GIR Titan-Hyperion

System prerequisites

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

Version 1.0-9, may 2011

The contents of this document are given for information only. Klervi is in no way liable for any mistake that could have slipped into the text.

Contents

Introduction

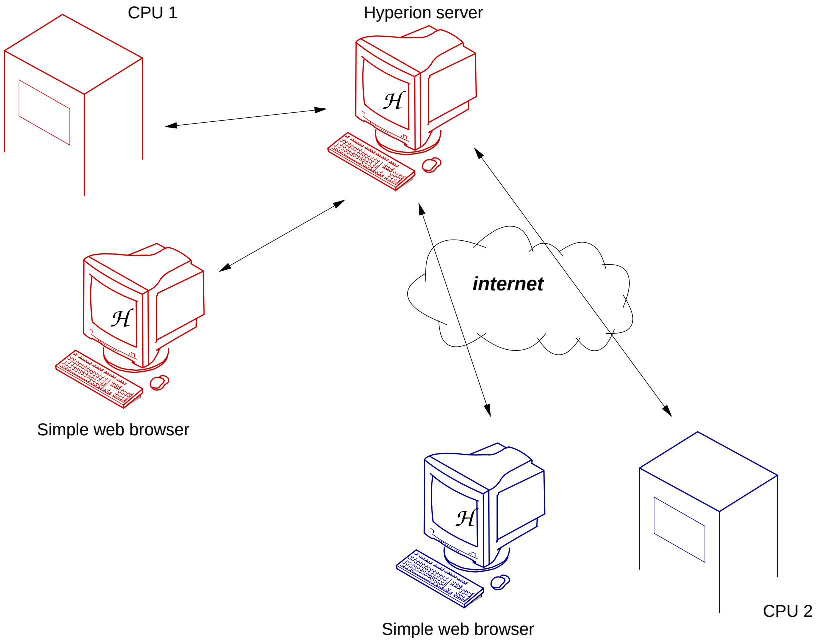
GIR Titan-Hyperion is a fleet management software. It covers fuel delivery, assignment of vehicles to drivers, vehicles maintenances, access to the station or to buildings, vehicles keys delivery.

GIR Titan-Hyperion includes CPU communication, data edition, and states consultation features.

Chapter 1

Generalities

1.1 Overview



The software GIR Titan-Hyperion is installed on a single computer, which will be refered in this document as “Hyperion server”. Hyperion is used through a simple web browser.

Communication with CPUs always occurs between the Hyperion server and CPUs, independently of the computer on which the web browser is installed.

Hyperion can be installed:

- On a server (dedicated or not): multiple workstations can access the application with a simple web browser.

- On an isolated workstation: the application is still usable by accessing the web server through the loopback interface (127.0.0.1).

- On a networked workstation: the application can be used on the computer where it is installed as well as on other computers on the same network.

Under Windows, an icon located near the clock in the taskbar makes it easier to access Hyperion from the computer on which it is installed. A double click on this icon directly launches a web browser with the appropriate location.

1.2 Database

Hyperion is written using C language, and uses an ISAM database format. It includes an automatic backup tool which can be configured to run at specific hours (See “Automatic schedules” in the user manual).

The size of a backup file varies from 500 KB to several MB (about 3 MB for a database with 1000 vehicles and 100 000 transactions).

Data export from Hyperion can be achieved in several ways:

- Fuel transactions export, “live” or on demand, in a traclient file. This file can be either a fix-length text file, or a CSV file (See 1.5, page 9).
- Export of any table directly from the user interface, by downloading a CSV file.
- Customized export, by selecting the fields to export in a script file.

Data import from another database is not possible directly, but is part of GIR’s services.

1.3 Minimal configuration

The Hyperion server requires the following minimal hardware configuration:

- Intel Pentium III or equivalent processor.
- 250 MB available on hard drive.
- 128 MB of RAM available for the application.
- Operating system:
  - Microsoft Windows 2000, XP, 2003, Vista, 7 or 2008 Server (Microsoft Windows 95, 98 and Me are also supported)
  - GNU/Linux kernel 2.2, 2.4 or 2.6

The client workstations must have a web browser:

- Microsoft Internet Explorer version 6 or above
- Mozilla Firefox version 1 or above
- Google Chrome, Apple Safari or any other web standards compliant browser.

1.4 Storage capacities

1.4.1 TIP1-Pabbay CPUs storage capacities

- unlimited number of products
- 3000 vehicles (prompt for odometer or hour meter is configurable for each product)
- 3000 drivers
- 2500 transactions (all transactions types included)
- 500 activities
- 100 messages

1.4.2 TIP2-Vatersay CPUs storage capacities

- unlimited number of products
- 15000 vehicles
- 15000 drivers
- 2500 transactions (all transactions types included)
- 1000 activities
- 100 messages

1.4.3 Key boxes storage capacities

Key boxes are remotely controlled by Hyperion and do not store any data.

1.4.4 GIR Titan-Hyperion storage capacities

- 15000 vehicles
- 15000 drivers
- 1000 activities
- 100 messages
- 150 000 fuel transactions
- 150 000 access transactions
- 150 000 pool transactions

1.5 Traclient export

All transactions stored by GIR Titan-Hyperion can be exported using a “traclient” file. This file can then be used to process transactions in other applications. Detailled formats are available in the user manual.

There are several ways to generate a traclient file:

1.5.1 Automatic exports

- Live export: transactions are added to the file as they are processed.

  File name: `data\dto\traclien.dat`. When the file doesn't exist, it is created. Otherwise, transactions are appended to the end of the file.

- Daily export: transactions done during day D are exported on day D+1. The exportation hour can be configured.

  File name: `data\dto\traclien-YYYYMMDD.dat` A new file is created every day.

- Monthly export: transactions done during month M are exported on month M+1. The exportation day can be configured.

  File name: `data\dto\traclien-YYYYMM.dat` A new file is created every month.

1.5.2 Custom export

- Transactions done during a user-defined period are exported when a connected user uses the Traclient export menu. Several optional filters can be specified in addition to the selected period.

  File name: `data\dto\traclient.dat`. A new file is created at each export. The file can also be downloaded by the user.

1.5.3 Note on transaction modification

GIR Titan-Hyperion allows to modify a transaction after its first processing. This can be used, for instance, to change a meter value or a vehicle assignment.

The exported transaction can vary with the selected exportation method:

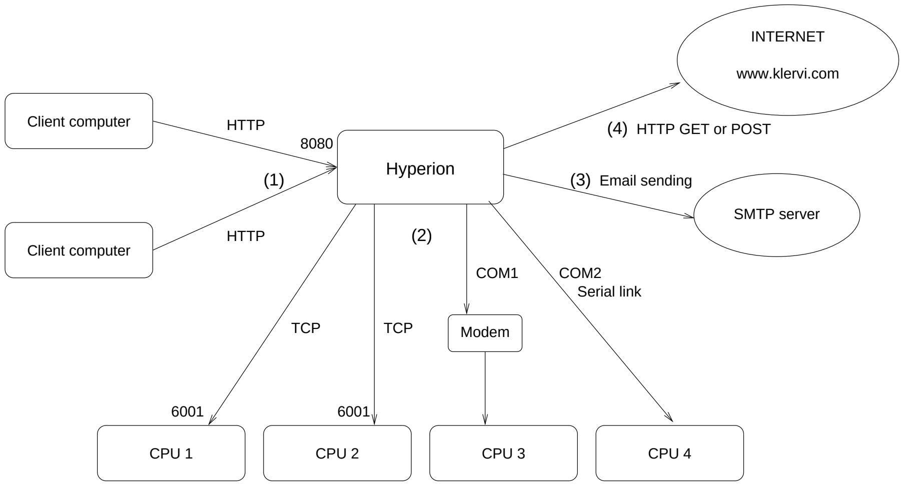
- Live export:
  - When using old formats (before C2 or F4), only the original transaction is exported.
  - When using recent formats (C2 or F4 and above), transactions modifications are also exported, with a special flag. This way, modifications can either be ignored, to keep the initial transaction, or processed, to duplicate the modification. However, doing so can require complex processing.
- Daily, monthly and custom export: Only the last version of a transaction is exported. For daily and monthly exports, this means that a modification will be exported if it is done between the end of the period and the actual export.

  Example: if transactions are exported each month on the 5th day, the modifications that are done before the 5th will be exported, and those that are done after won't be exported.

Chapter 2

Network access

2.1 General mechanism



All the network traffic generated when using GIR Titan-Hyperion can be classified among four types of queries:

1. Connections of client workstations to the Hyperion server: those are HTTP queries to TCP port 8080 of Hyperion integrated web server.

2. Connections to networked CPUs: Hyperion connects to the CPU network interface (custom IP address or DNS name) on the TCP port 6001. Communications with CPUs can be launched manually by the user, scheduled to be automatically runned at a given hour, of automatically launched when using “real time” mode.

   Fuel and access management CPUs:

   When the network link between CPUs and Hyperion is down, fuel delivery is still working: each CPU is autonomous[1], and data is synchronized between CPUs and Hyperion during dialogues. If a dia-

[1] TIP CPUs have a storage capacity of 2500 transactions

logue can't be run because of a bad network, synchronization will be postponed but fuel delivery and access control remain usable.

Key boxes:

Key boxes are remotely controlled by the software. Consequently, key delivery and return is only possible when the network connection between the key box and Hyperion is working. When this is not the case, an error message signaling a link failure will be displayed for every operation attempted on the key box.

3. E-mail sending: Hyperion can be configured to send information as e-mails. Those messages can be periodical (e.g. daily report) or triggered by an event (e.g. alerts). E-mail sending requires access to a SMTP server. Hyperion supports the Login, Plain and Cram-md5 authentication methods. It is possible to configure the SMTP server address and the e-mail sender address (identical sender address for all sent messages).

   Two formats are available for e-mail messages:

   - HTML format, for end users.
   - Text format, designed for post-processing by a robot, for conversion to other media types (SMS, Fax...). A description of this format is available in the user manual.

   Text messages conversion is part of the services proposed by GIR. When subscribing to this service, the SMTP server must allow to send e-mails to an address like address@klervi.com.

4. Internet access: Hyperion uses the Internet to send or receive files with GIR's after-sales-service, in a simplified way for the end user. File transfers can only be launched after an explicit validation by the final user. They are processed through GET or POST HTTP queries to www.kervi.com. It is possible to configure an access via a proxy server, with an optional authentication. Hyperion supports Basic, Digest and NTLM authentication types.

2.2 Traffic volume

Connection between client workstations and web server : similar to any standard web traffic. The size of HTML pages can vary from 10 KB to 1 MB (generally less than 100 KB).

Connection to CPUs :

1. Autonomous mode (Fuel, access)
   - with TIP1-Pabbay:
     - About 10 KB to retrieve 100 transactions.
     - About 100 KB to send 500 vehicles and 500 drivers.
     - Up to 1 MB for a full initialization.

    - Hyperion handles a synchronization with CPUs to minimize the amount of data to send. Hence, traffic volume during automatic dialogues will generally be limited to some KB. For example, a synchronization dialogue with a CPU containing 100 transactions will transfer around 20 KB during 30 seconds.
  - with TIP2-Vatersay:
    - About 20 KB to retrieve 100 transactions.
    - About 200 KB to send 500 vehicles and 500 drivers.
    - Up to 2 MB for a full initialization.
    - Same synchronization mechanism as TIP1-Pabbay.

2. Real time mode (Fuel, access) Daily traffic (in KB):

   $15 + 0.3 * N * B$

   where N is the number of transactions per CPU and per day, and B the number of CPUs
3. Real time mode (Key box) Daily traffic (in KB):

   $15 + 2 * N * B$

   where N is the number of transactions per CPU and per day, and B the number of CPUs

Internet connection :

- Downloading a new version: some MB.
- Sending the database: it really depends on the database size, about 1 MB for 100 vehicles and 4000 transactions, up to 50 MB for a very large database.

Chapter 3

System access rights (Windows version)

- Hyperion doesn't require any software other than the operating system: it completely stands alone, and integrates its own web server[1].

  Two web servers are available:

  - The web server integrated in the application hyperion.exe.
  - The web server tinyweb (tiny.exe).

  Those applications are launched by winhyprn.exe, which is displayed as an icon in the taskbar. The web server to use is configurable in the file winhyprn.ini.

- Hyperion doesn't need any particular system right other than total access to its installation tree (`c:\hyperion` by default). It doesn't use the registry nor any shared DLL.

- Dialogue with CPUs:

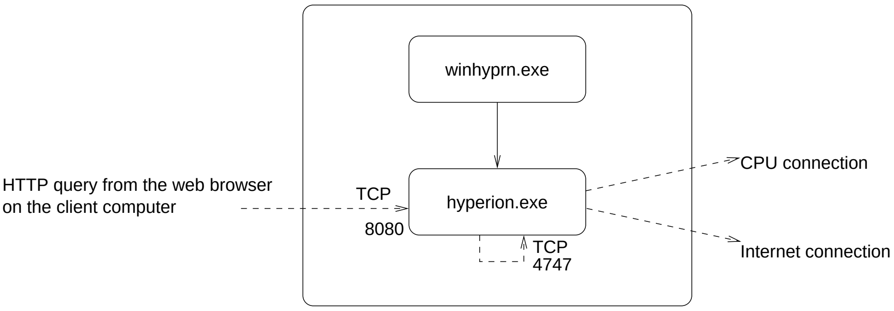
  for serial links or modems, Hyperion must be allowed to open the corresponding serial port.

  for network links, Hyperion must be allowed to establish a TCP connection to the CPU network interface

Under Windows, Hyperion is composed of the following processes:

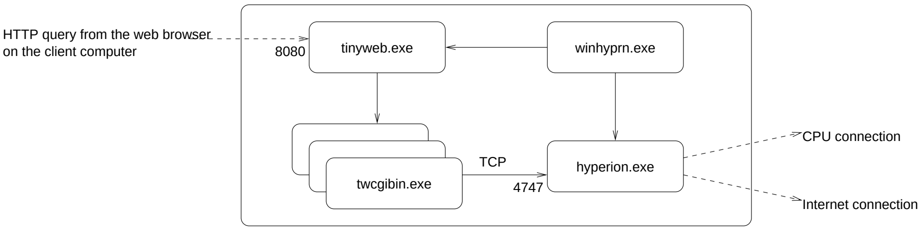
1. With integrated web server:

[1] Hyperion doesn't require to install Apache, IIS or any other web server. If such a server is installed on the same computer as Hyperion, it will run independently (Both servers must listen on different TCP ports, the default port for the Hyperion server is 8080)



- winhyprn.exe: Application allowing to launch hyperion.exe from an icon in the taskbar.
- hyperion.exe: Listens locally on the TCP ports 8080 and 4747, connects to CPUs using TCP or a serial port, connects to the internet for files transfers and e-mail sending.

2. With tinyweb web server:



- winhyprn.exe: Application allowing to launch tiny.exe and hyperion.exe from an icon in the taskbar.
- tiny.exe: Web server, listens locally on the TCP port 8080 and launches twcgibin.exe.
- twcgibin.exe: Connects to hyperion.exe on the loopback interface (127.0.0.1) on the TCP port 4747. Multiple instances of this process may run simultaneously.
- hyperion.exe: Listens locally on the TCP port 4747, connects to CPUs using TCP or a serial port, connects to the internet for files transfers and e-mail sending.

Chapter 4

System access rights (Linux version)

- Hyperion doesn't require any software other than the operating system: it completely stands alone, and integrates its own web server.

  However, it is possible to disable the integrated web server to use a third-party application (for example an Apache web server). In this case, the web server must provide cgi-bin management.

- Hyperion doesn't need any particular system right other than total access to its installation tree (`/home/hyperion` by default). It doesn't use any shared library.

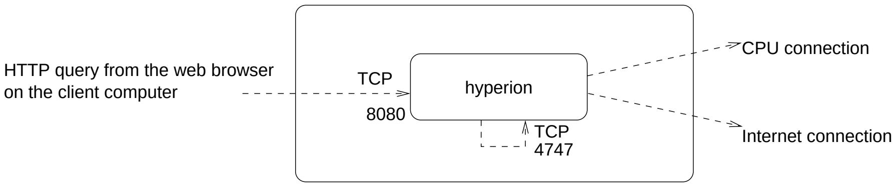
- Dialogue with CPUs:

  for serial links or modems, Hyperion must be allowed to open the corresponding serial port.

  for network links, Hyperion must be allowed to establish a TCP connection to the CPU network interface

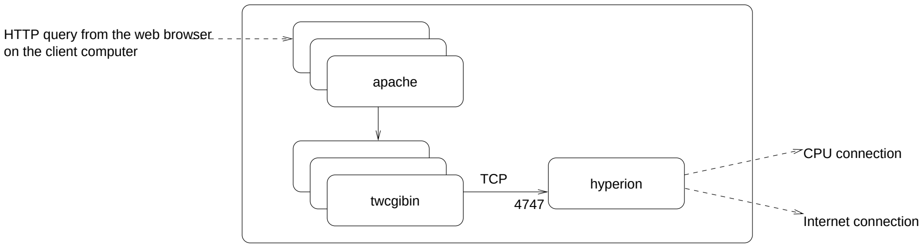
Under Linux, Hyperion is composed of the following processes:

1. In autonomous mode:



   - hyperion: Listens locally on the TCP ports 8080 and 4747, connects to CPUs using TCP or a serial port, connects to the internet for files transfers and e-mail sending.

2. With an external Apache web server:



- apache: Web server, listens on a dedicated port and launches twcgibin. Multiple instances of this process may run simultaneously.
- twcgibin: Connects to the process hyperion on the loopback interface (127.0.0.1) on the TCP port 4747. Multiple instances of this process may run simultaneously.
- hyperion: Listens locally on the TCP port 4747, connects to the CPUs using TCP or a serial port, connects to the internet for files transfers and e-mail sending.

Chapter 5

Intranet integration

5.1 Graphical integration

Multiple aspects of GIR Titan-Hyperion's appearance can be customized, allowing to adapt it to the graphical style of an intranet.

Those settings are read by GIR Titan-Hyperion on startup, in the `data\dti\cust_cfg.txt` file. A documented example of this file is provided in the `examples` directory of the application. It contains a set of variables of the form "Name=Value".

GIR Titan-Hyperion can also include HTML files placed in `data\dti` to allow a more powerful integration. See the `cust_cfg.txt` example file for more information.

5.2 Authentication

GIR Titan-Hyperion supports several methods to authenticate users connecting to the application:

Classical: Users are declared in the application by a login and a password, that they have to enter when connecting to the application. Password validity is checked by Hyperion. This is the default authentication mode.

External authentication: Users are declared in the application only by their login. To connect to the application, they enter their login and password as above, and this data is then forwarded by Hyperion to an external script, which checks the password validity.

Automatic connection: Users are declared in the application only by their login, and the authentication is entirely processed by a third-party application. This solution is highly flexible, and will generally be used to make the authentication process invisible for end users, who will be able to connect simply by clicking on a hypertext link.

Whatever the method, the user login must be defined in GIR Titan-Hyperion database. The Password field specifies if the user can connect using the first method. The *Auto. id.* option specifies if he can connect using one of the two other methods.

The last two authentication methods are detailed below.

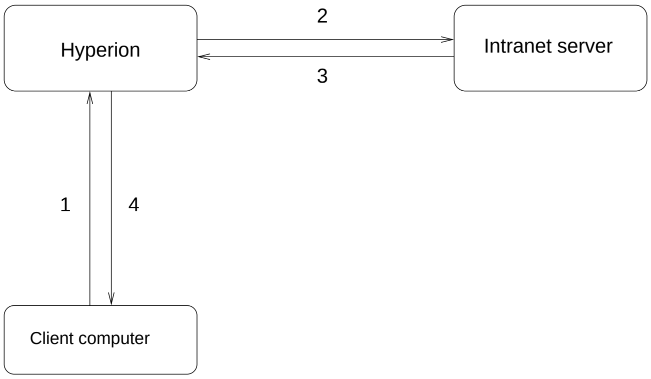
5.2.1 Users access rights

Each user is assigned one of the following 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.
- *Inherited*: Identical to the *Standard* level, with the additionnal possibility to specify an authorization profile at login time. This feature is only available when using external authentication or automatic connection. It allows the following possibilities:
  - Factorizing authorizations definition when multiple users have the same rights.
  - Not defining explicitely every user in GIR Titan-Hyperion database. It is possible to define only authorization profiles, and to automatically create an authenticated user when he doesn't exist in the database. This feature must be enable in the `cust_cfg.txt` file.

5.2.2 External authentication

External authentication allows to export the control of a (login, password) couple validity, using the following scenario:



1. Entry of login and password by the user on GIR Titan-Hyperion connection page.
2. Forwarding of this data to an intranet server using a HTTP GET query.
3. The intranet server processes the authentication and returns a result.
4. Login to home page on success, or error message on failure.

The URL to call to process an external authentication is defined by the "ExtAuthScriptURL" variable in the `cust_cfg.txt` file.

Authentication result must be a web page in text/plain format, with its first line beginning with "OK" or "ERR". For more information, see the "External authentication" section in the `cust_cfg.txt` example file.

During the first step, before forwarding information, Hyperion first checks that the login is defined in its database. If a password is defined for this login, it is compared to the entered password. If the passwords match, authentication is successfull without any external call.

If the passwords don't match, or if no password is defined, Hyperion checks that the user has the *Auto. id.* option, and then processes the external authentication.

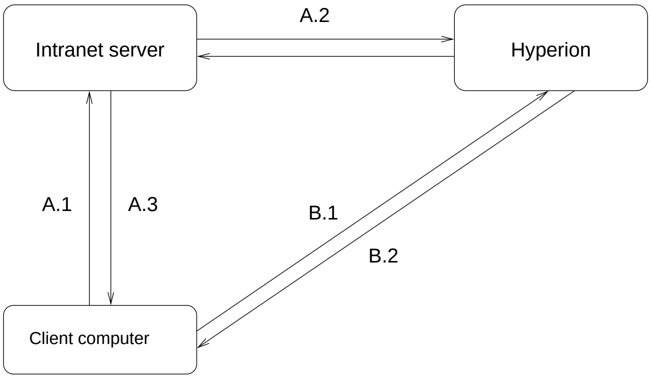
Example:

User	Password	*Auto. id.*	Authentication
A	Yes	No	Hyperion only
B	No	Yes	External only
C	Yes	Yes	Hyperion then external on failure
D	No	No	User forbidden

A demonstration PHP script is available in `examples\hyperion-extauth.php`. It shows how to set up an external authentication using a LDAP directory.

5.2.3 Automatic connection

Automatic connection allows to totally export the authentication process in a third-party application. This solution is implemented by GIR Titan-Hyperion using the following design:



A. Query from the client computer to an intranet server

1. The client computer connects to an intranet server which processes the authentication.
2. The intranet server registers the user as authenticated in GIR Titan-Hyperion, with a HTTP GET query.
3. The intranet server forwards the client computer to GIR Titan-Hyperion login URL

B. Connection from the client computer to GIR Titan-Hyperion

1. The web browser processes the forwarding.
2. The user is connected.

The intranet server processing authentication can be installed on the same computer as GIR Titan-Hyperion as well as on another computer on the network. For the latter, the intranet server must be able to launch an HTTP GET query on the server hosting GIR Titan-Hyperion.

Security of the authentication is based on the following elements:

- When registering authentication (A.2), the IP address of the sender must match the mask defined by the variable “AutologAuthAddr” in the `cust_cfg.txt` file. If this variable is not set, no authentication can be registered.
- After processing the authentication (A.1), the intranet server generates a random number (“cookie”), which is forwarded to GIR Titan-Hyperion during registration (A.2), then to the client computer during forwarding (A.3 and B.1). GIR Titan-Hyperion keeps this cookie during a short delay, and checks that the value provided by the client computer matches the value previously registered.
- An optional password protection (variable “AutologAuthPassword” in `cust_cfg.txt`) during registration.

Users declaration

Automatic connection is only allowed to users declared in GIR Titan-Hyperion with the option *Auto. id.* enabled. Defining a password allows the user to connect manually.

Example:

User	Password	*Auto. id.*	Allowed connections
A	Yes	No	Manual only
B	No	Yes	Auto. only
C	Yes	Yes	Manual + Auto.

Users names in GIR Titan-Hyperion are limited to 20 characters, and are not necessarily identical to those used in the intranet. Hence, it is possible to define a map between intranet logins and Hyperion logins. Those matches can be declared manually or be implicit. For more information, see the “Autologin” section in the `cust_cfg.txt` example file.

Logout

By default, when a user logs out from GIR Titan-Hyperion, he is forwarded to the application connection page. When using automatic connection, the user doesn't need to access this page to log in. Hence, it is possible to redefine the forwarding URL, for example to forward the user to the intranet home page, or to close the browser window. For more information, see the “Autologin” section in the `cust_cfg.txt` example file.

Installation on an IIS server

An ASP script is provided to implement automatic connection to GIR Titan-Hyperion using an IIS web server.

Installation procedure:

- Install the `hyperion-autologin.asp` script in a public directory of the IIS web server (for example `c:\inetpub\Scripts`)
- Edit the file to customize the line

  ```
  cfgURL = "http://MYSERVER/cgi-bin/twcgibin?p=4747"
  ```

  depending on the server hosting GIR Titan-Hyperion.
- Go to “Internet services manager”
- Display the properties of the `hyperion-autologin.asp` script
- Click on the “File or directory security” tab
- Under “Anonymous access and authentication control”, click on Modify
- In the “Authentication methods” dialog box, disable the “Anonymous access” checkbox, then enable the “Basic authentication, Digest authentication or integrated authentication (Stimulation/Reply of Windows NT)” checkbox.

- Click on OK to close the dialog boxes.
- Define a link to http://*IIS server address*/Scripts/hyperion-autologin.asp to realize an automatic connection to GIR Titan-Hyperion.