VSC Commercial vehicles

Vehicle	Maintenance	Meter (mi.)			Meter (h)			Date			State
		Freq.	Current Predicted	Alert Blocking	Freq.	Current Predicted	Alert Blocking	Freq. (months)	Current Predicted	Alert Blocking	
177HR53	Drain	15 000	91 800 125 942	95 740 96 240	0.00			12.00	07-28-2004 05-20-2005	03-05-2005	■ Blocking
2248XR69	Drain	15 000	47 521 47 521	47 400 48 900	0.00				07-28-2004		■ Alert
4294NF51	Drain	15 000	81 091 81 091	79 830 81 030	0.00				07-28-2004		■ Blocking
5263JB92	TC		67 250		0.00			24.00	07-28-2004 05-20-2005	07-25-2004 09-24-2004	■ Blocking
5263JB92	Drain	15 000	67 250 75 126	79 100 81 100	0.00			12.00	07-28-2004 05-20-2005	05-11-2005 08-10-2005	■ Alert
9738PT05	TC		104 176		0.00			24.00	07-28-2004 05-20-2005	09-20-2004 11-20-2004	■ Blocking

9.4   Maintenance recomputing

Recomputing a maintenance means determinating its new state, using the new meters of a vehicle or the elapsed time. Therefore, if a maintenance reaches its alert limit on 18/05/2005, it will only be in an *Alert* state after the first recomputing made after 18/05/2005 at midnight.

Maintenances are recomputed:

- For each vehicle: each time a fuel transaction is processed for this vehicle.

- For all vehicles: in the first hour after GIR Titan-Hyperion startup, then every 24 hours.

Recomputing can also be launched manually via the menu ▷ Config., Transactions management - Recomputing, Maintenances recomputing. This operation is available to the manager only.

Chapter 10

Credits

10.1 Overview

GIR Titan-Hyperion allows to manage various credits types:

- Credits by vehicle, defined in ▷ Vehicles, Tanks

- Credits by department, defined in ▷ Misc, Departments credits. For TIP2-Vatersay CPUs, they can be either standard fuel credits or monetary credits.

- Credits by activity, defined in ▷ Misc, Activities credits

For every type, a credit is defined by the following data:

Product: Product for which the credit is deducted.

Initial credit: Initial credit value

Credit: Current credit value

Period: Credit reset period, defined in ▷ Misc, Credit periods

Credit type: Blocking or non-blocking credit. For blocking credits, it becomes impossible to take fuel when the credit value reaches zero. On the contrary, non-blocking credits are decreased indefinitely, without preventing fueling.

Credits are deducted when processing a transaction with the corresponding product and when the following condition is true:

- Credits by vehicle: The *Vehicle* option is enabled in the Credit options field of the vehicle's tank.

- Credits by department: The *Department* option is enabled in the Credit options field of the vehicle's tank.

- Credits by activity: The activity code has been entered in the transaction.

10.2   Monetary credits

Monetary credits require a unit price to be defined for the concerned products.

This price can be defined under the Products tab in the ▷ Misc, Departments credits menu. It is totally independent from the tank stock's unit price.

10.3   Credit periods

▷ Misc, Credit periods

Credit periods allow to define how and when credits are recomputed.

Short name, Full name: Names used to refer to a credit period in the software

Period: Time unit according to which the Day/Hour and Frequency fields will be interpreted. It can be *Daily*, *Weekly*, *Monthly* or *Yearly*.

Day/Hour: Numerical value interpreted according to the defined period:

Period	Day/Hour meaning
Daily	Reinitialization hour (from 0 to 23)
Weekly	Reinitialization day in the week (from 1 for Monday to 7 for Sunday)
Monthly	Reinitialization day in the month (from 1 to 31)
Yearly	Reinitialization month (from 1 for January to 12 for December)

Carry: The carrying over method to use when a credit is reinitialized. The various methos are:

- *None*: The credit is always reinitialized to the initial credit.

- *Unlimited*: The credit is increased by the amount of the initial credit, without any upper limit.

- *Max. 2 * init. credit*: The credit is increased by the amount of the initial credit, up to twice this value.

Frequency: Allows to define the time limit between two credit reinitializations. It depends on the selected period. The table below shows the time between two reinitializations for a frequency value F:

Period	Time between two reinitializations
Daily	F days
Weekly	F weeks
Monthly	F months
Yearly	F years

Options:

- *Event on depleted*: Generates an event (See 5.3, page 68) when the credit is depleted.

10.4 Examples

Periods:

- (1) Credit reinitialized every month, on the first day of the month:

  Period: *Monthly*, Day/Hour: *1*, Frequency: *1*

- Credit reinitialized every three months, on the 15th day of the month:

  Period: *Monthly*, Day/Hour: *15*, Frequency: *3*

- Credit reinitialized every two weeks, on Monday:

  Period: *Weekly*, Day/Hour: *1*, Frequency: *2*

- (2) Credit reinitialized every year, in June:

  Period: *Yearly*, Day/Hour: *6*, Frequency: *1*

- Credit reinitialized every morning at 6 a.m.:

  Period: *Daily*, Day/Hour: *6*, Frequency: *1*

Credits definition:

- Definition of a vehicle 100 gallons per month gasoil credit: Enter the following data in the Tanks tab of the desired vehicle, for the *GOI* product:

  Initial credit: 100 gal

  Credit: Current credit value (e.g.: 100 gal if data is entered the first day of the month, 50 gal if it is in the middle of the month...).

  Period: The period (1) defined above.

  Credit options: *Vehicle*

- Definition of a department 10 000 gallons per year gas credit: Enter the following data in ▷ Misc, Department credits:

  Initial credit: 10 000 gal

  Credit: Current credit value.

  Period The period (2) defined above.

  Moreover, the *Department* option will have to be enabled in the gas tank of all the department's vehicles on which the credit should be deducted.

10.5 Credits recomputing

Credits recomputing checks all the credits defined in GIR Titan-Hyperion for a possible reinitialization. This process is launched during the first hour after the application startup, then every hour. It is also possible to launch it manually (manager only), using the menu ▷ Config., Transactions management - Recomputing, Credits recomputing.

Thus, if a credit is defined to be reinitialized every morning at 6 a.m., GIR Titan-Hyperion guarantees that it will be reinitialized between 6 a.m. and 7 a.m. To ensure that all CPUs will be updated with the new values, an automatic dialogue should be defined every morning at 7 a.m.

For daily and weekly periods, reinitialization is made during the first hour of the selected day (i.e. between midnight and 1 a.m.). For yearly period, reinitialization is made during the first hour of the first day of the selected month.

If GIR Titan-Hyperion is not running when a credit must be reinitialized, the reinitialization will occur during the next recomputing: during the first hour after the application startup, or during a manually launched recomputing.

Should the application not be runned during several periods in which a credit should have been reinitialized, only one reinitialization will occur for the first recalculation.

Example:

A credit is defined to be reinitialized every day at 6 a.m., with an *Unlimited* carrying over method, and an initial value of 50 gal. The vehicle doesn't make any fueling during two days. On 10/10/2004 at 8 p.m., the credit is 10 gal and GIR Titan-Hyperion is stopped.

The application is restarted on 12/10/2004 at 8 a.m. In the following hour, a credit recomputing is launched and the credit is reinitialized to 60 (10 + 50) gallons.

If the application had been runned at least once between the 11/10/2004 at 6 a.m. and the 12/10/2004 at 6 a.m., or if it had continued to run during all this time range, the credit would have been reinitialized to 60 gal on 11/10/2004, then to 110 gal on 12/10/2004.

Chapter 11

Invoicing

11.1 Configuration

▷ Misc, Invoices

An invoice profile is made of the following elements:

- General profile configuration (**Parameters** tab)

- Products to invoice (**Parameters** tab)

- Providers definition (**Providers** tab)

- Customers definition (**Customers** tab)

11.1.1 General parameters

Name: Profile name

Label: Text to display on the final document (ex: "Invoice", "Bill", . . .)

Options:

- *Transactions detail*: Adds a transactions history to the generated document.

- *Disable auto. num.*: Disables invoice number auto-generation. This forces to manually enter a invoice number before the document generation.

- *Mandatory provider*: Makes the provider selection mandatory in the invoice generation form.

- *Disable global customer*, *Disable global provider*: Hides the global customer or provider in the generated documents.

- *Protected products*: By default, prices to use for products to invoice can be defined by any user allowed to generated an invoice. This option restricts prices definition to the manager only.

11.1.2   Products

Product: Product to invoice.

U.P. charged: Defines how the invoiced unit price will be computed:

- *Transaction*: Uses the U.P. at which transactions were valued.
- *Purchase*: Uses the U.P. of tank supplies.

V.A.T.: V.A.T. rate

Coef. disp., Unit disp.: Coefficient and unit to use to display volumes. By default, the coefficient is 1 and the unit is the same as in the application.

Coef. eq., Unit eq.: Coefficient and unit to use to display an equivalent volume, on the "Total" row.

11.1.3   Customers and providers

These tabs define customers and providers addresses to use when generating invoices. Addresses positions on the document are defined by the geometric parameters in the Manager config.. A provider can be a site or a region, and a customer can be a department or a direction.

The "global provider" regroups all the sites that are not explicitly defined in the invoice profile. Thus, if three sites A, B and C are declared in the software, with only the B site defined under the Providers tab, the global provider is the union of A and C sites.

Similarly, the "global customer" regroups all the departments that are not explicitly defined under the Customers tab. Those two special elements can be disabled using the dedicated options under the Parameters tab.

When generating invoices, GIR Titan-Hyperion uses all customer-provider couples for which transactions are found.

11.2   Invoices generation

▷ Management, Invoices

11.2.1   Form

The invoices generation form contains the following fields:

- First screen:

  Start num.: Invoice number. When multiple invoices are generated, this number is incresased for each new invoice.

  Period: Time range to invoice.

  Customer: Filter on a specific customer.

  Provider: Filter on a specific provider.

- Second screen: for each product, invoiced unit price settings. Multiple choices are available depending on the configuration. For each one, an increase can be defined, either with a percentage or with a fixed value.

- *Exact*: Uses the U.P. at which each transaction was valued, as shown in the transactions history.

- *Average by cust.*: Computes the average valued U.P. for each customer on the selected period (this will give the same U.P. to all the transactions on a given customer).

- *Global average*: Computes the average valued U.P. on the selected period (same U.P. for all customers).

- *Weighted average*: Computes the average U.P. of all the supplies on the selected period, weighted by the volume of each supply.

- *Max. value*: Uses the maximum U.P. of all the supplies on the selected period.

- *Last purchase*: Uses the last supply U.P. for each provider.

- *Forced to*: Specifies entirely the U.P. value to use.

11.2.2 Result

Once all desired parameters have been entered in the form, click on Next to display the result page.

It contains the following elements:

- Modify button: goes back to the form to modify the selected parameters.

- View all button: view all generated invoices on the same page.

- Printable version button: printable version of all the generated invoices (each invoice on a separate page).

- Generate all files button: generates one file per invoice in `data\dto\invoices`. For traclient files, the file name has the pattern `invP-NNNNNN-YYYYMMDD` where `P` is the invoice profile, `N` the invoice number, and `YYYYMMDD` the current date. For transcodings, the transcoding name is added before the invoice number.

- For each customer-provider couple:

  - View button: consult the invoice generated for this customer-provider couple.

  - Printable version button: printable version of this invoice.

  - Download button: download the traclient or transcoding file, containing all the transactions used for the invoice.

Chapter 12

Data import and export

12.1 Oil disks import

Oil disks import is configured in ▷ Misc, Oil cos, which defines the various data formats to import. Once this configuration is done, import may be launched in ▷ Management, Oil disks import.

12.1.1 Configuration

The Oil cos table contains the following fields:

Short name, Full name: Names used to refer to an oil co. in the software.

File Id: Identifier user to guess the name of the file to import: `data\dti\petro-<id>.txt`, or any file in the directory `data\dti\petro-<id>\`.

The other fields define the position and the length of the various values to read in the file. The Length filter allows to only read lines of a given length, the Type filter and Client filter allow to only read lines for which the type or client field matches the specified value.

Some options are dedicated to drivers and vehicles assignment:

Vehicle word: Position of the registration number in the Registration field of the vehicle. Words are separated by one or more spaces, and first word is numbered as 1. When this value is zero, the whole Registration field is used as registration number.

Ex: Registrations were entered in GIR Titan-Hyperion in the form "ABCD 1234ZZ69", where "ABCD" is a fleet code and "1234ZZ69" the registration number of the vehicle. To import an oil file in which vehicles are refered by this number, we would define 2 as Vehicle word to use the second word in the Registration field during the import.

Driver word: Position of the code in the driver Ref. field. Usage is the same as for Vehicle word.

Preprocessing:

- *Adjust tabs*: When a tab character is found in the input file, the position counter is increased up to the next multiple of 8. This allows to process a file where fields are separated with tabs as if it were a fixed format.

Options:

- *Ignore leading zeroes*: Ignores leading zeroes in the registration number. When this option is enabled, "123ZZ69" and "0123ZZ69" are processed the same way.

To be valid, a configuration must define at least a date, a vehicle, a product and a volume. Common oil disks formats are available in appendix to this document.

12.1.2   Products assignment

When importing oil disks, the conversion from the product code read in the file to a product defined in GIR Titan-Hyperion is done using the Oil products subtable, available in the tab of the same name, in the oil co sheet. Number defines the product code, and Product or Pump defines the product or pump to assign to the transaction when the code is read in the file. If both the product and the pump are set, the pump is used.

Finally, the Options field indicates which options should be set in the imported transaction.

12.1.3   Vehicles assignment

Vehicles can be assigned either by oil card or by registration. Oil cards are defined in the Oil cards table, available in the ▷ Misc menu.

Number defines the card number read, and Vehicle defines the vehicle to assign to the transaction when this number is read for the corresponding Oil co.

When importing, assignment is first tried using oil cards, then, if no match was found, using registration.

The *H meter* option allows to swap the milometer and the hour meter for a given vehicle.

Oil cards can also be used to define drivers assignment.

12.1.4   Import

When the configuration of an oil format is completed, select it in the menu ▷ Management, Oil disks import, then click on Next. In the file list that appears, select a file by clicking on %.

A first analysis of the file is then displayed. It allows to view what will be imported. At this point, no transaction has been imported in the database.

The file analysis simulates an import, and classifies input lines into various categories:

- Processed: Number of lines that will be imported as a fuel transaction.

- Filtered: Number of lines ignored because of filtering settings.

- Doubles: Number of lines that define an already imported fuel transaction.

- Not processed: Number of lines that cannot be imported. The right column indicates the causes of these errors.

- Warnings: Warning messages, generally corresponding to pieces of information in the input file which could not be decoded, without preventing an import. For instance, an activity code for which no assignment was found will appear here.

The total of the first four categories is the number of non-empty lines in the file.

Below this table, the dates of the first and last transactions that will be imported are displayed. To import the transactions into the database, click on Yes.

If an error occurs, it is possible to erase the transactions imported in a given time range, in order to launch a new import. This feature can be accessed in the lower part of the screen, in the ▷ Management, Oil disks import menu.

12.1.5   Note on meters

Meters entered during an oil-card fueling are not always controled, contrary to those entered on CPU (See 4.1, page 59).

Therefore, computing covered distance and consumption using such meters is not reliable and it might be interesting to disable it. To do so, just check the *Add-on* option for all the Oil products defined in this import.

Transactions imported under this configuration will be considered as add-ons, for which consumption is reported on the next fueling.

Moreover, updating the vehicle meter on oil disk import can be configured in the Manager config. (See 1.9, page 31).

12.2   Traclient export

The traclient file registers transactions processed by GIR Titan-Hyperion. It is meant to be used in other applications.

There are two ways to generate a traclient file:

Automatically : by selecting the desired format in ▷ Misc, Manager config..
Several automatic exports are available:

- Live: transactions are added to the file as they are processed.
  File name: `data\dto\traclien.dat`

- Daily: a file containing all transactions done on the eve is generated at the hour defined in the Manager config..
  File name: `data\dto\traclien-AAAAMMJJ.dat`

- Monthly: a file containing all transactions done during the last month is generated at the day defined in the Manager config..
  File name: `data\dto\traclien-AAAAMM.dat`

In formats anterior to C2 and F4, these exports do not take into account the possible modifications on transactions after their processing.

Types of transactions to export are defined in ▷ Misc, Manager config.. The export can also be limited to specific products, in ▷ Config, CPU, tanks tresholds, ref. and zones management (See 1.11, page 37).

On demand : in ▷ Management, Traclient export. Data is written in `data\dto\traclient.dat`.

It is possible to filter transactions on a given time range, on their origin (on CPU, manual or imported), and on their site, direction or department.

There are several formats of traclient files:

- F1, F1-Ref: Fixed-length text format with 2-figures year.

- F2, F2-Ref: Fixed-length text format with 4-figures year.

- F3, F4: Fixed-length text format

- C1, C2: CSV format.

Formats F1, F1-Ref, F2, F2-Ref are present for compatibility with existing software. To develop a new application, it is better to use the more complete F4 or C2 formats.

In fixed-length formats, each transaction is a fixed-length line terminated with CRLF. In CSV formats, each transaction is a line, containing multiple fields separated by ';' and terminated by CRLF. Text fields are delimited by double quotes (`"`).

The various traclient formats are detailed in appendix to this document.

In addition to fuel transactions, traclient export can be used for the following data. Files are always created in the directory `data\dto\`.

Format	File	Data type
C2/F4	`supclient.dat`	Supplies
M1/N1	`fsmclient-AAAAMM.dat`	Monthly fuel sums
M1/N1	`fsmclient-AAAAMMJJ.dat`	Daily fuel sums
W1/X1	`accclient.dat`	Access transactions

12.3   Transcodings

Transcodings allow to automatically convert traclient files into other formats. They may appear under two forms:

- Internal transcodings: they are directly built into GIR Titan-Hyperion, and can be enabled by a simple selection in a list.

- External transcodings: they are plug-ins to install in the directory `data\libexec\`.

Transcodings are defined in ▷ Misc, Transcodings, and are enabled by the installer of GIR Titan-Hyperion in Installer config.. They allow to export the different transactions types, either live or on demand.

They are configured as follows:

- Internal transcodings:

  Type: Name of the transcoding to use.

- External transcodings:

  Name: Name of the transcoding plug-in. If `Name` is set to "abc", the searched plug-in will be `data\libexec\transcod-abc.exe`

  Input format: Traclient format to provide as input to the transcoding plug-in. See the specific transcoding documentation to know which format to use.

- Common parameters:

  Fuel live options:

  - *On CPU*: Enables live transcoding for transactions retrieved from CPUs.
  - *Manual*: Enables live transcoding for manual transactions.
  - *Imported*: Enables live transcoding for imported transactions.
  - *Tank supplies*: Enables live transcoding for tank supplies.

Appendix A

Licence agreement

End User License Agreement for Hyperion software

IMPORTANT - READ CAREFULLY

This End User License Agreement (the "EULA") constitutes a legal
contract between you (physical or moral person) and the klervi
company, regarding the Hyperion software product ("Software"), as
well as any document, software and documentation that deals with it.

INSTALLING, COPYING, OR USING THIS SOFTWARE IN ANY OTHER WAY MEANS THAT YOU
ACCEPT TO CONFORM TO ALL THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT.

Should you not accept the terms and conditions of this Licence
Agreement, you will not be authorized to install or use the Software.
Should you accept the terms and conditions of this Licence Agreement,
and should you have bought the Software, the klervi company allows
you to use the Software in the way described below.

LICENSE CONCESSION

The present Software remains the exclusive property of the klervi
company. It is distributed by the GIR company, and usable with
Windows 9x, 2000 or XP, on compatible PC.

The present E.U.L.A. authorizes you to install and use the
Software program.

RIGHTS AND LIMITATIONS

The present Software and its documentation, as well as any copy,
remain the exclusive property of the klervi company. In France,
the Software is protected by the Intellectual Property Code and
abroad by the international conventions on the author's rights.
In consequence, you should act towards this Software as if it was
any other material submitted to the author's rights (as a book for
instance), with only one difference, that you are allowed to make a
copy of this Software, but only for backup or filing purposes.
You are liable for the physical security of this copy.

You can only yield the Software and its documentation to another
person if you have the previous written agreement of the klervi
company, and on a permanent way, on the condition that you do not
keep any copy and that the beneficiary accepts the stipulations of
the Licence Agreement.

To the exception of the cases expressly previewed in the Licence
Agreement, the transfer, renting, hire purchase, loaning, copying,
modification, translation, sub-licence, part-time sharing, transmission
or electronic reception of the Software, of its base or its concerned
documentation are not authorized.

You acknowledge that the Software under the source code form remains
a commercial secret of the klervi company, and for this reason, you
accept to not modify the Software, nor try to take it to pieces, nor
discompile or disassemble this Software, except and only when such
activity is expressly authorized by the applicable legislation despite
of this restraint.

You commit yourself to not develop or commercialize the Software, object
of the present contract, or products that could be in competition with it
and/or based on one or more elements that have been extracted from it.
It is forbidden to reproduce, translate, adapt, fix, or modify the Software
as well as its documentation, even when it should be necessary in order
to allow the use of this Software in conformity with its destination.

In application to the L 122-6-1 article of the intellectual property code,
the klervi company keeps the right to correct mistakes, and to establish
the particular modalities to which will be submitted the actions defined
above.

The reproduction of the Software code or the translation of the code
form are forbidden even when they are essential to obtain the necessary
information to the interoperability of the Software with other ones,
the klervi company keeps this information available to the user.

We remind you that the violation of one of the Software Authors' rights
is a counterfeiting crime and that any counterfeiting is punished in France
by the L.335-2 article of the intellectual property code and is liable
to a 2 years penalty and a 150 000 EUR fine.

UPDATES

Should you have installed an updated version of the Software, it
constitutes an only one and unique product with the Software that
you put up to date. You can use or transfer the new Software version
only according to the Licence Agreement dispositions. Any new version
of the Software comes to take over from the previous version and remains
submitted to the present Licence Agreement, with modification reserve
of this Agreement.

NOMINATIVE INFORMATION

The Software possesses a data base which can constitute an
automatized nominative information treatment, according to articles
16 and 17 of the dated 6th January 1978 Law concerning the information
technology, files and liberties, and which should be the object of a
declaration. In consequence, we remind you that you are liable for
your data file, and that you are compelled to declare it to the
National Commission for Information Technology and Civil Liberties
before creating it.

RESPONSIBILITY LIMIT

Up to the maximal evaluation authorized by the applicable legislation,
the klervi company can in no way be responsible for any special,
accidental, indirect or consecutive damage, of any nature (like an
explotation loss, a data loss or any other financial loss) resulting
from the use or from the impossibility to use the product, or from
the material or the impossibility to supply support services, even
if the klervi company has been expressly informed of the possibility
of such damages.

HIGH RISKS ACTIVITIES

The Software does not admit mistakes and is not designed or
manufactured for real time control gears that work in dangerous
environments and which require a breakdown resistance, as for
nuclear sites, aerial navigation or communication systems of air
traffic control, life assistance appliances or arms systems, for
which a Software defect could involve death, important injuries or
important physical or environmental damages (Hish risks activities).
The klervi company does not commit itself in anything particularly
in no explicit or implicit guarantee conformity or in High Risks
Activities adequation.

GENERAL DISPOSITIONS

The present Licence Contrat will be governed and interpreted in
conformity with the French Law. In case of dispute, both litigants
commit themselves to look for an amicable solution. By default, only
the French Court will be entitled to resolve this dispute. The klervi
company reserves its rights on every non expressly granted rights. The
klervi company also reserves the right to modify this Agreement without
any previous notification. If one of the stipulations of the Agreement
is considered null and void or unfounded, it will be qualified as non
written and will not involve the nullity of the other stipulations.

Appendix B

Traclient files format

F4 version - Fuel transactions

Position	Length	Description
000-018	19	Date (format: `"dd/mm/yyyy␣hh:nn:ss"`)
019-038	20	Vehicle ref.
039-068	30	Registration
069-088	20	Vehicle type ref.
089-108	20	Direction ref.
109-128	20	Department ref.
129-148	20	Driver ref.
149-178	30	Driver name
179-198	20	Driver direction ref.
199-218	20	Driver department ref.
219-238	10	Odometer (km)
239-248	10	Covered distance (km)
249-258	06	Instant consomption (cL/100 km)
259-264	06	Reference consumption (cL/100 km)
265-270	10	Hour meter (1/100 H)
271-280	10	Covered hours (1/100 H)
281-286	06	Instant consumption (cL/H)
287-292	06	Reference consumption (cL/H)
293-312	20	Site ref.
313-332	20	Tank 1 ref.
333-338	06	Tank 1 ratio (percentage * 100)
339-348	20	Tank 2 ref.
349-354	06	Tank 2 ratio (percentage * 100)
355-374	20	CPU ref.
375-394	20	Pump ref.
395-414	20	Product ref.
415-424	10	External volume (cL)
425-434	10	External price (cents)
435-444	10	Internal volume (cL)
445-454	10	Internal price (cents)
455-464	10	Unit price (cents)
465-470	06	Duration (seconds)
471-490	20	Activity ref.
491-500	10	Specific code
501-516	16	Options (see the options fields for format C2 - fuel)

C2 version - Fuel transactions

Position	Description
01	Date (format: `"dd/mm/yyyy␣hh:nn:ss"`)
02	Vehicle ref.
03	Registration
04	Vehicle type ref.
05	Vehicle type name
06	Vehicle direction ref.
07	Vehicle direction name
08	Vehicle department ref.
09	Vehicle department name
10	Driver ref.
11	Driver name
12	Driver first name
13	Driver direction ref.
14	Driver direction name
15	Driver department ref.
16	Driver department name
17	Odometer (km)
18	Covered distance (km)
19	Instant consomption (cL/100 km)
20	Reference consomption (cL/100 km)
21	Hour meter (1/100 H)
22	Covered hours (1/100 H)
23	Instant consumption (cL/H)
24	Reference consumption (cL/H)
25	Site ref.
26	Site name
27	Tank 1 ref.
28	Tank 1 name
29	Tank 1 ratio
30	Tank 2 ref.
31	Tank 2 name
32	Tank 2 ratio
33	CPU ref.
34	CPU name
35	Pump ref.
36	Pump name (CPU name + pump name)
37	Product ref.
38	Product name
39	External volume (cL)
40	External price (cents)
41	Internal volume (cL)
42	Internal price (cents)
43	Unit price (cents)
44	Duration (secondes)
45	Activity ref.
46	Activity name

47	Specific code
48	Options: 16 characters string, where each character is space when the option is not set, or the character below when it is set: 1: '+' over-consumption 2: '-' under-consumption 3: 'I' imported transaction 4: 'M' manual transaction 5: 'P' attendant 6: 'G' remote 7: 'E' external 8: 'A' cancelled 9: 'N' replacement of a cancelled transaction 10: 'K' renewed meter 11 to 16: unused. Example: Over-consuming attendant transaction: "+␣␣␣P␣␣␣␣␣␣␣␣␣␣␣␣"

For transactions with the 'A' option, the volume exported is the opposite of the volume of the original transaction. 'A' and 'N' options can be set only in live exports. In that case, the transactions are successive in the file.

F4 version - Supplies

Position	Length	Description
000-018	19	Date (format: "dd/mm/yyyy␣hh:nn:ss")
019-038	20	Tank ref.
039-058	20	Product ref.
059-078	20	Site ref.
079-098	20	CPU ref.
099-108	10	Supply volume (cL)
109-118	10	Total price (cents)
119-128	10	Unit price (cents)
129-146	16	Options: see the options fields of the C2-Supplies format

C2 version - Supplies

Position	Description
01	Date (format: "dd/mm/yyyy␣hh:nn:ss")
02	Tank ref.
03	Tank name
04	Product ref.
05	Product name
06	Site ref.
07	Site name
08	CPU ref.
09	CPU name
10	Supply volume (cL)
11	Total price (cents)
12	Unit price (cents)
13	Options: 16 characters string, where each character is space when the option is not set, or the character below when it is set: 1: 'B' on cpu 2: 'A' cancelled 3 to 16: unused

A cancelled supply is identical to the original one, with an opposite volume.
The 'A' option can be set only for live exports.

M1 version

Position	Length	Description
000-000	1	Type ('V': vehicle, 'U': driver)
001-030	30	Registration (vehicles only)
031-060	30	Name (drivers only)
061-080	20	First name (drivers only)
081-100	20	Ref.
101-101	1	Identification ('B': badge, 'C': code, 'I' initialization, 'P': code via attendant)
102-111	10	Badge type
112-121	10	Badge number
122-131	10	Badge code
132-141	10	Code
142-151	10	Secret code
152-152	1	Transaction type ('I': internal, 'E': external)
153-172	20	Product ref.
173-180	8	Transactions count
181-196	16	Volume (cL)
197-212	16	Total price (cents)
213-222	10	Total covered distance (km)
223-238	16	Distance consumption volume (cL)
239-248	10	Last odometer (km)
249-258	10	Total covered hours (1/100 h)
259-274	16	Hour consumption volume (cL)
275-285	10	Last hour meter (1/100 h)
286-385	100	Unused (reserved for future use)

The file contains vehicles, then drivers, sorted alphabetically.

For each vehicle, there is at least one line per product authorized to the vehicle. Each product can potentially have two lines, when external transactions have been made. Thus, a vehicle authorized on products 'GOI' and 'FUE', that has made internal fuelings on both products, as well as an external GOI fueling, will be on 3 lines in this file.

For each driver, this file contains at least one line per product defined in the software.

N1 version

Position	Description
01	Type ('V': vehicle, 'U': driver)
02	Registration (vehicles only)
03	Name (drivers only)
04	First name (drivers only)
05	Ref.
06	Identification ('B': badge, 'C': code, 'I' initialization, 'P': code via attendant)
07	Badge type
08	Badge number
09	Badge code
10	Code
11	Secret code
12	Transaction type ('I': internal, 'E': external)
13	Product ref.
14	Transactions count
15	Volume (cL)
16	Total price (cents)
17	Total covered distance (km)
18	Distance consumption volume (cL)
19	Last odometer (km)
20	Total covered hours (1/100 h)
21	Hour consumption volume (cL)
22	Last hour meter (1/100 h)
23-32	10 Unused (reserved for future use)

See the M1 format for more information on the file content.

W1 version - Access transactions

Position	Length	Description
000-018	19	Date (format: `"dd/mm/yyyy␣hh:nn:ss"`)
019-038	20	Vehicle ref.
039-068	30	Registration
069-088	20	Vehicle type ref.
089-108	20	Vehicle direction ref.
109-128	20	Vehicle department ref.
129-148	20	Driver ref.
149-178	30	Driver name
179-198	20	Driver direction ref.
199-218	20	Driver department ref.
219-219	01	Access result (see X1 format)
220-239	20	Site ref.
240-259	20	CPU ref.
260-279	20	Access set ref.
280-299	20	Access type ref.
300-315	16	Options: see the options field of the X1 format

X1 version - Access transactions

Position	Description
01	Date (format: `"dd/mm/yyyy␣hh:nn:ss"`)
02	Vehicle ref.
03	Registration
04	Vehicle type ref.
05	Vehicle type name
06	Vehicle direction ref.
07	Vehicle direction name
08	Vehicle department ref.
09	Vehicle department name
10	Driver ref.
11	Driver name
12	Driver first name
13	Driver direction ref.
14	Driver direction name
15	Driver department ref.
16	Driver department name
17	Access result, 1 ascii character: 'A': authorized, 'R': denied, 'Z': denied/zone
18	Site ref.
19	Site name
20	CPU ref.
21	CPU name
22	Access set ref.
23	Access set name
24	Access type ref.
25	Access type name
26	Options: 16 characters string, where each character is space when the option is not set, or the character below when it is set: 1 to 16: unused

Appendix C

Text e-mails format

The first four lines are the same for all messages, the following lines depend on the message type.

Line	Description
1	Application name and version
	e.g. "GIR Titan-Hyperion␣1.2.10"
2	Customer name
3	Empty line
4	Message type, of the form "Type:␣<Type>-<Subtype>"

Messages types and their associated subtypes are:

- ALRT: Alert

  - STOC: Tank stock
  - CONS: Over-consuming
  - PUMP: Pump blocking or unblocking
  - MTNC: Maintenance
  - EXPR: Authorization expiry or credit depleted
  - WTCH: Zone watch
  - LINK: CPU link error
  - IBDG: Invalid badge

- REPT: Periodical report

  - Not implemented, no subtype

The next lines depend on the message type, they are of the form "<Field>:␣<Value>" where Field is the field name on 4 characters, and Value a variable length text.

C.1   Alert messages

Field	Description
Tphr	"<Type name> <Subtype name>"
	Type name: fixed length, 10 characters
	Subtype name: fixed length, 20 characters
Date	Date under the format "dd/mm/yyyy␣hh:nn:ss"
Site	"<Ref>␣<Short name>␣<Full name>"
	Ref: fixed length, 20 characters
	Short name: fixed length, 10 characters
	Full name: variable length, at most 30 characters
Dire	"<Ref>␣<Short name>␣<Full name>"
	Ref: fixed length, 20 characters
	Short name: fixed length, 10 characters
	Full name: variable length, at most 30 characters
Serv	"<Ref>␣<Short name>␣<Full name>"
	Ref: fixed length, 20 characters
	Short name: fixed length, 10 characters
	Full name: variable length, at most 30 characters
Afct	Assignement, at most 21 characters
	For sites: Short name of the site
	For directions: Short name of the direction
	For departments: Short names of the direction and department
Mesg	Text message, at most 80 characters

C.2   Report messages

Field	Description
Mesg	"Not␣implemented"