Webdyn*gate* Modbus User's Manual

Models concerned
WGM10 (-M)
WGM20 (-M)
WGM30 (-M)

Version 1.10
September 2003

Webdyn

Table of Contents

1 Preamble

This document is the complete user's manual for the entire range of Webdyngate Modbus gateways. As a result, some of the functions described in this document are not applicable to all the models:

Restrictions according to the models:

Option **–M**
This option means that your gateway is equipped with a built-in modem. If you do not have this option, the ''incoming modem'' administration menu and the ''modem'' option of the ''messaging'' administration menu will not be accessible.

WGM10
This model only features the ''Administration'' menu and the ''Serial Modbus'' operating menu. It does not have a flash disk and only operates in the Modbus TCP mode.

WGM20
This model does not have a flash disk, it cannot be customized. The ''user'' menu is not accessible.

WGM30
This model is the most complete and features all the options (WGM30 –M for the option with modem).

2 Introduction

2.1 Description

The serial Modbus protocol is a master/slave protocol (1 single master per network) used mainly in the industrial environment. It enables supervision equipment to communicate with one or more industrial devices (Programmable Logic Controllers, automatons, probes, etc.).

Exchanges are initiated by the master in the form of requests sent to a specified slave. When the destination slave has understood the request, it sends the response.

The Modbus protocol is a half duplex collision-free protocol. At a given time there can only be one entity on the bus, either the master or the slave.

A master cannot send a new request before receiving the response to the preceding request or until a time out has lapsed.

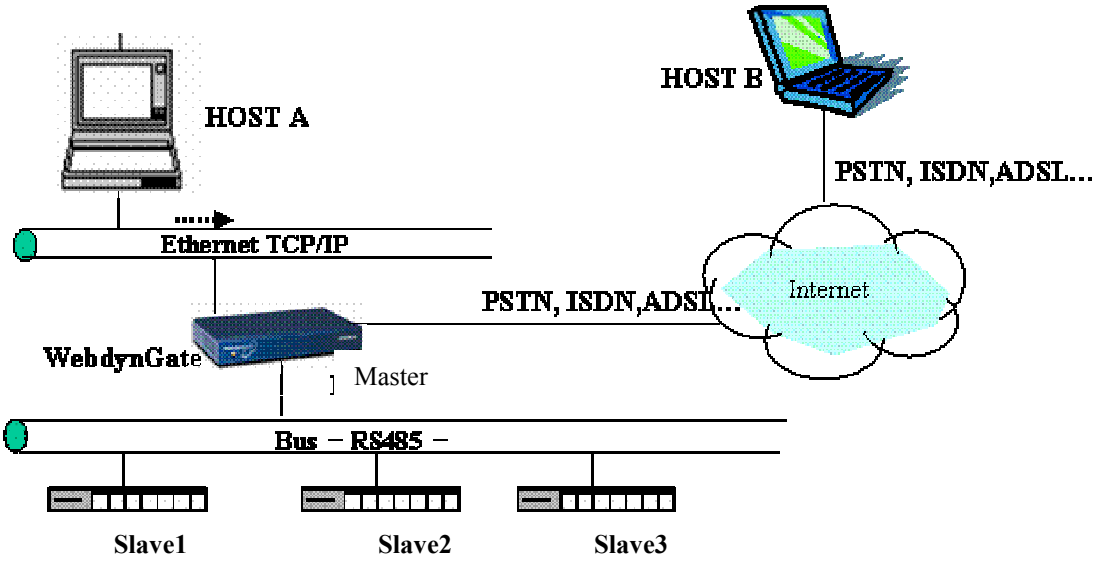
Modbus/TCP is an extension of serial Modbus that determines how messages exchanged on a serial Modbus are encoded and carried on a TCP/IP network.

Modbus/TCP is generally used in situations where the services associated with a master are remote.

You can find all the serial Modbus and Modbus TCP protocol specifications on the www.modicon.com and www.modbus.org website.

Webdyngate is a new gateway concept that allows serial Modbus equipment to be connected to TCP/IP equipment. More than just a TCP encoding system, Webdyngate gateways allow the implementation of new services like formatting Modbus variables to the HTML format or alarm feedback.

Figure 1.1 shows all the elements that can be integrated around a Webdyngate gateway.



- Serial Modbus slave equipment (1, 2, 3) connected to the gateway through a RS485 serial link.
- Host A connected to the gateway through a local TCP/IP Ethernet.
- Host B connected to the gateway through a remote Internet or point-to-point STN network.

As regards the serial Modbus protocol, the Webdyngate plus a full master role. The local or remote IP network is there only to remote the services associated with the use of the Modbus protocol.

Take the example of a Modbus TCP program installed on Host A. This program may want to collect data from slave 1. To do so, Host A prepares a Modbus request, packets in a TCP frame and sends it to the Webdyn gateway on which a Modbus TCP server is started up.

Once received, the frame is decoded by the gateway and sent through the serial port to slave 1. As we have seen previously, any frame transmitted by a Modbus master requires a response! In the opposite direction, this response is encapsulated in an IP frame and returned to Host A.

At the protocol level, the slaves (1, 2, 3) only see a single master: the Webdyngate gateway. It is up to the gateway to synchronize all the requests from the virtual masters, that is the different hosts, connected on the local and remote IP networks.

2.2 Webdyngate & IP services

2.2.1 Modbus/TCP

Modbus/TCP allows client/server model messages to be exchanged between devices connected on the same IP network.

Basically, Modbus TCP encapsulates a serial Modbus frame in a TCP frame.

The TCP protocol is an connection-oriented type protocol, that is that before any message exchange a client must be connected to a server and this connection lasts until one of the two ends disconnects.

The TCP protocol is thus suited to the master/slave serial Modbus protocol where any request by the master waits for a response from the slave.

3 Webdyngate Interfaces

Serial Modbus Interface

- Two wire
- Four wire

Ethernet Interface

I/O on-off interface

STN Modem interface

(See implementation manual for more information).

4 Webdyngate Man-Machine Interface

4.1 Connection

Webdyngate is configured and operated using an Internet browser (Internet Explorer version 4 and over or Netscape version 6 and over).

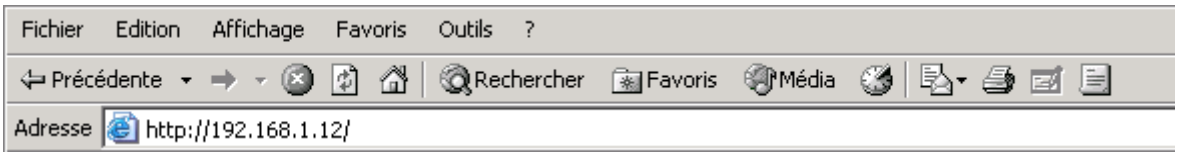
The Webdyngate gate behaves like a conventional Web server. The gateway delivers HTML pages according to the requests transmitted by the browser.

Like a Web server, the gateway has a home page from which the user can browse and load all the pages of the server.

This page is accessed using the URL:

- http://@IP/ where @IP is the IP address of the gateway.

By default, the Webdyngate gateway has the factory IP address 192.168.1.12



If the user changes the IP address of the gateway in the configuration pages, it is that address that must always be indicated in the URL.

As we will see further on in this document (see Security section for user rights), the use rights of a gateway depend on the identity of the user. It is thus necessary for the user to identify him/herself before accessing the gateway's functionalities.

This identification is achieved using the Webdyngate dialogue box.

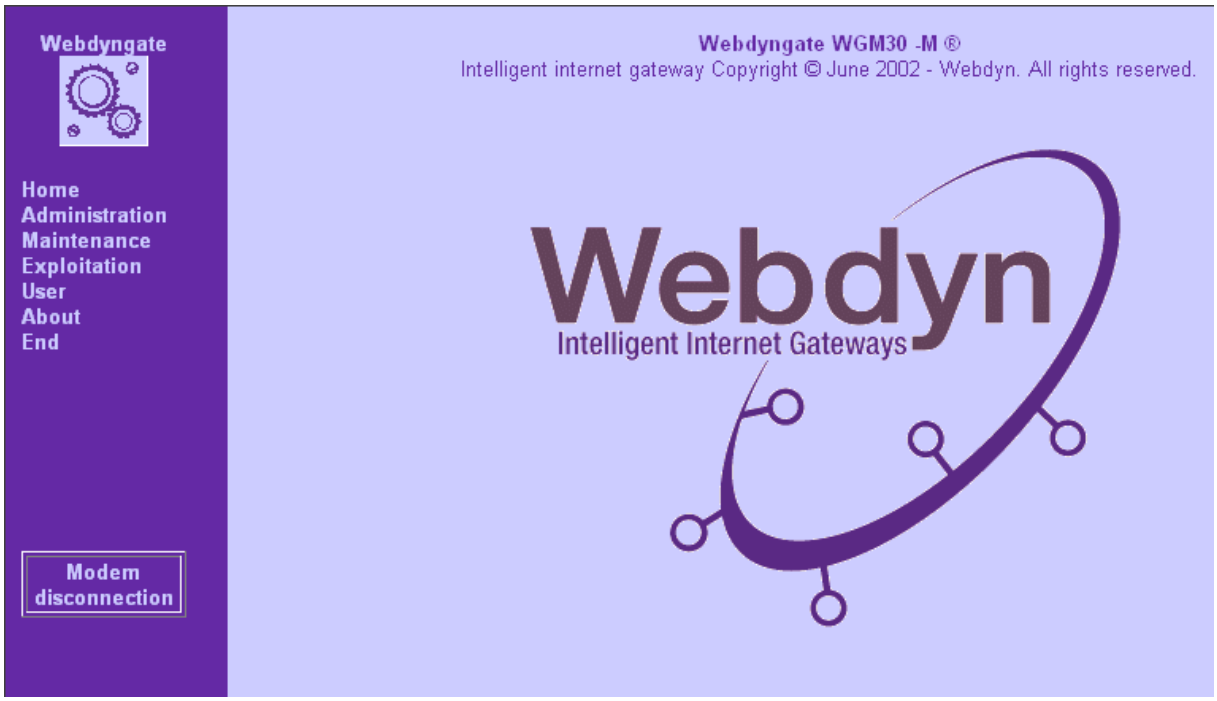
After having filled in the ''user name'' and ''password'' fields, and clicked on OK, the gateway checks the identity. If the user is recognized, the home page is displayed.

4.2 Browsing through the menus

All the pages are configured so that they are compatible with an 800 x 600 display.

The HTML pages have 2 parts:

- On the left: the menu
- On the right: the data display part

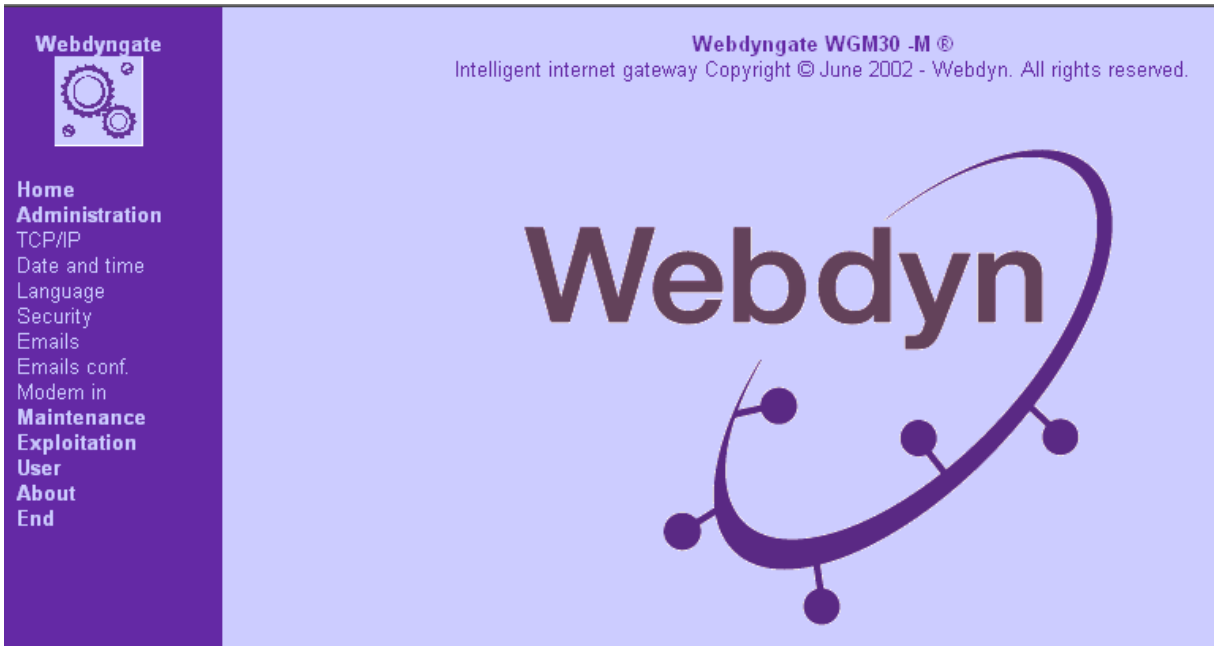


The menu part is used to access the following menus:

- Administration
- Maintenance (model >10)
- Operation (model > 10)
- User (HTML Pages developed by the client – 30 Models).
- Modem disconnection (M Models).

The ''**Administration''** menu provides access to general parameter configuration.

Its accessibility depends on the use rights of the user who logged in.

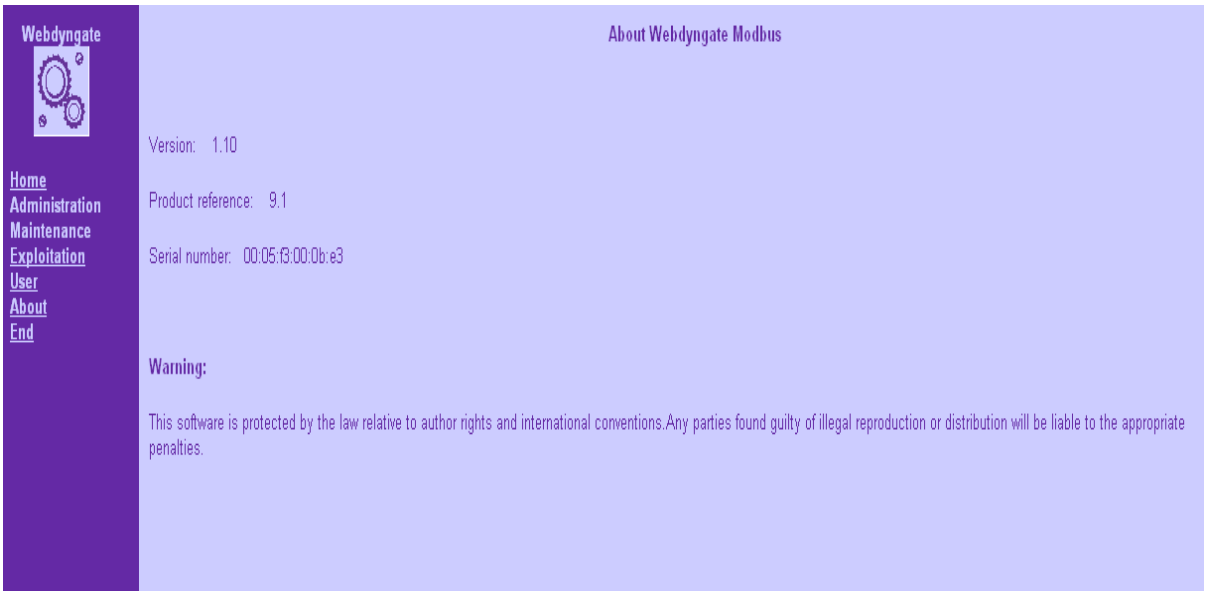


The ''**Maintenance''** menu is used to configure Modbus parameters.

Its accessibility depends on the use rights of the user who logged in.

The ‘’**Exploitation’’** menu is used to read the parameters and variables managed by the gateway.

The ''**About''** menu displays the hardware and software references of the gateway.



The ''**End''** menu closes the connection in progress and returns to the login page so that a login under another name is possible.

The ''**Modem disconnection''** button is used when a user is connected to the gateway via a modem. It ends the connection either via the modem link or from the locally connected Internet browser.

5 Webdyngate configuration

5.1 Local IP Configuration

5.1.1 Introduction

Connected on a network (Internet, Intranet or simple LAN) the Webdyngate must have an IP configuration to dialogue with the other devices connected on the same network.

This configuration is composed of three elements, each one being equivalent to an address:

- IP address
- Subnet mask
- Router address

Only the first two elements (IP address and subnet mask) are compulsory, the router address is optional.

All these addresses are represented in the following alphanumerical form (dotted decimal ASCII string):

- w.x.y.z where (w, x, y, z) are whole numbers between 0 and 255.

By default, Webdyngate Modbus is supplied with the factory configuration:

- **IP Address = 192.168.1.12**
- **Subnet mask = 255.255.255.0**

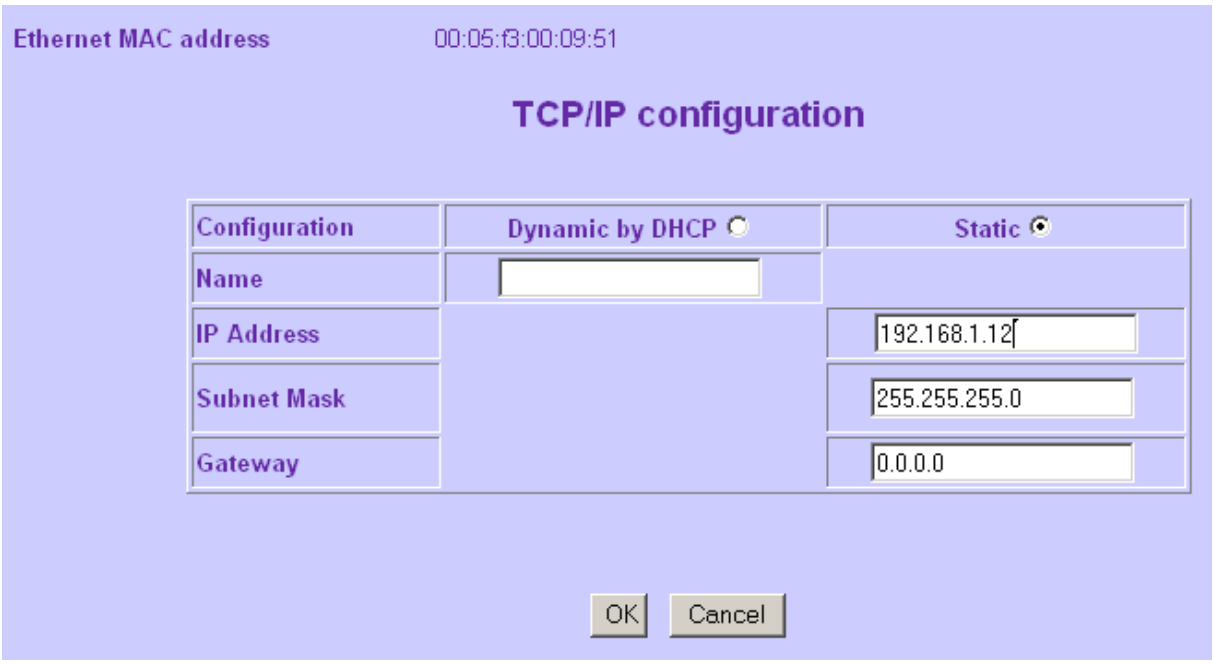
This factory configuration, identical for all Webdyngate gateways, can only be used temporarily and for initialization purposes.

In the operational phase, the Webdyngate gateway must have a complete configuration compatible with its connection network and different from the factory configuration.

Any new configuration is saved in the Flash EPROM. On power-up or following a Reset, a Webdyngate gateway starts up with the last recorded parameters.

5.1.2 Configuration

The operational configuration is set up from the TCP/IP configuration page.



By default, this page indicates the current configuration of the gateway. On first use, this current configuration is the factory configuration.

The gateway's MAC address is found at the top left of the page. This address is linked to the gateway's Ethernet interface.
It is represented in the form:
ab:cd :ef :gh :ij : kl where (a,b,c,d,e,f,g,h,i,j,k,l) are hexadecimal numbers.

Supplied by IEEE this address is unique and serves as the gateway's serial number.

5.1.3 Static or Dynamic Configuration

A Webdyngate gateway can be configured statically or dynamically.

In the **static mode,** the user or administrator fills in all the information using the complete IP address. If the router is not used, the address supplied is nil 0. 0. 0. 0

This working mode must be used with care as two devices connected on the same network must have different addresses.

In the **dynamic mode** (DHCP) the attribution of the IP configuration is ''outsourced'' to an outside server.

This server has a table that lists all network devices.

On power-up or following a Reset, the Webdyngate gateway, configured in the DHCP mode, sends a request over the IP network searching for a server that has an address book listing the Webdyngate gateway.

If the result is positive, the server thus contacted returns its complete IP configuration to the Webdyngate gateway which can then start up.

If the result is negative, (no server), at the end of one minute, the Webdyngate gateway abandons the search and starts up under the last static configuration recorded.

The way a DHCP server recognizes all the network-connected devices is the MAC address or DHCP name. The gateway can be called from the browser with this name: for example: http://webdyngate.webdyn.com

The dynamic mode allows all the addresses of the network devices to be centralized on the same server.

5.1.4 Router

When the router address of the Webdyngate gateway differs from the nil value (0.0.0.0) all the IP packets from the Webdyngate gateway that do not have a direct route to their destination are sent to the router.

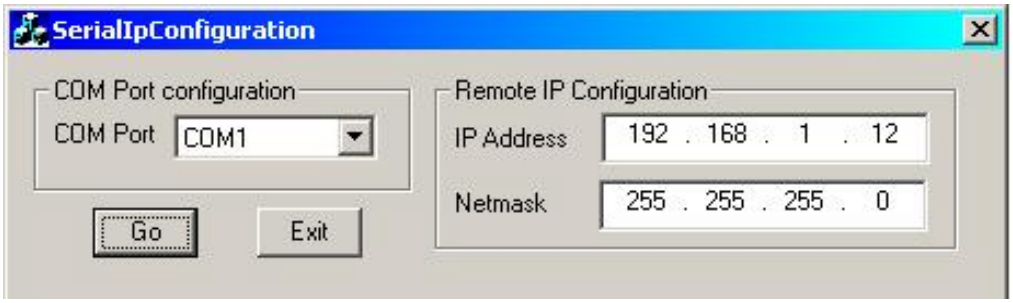
The router is a specialized device that interfaces two different networks.

The router address is used when one wants to connect a Webdyngate gateway to a remote device that is not physically on the same network.

5.1.5 Configuration with serial port

If the IP address of your Webdyngate is not compatible with your network, you cannot connect it by internet. The tool **SerialIpConfiguration.exe** allows you to change this IP address by the serial port of the gateway.

a) Plug a RS232 cable female-female between the PC ant the serial connector of the Webdyngate (catalog Webdyn ref FP-CAB-RSFF1 (serial cable(DB9-DB9 1.5m))
b) On the PC, go in the directory of the tool : SerialIpConfiguration.exe (on the CD-ROM by default)
c) Launch SerialIpConfiguration.exe by double-clicking on it
d) The following window opens:

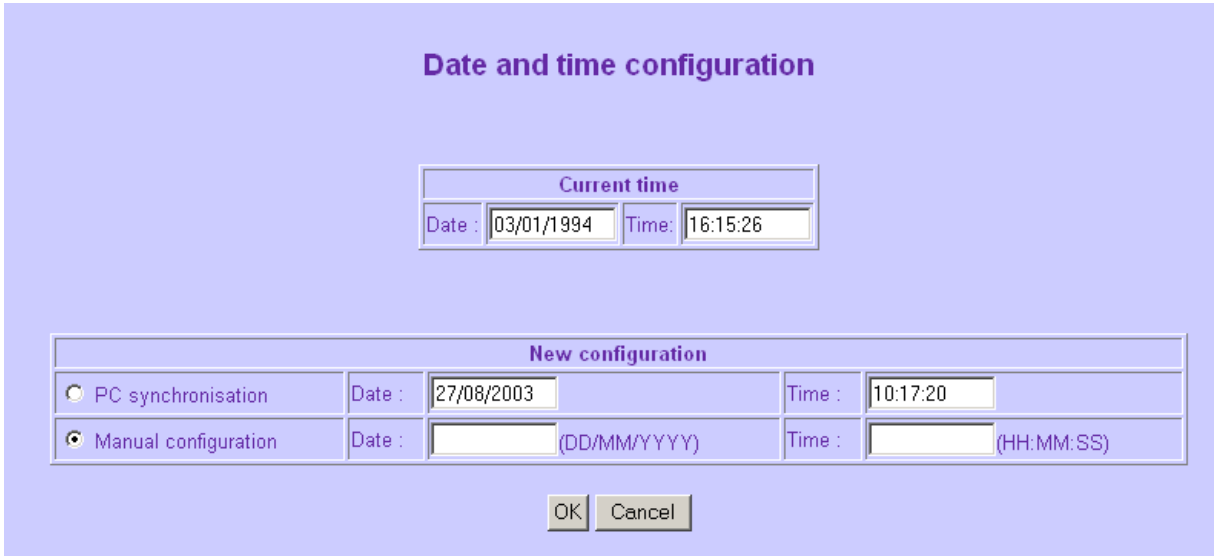


e) In « COM Port » choose the pc serial port on which the cable is connected (COM1 by default)
f) In « IP Address » enter the new IP address chosen for the gateway (192.168.1.12 = factory configuration)
g) In 'Netmask » enter the subnet mask address (255.255.255.0 by default)
h) Click on « GO » and reboot hte gateway. When « current status » displays « configuration successfull », the gateway is reconfigured.
i) Click on « exit » to quit
j) You can connect immediately on the gateway with your internet browser without to have toreboot.:
e.g. :if the new IP adress is 192.121.50.50 , enter :
http://192.121.50.50

5.2 Date and time

Certain events (alarms, traces, etc.) related to the Webdyngate gateway can be date/time stamped. Webdyngate has a real time clock that runs on a battery thus enabling it to follow the current date and time.

This clock is enabled in the ''date and time configuration'' page. Time setting can be automatically synchronized with the remote PC or entered manually.



By default, the "manual configuration" choice is validated. Whatever your choice is, the time setting is done only once when clicking on "OK". The choice '"PC synchonisation" is not saved.

5.3 Language

Webdyngate has a dictionary that enables it to deliver HTML pages according to the language chosen.

This configuration can be activated dynamically.
By default , there are two languages, French and English; French is the default choice. For other languages, see with commercial service.

5.4 Password

The use and configuration of Webdyngate are protected by an identification system. There are three identification levels and use rights are associated with each level: Administration, Maintenance, Operation.

The Administration level enables the administration of the gateway, the configuration of the Modbus interface, the operation of the Modbus equipment and access to user pages for model 30.

(Only the underlined commands are accessible)

The Maintenance level is used to configure the Modbus interface and operate the Modbus equipment and access the user pages for model 30.

(Only the underlined commands are accessible)

The Operation level only allows the operation of the Modbus equipment and access to the user pages for model 30.

(Only the underlined commands are accessible)

A user is identified by a name (or login) or password.

By default, the different logins and passwords are initialized at:

Level	Login	Password
Administration	userhigh	high
Maintenance	usermiddle	middle
Exploitation	userlow	low

The login and password can be changed for each level:

The only condition to change the password of a level is to know the old password.

You cannot change both name and password, you have to click on "OK" between the two modifications.

The number of characters must be between 3 and 20 for the user name and between 3 and 8 for the password. Spaces are not allowed at the end of words and no two names must be identical.

A single name and a single simultaneous connection are accepted for the Administration and Maintenance levels.

A single name but several simultaneous connections are accepted for the Operation level.

5.5 Modem and Messaging Configuration

5.5.1 Introduction

The Webdyngate "-M" models are supplied with an onboard PSTN modem allowing various modem based functions.

The physical connection is made using the standard RJ11 socket located on the rear panel of the devic enclosure (the cable is supplied).

Two different situations are possible :

- The modem is the call initiator (via a Webdyngate request) this is known as the INIT condition. This is the case when an alarm is triggered or when the IPCallBack function is activated. The Webdyngate connects via the telephone line to an Internet Service Provider (ISP) and sends an email.
- The modem is in the WAIT condition where it may be called for one of the following reasons :

- The gateway is configured in IPCallBack mode. In this case the gateway detects that the administrator wishes to connect, it doesn't answer but instead waits until the line stops ringing. It then passes into the INIT state and connects to the ISP, it then sends an email to the configured destination(s) that contains its current temporary public IP address (note : this address was provided by the ISP during the connection phase). If however the number of rings is greater than the IPCallBack ring number configured then the gateway goes back into the WAIT condition and will answer the call using the configured modem parameters. This answer is like a classic RAS (Remote Access Server).
- The gateway is configured to answer the incoming call (using the RAS function). In this mode the IPCallBack feature has not been selected and so the gateway will answer the incoming call using the PPP settings that have been defined.

By default the gateway and the modem are initialized in the WAIT state. If an external event creates a PSTN connection then the system switches into the INIT state and the connection will be established. At the end of a session when the modem hangs-up, the gateway and the modem go back into the WAIT mode. If during an incoming call there is an external event that occurs, then the administrator has the choice to configure the gateway to either drop the line and send the email or wait until the session is over before sending the email. (Note : this configuration must be performed during the installation procedure).

5.5.2 E-mails

There can be many destination addresses for the e-mails sent by the gateway.

Webdyngate can group the e-mails as aliases.

An alias is composed of one or more e-mails, each e-mail being separated by a semi-colon '' ;''.

The administrator is free to name the alias. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that is used to identify the alias.

The ''Alias Configuration'' HTML page is used to visualize all the aliases already registered by the gateway and modify, add or delete an alias.

Each new or old alias is linked to a line of the form and only one line can be changed at a time.

Two identical alias names are accepted but displayed in red to signal the risk of error.

To validate a line, you must click on the OK button and only the line concerned by that button will be taken into account by the gateway.

If a modification is necessary on several aliases, the operation must be repeated as many times as there are aliases being modified.

5.5.3 Messaging

The Webdyngate gateway can send an e-mail synchronized with an outside event:

- Alarm
- IPCallBack
- Life signal

An e-mail is always sent to a messaging server (SMTP server). This server's address is configured in the ''Messaging configuration'' page.

This server can be located on the same local network as the gateway or accessible through the STN.

Generally this server is known as a domain name, for example ''smtp.webdyn.com'' rather than as an IP address ''1.2.3.4''.

The role of the DNS server is to translate a domain name written in ASCII (smtp.webdyn.com) into an IP address (1.2.3.4). The DNS server plays the role of a directory.

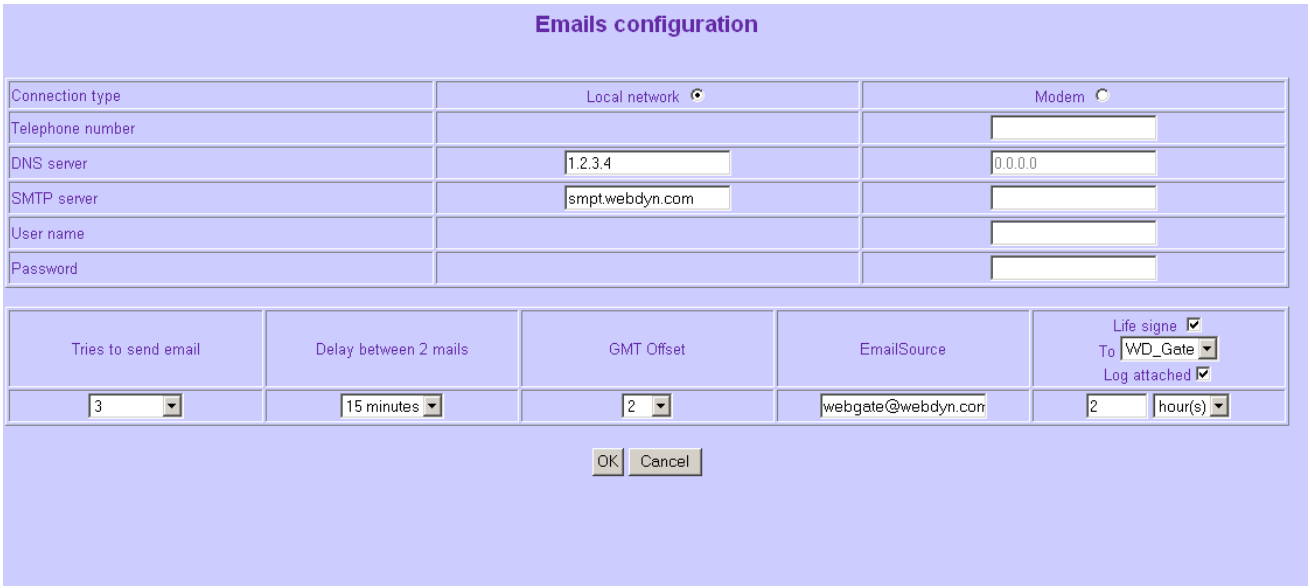
It works as follows: When the gateway wants to sent an e-mail to the smtp.webdyn.com messaging server, it first contacts the DNS server asking for the IP address corresponding to the smtp.webdyn.com domain.

If this domain is known to the DNS server, it answers by sending the IP address of the SMTP server and the ''send an e-mail'' application can begin.

A DNS server is always known by its IP address. This address can be provided by your network administrator for a local network. When you use remote access, your IAP supplies this address.

For non-local machines, the DNS server knows how to contact other DNS servers present on the network able to provide the translation.

The use of a DNS server is optional. If no DNS server (DNS address 0.0.0.0) is specified, the user must know the IP address of the messaging server (SMTP server) to be contacted.



In addition to the SMTP and DNS server addresses, the remote access by modem configuration requires knowing the telephone number of the IAP and your identifier (user name and password).

All these characteristics are provided with your subscription.

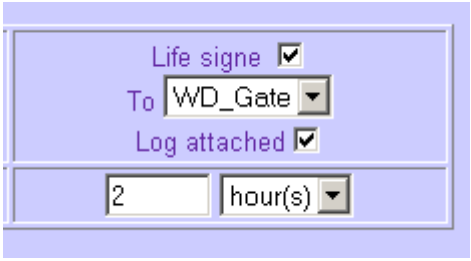
To enter a number going through a switchboard, write the switchboard number directly followed by a 10-digit number.

If a pause must be left between these 2 numbers, insert a semi-colon (;)
e.g. 0 ;0123456789

Sending an e-mail can fail for various reasons (busy signal, SMTP server unlocatable etc.). When an e-mail is linked to an alarm, this loss can have serious consequences. In the event of error, Webdyngate can repeat the e-mail. The number of repeats and the waiting time between two attempts can be programmed.

De même le destinataire de l'e-mail a besoin de connaître son émetteur, le champ Emetteur sert à personnaliser ce paramètre, il n'a pas besoin d'exister réellement, il doit simplement avoir la syntaxe d'une adresse e-mail. Cependant, si l'e-mail ne passe pas, il est renvoyé à son émetteur .L'administrateur peut donc entrer sa propre adresse dans ce champ, il sera ainsi averti si les e-mails envoyés par la passerelle ne passent pas. La passerelle elle-même ne sait pas recevoir d'email.

A simple way to find out if the gateway is ''still alive'' is to activate the life signal. This life signal is composed of an e-mail periodically sent by the gateway. It is possible to program the frequency and the presence of the log as an attachment (beginning with model 20).



There are two possible choices concerning the frequency:

- every x hours
- every y days

In case of days, the sending time is set to 1 :00 a.m. (gateway time).

To limit the size of the e-mails, if the attached log option is activated, the log is compressed to ''gz'' format. This compression divides the size of the attachment by 10. The ''gz'' format is compatible with all decompression tools (Winzip, PkUnzip, etc). available with the main operating systems.

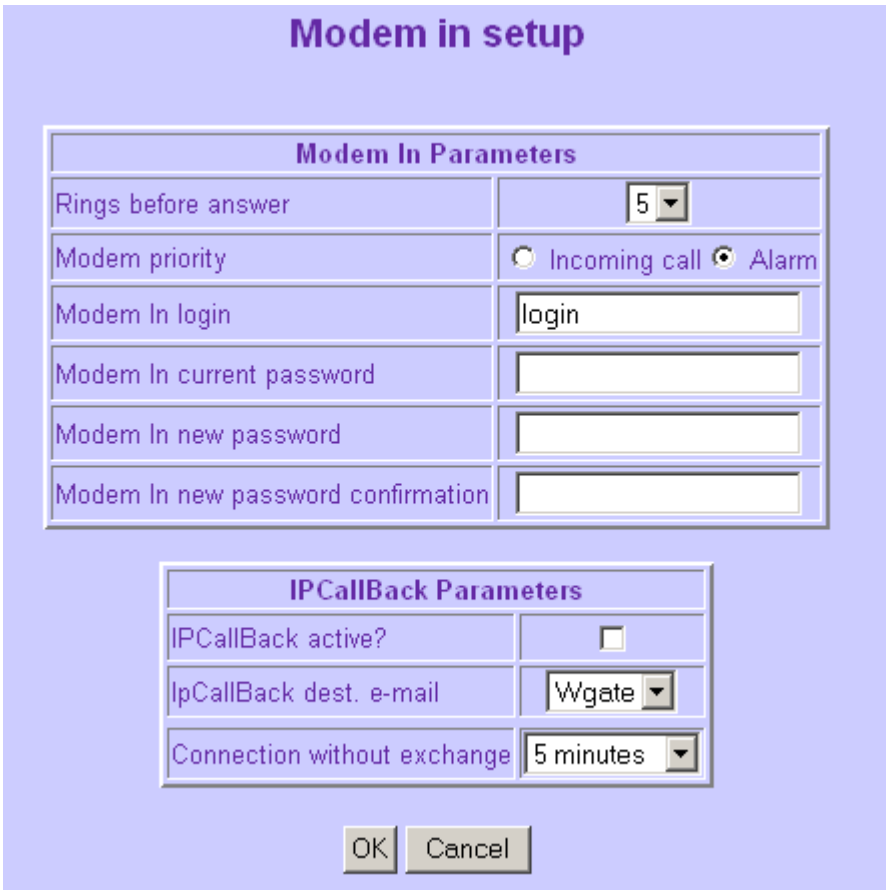
Note : If the log reaches 2 MB and the "life sign" is activated then a mail will be sent even if the programmed delay has not been reached.

The remote connection of the gateway to an IAP can be summarized as comprising the following actions:

- The gateway and modem initiate a telephone call to the IAP.
- The IAP off-hooks.
- The IAP authenticates the identification (user name, password) of the gateway.

- The IAP attributes an official IP address (address accessible to any Net-enabled device).
- The gateway connects to the SMTP server and sends its e-mail.
- Once the e-mail is sent the gateway on-hooks.

5.6 Incoming modem configuration



In the WAIT mode two cases may arise:

- An administrator wants to set up a connection to access the gateway's IP services. In this case, the Administrator doesn't validate the IPCallBack active option.
- An administrator wants to activate the IPCallBack function. In this case, he/she validates the IPCallback option.

5.6.1 Incoming connection

In the WAIT incoming call mode, the connection is at the initiative of the remote user. The parameters to be configured are:

- the user name
- the password

By default these two parameters are initiated with the following values:
The user name ***login***
The password ***password***

They are modifiable with the rights of Administration. The User name string must have a length between 3 and 20 caracters, the Password string must have a length between 3 and 8 caracters.

In a remote point-to-point connection, in addition to authentication parameters an IP address must be attributed to both extremities of the connection (user & Webdyngate). This attribution is handled by the PPP protocol (Point-to-Point Protocol). This protocol specifies that once the connection is set up over the STN network, both IP addresses can be attributed either by the device configured in the INIT mode or by the device configured in the WAIT mode. To preserve the compatibility with the Internet operating mode in our case, it is the device configured in the WAIT mode that supplies the IP addresses. In the incoming mode, that is when the user connects to the gateway, it is the gateway that supplies the IP addresses.

In any case, the address supplied to the user is **192.168.2.13** and the address configured on the gateway is **192.168.2.12**

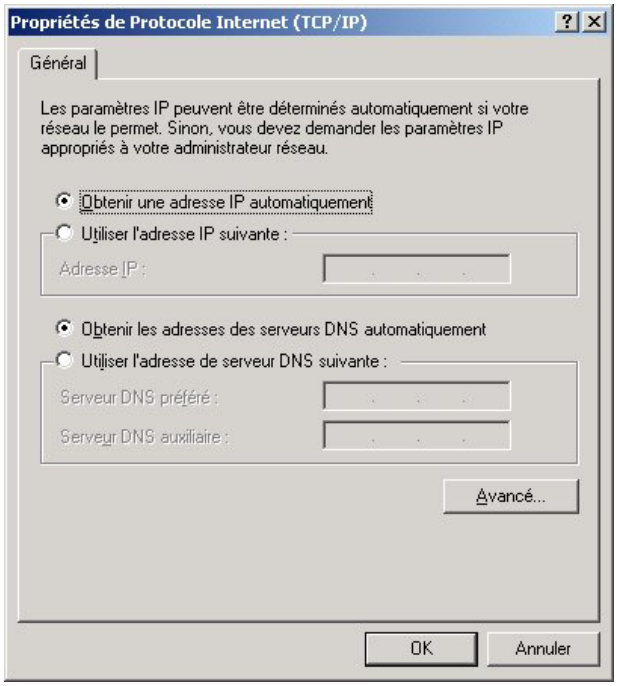
The gateway answers at the end of the number of rings that are configured in the field called "Rings before answer" (the default is 3 rings).

In short, the remote connection to the gateway includes the following actions:

The administrator configures his/her PC for remote access. The parameters of this configuration are:

- the gateway's telephone number,
- the user name,
- the password.

The configuration of the PPP server. This configuration is the default configuration of any Windows or Linux OS. In particular, the administrator can check that in the properties of the TCP protocol, the IP address is obtained automatically from the remote gateway.



Once the configuration is finished, the user can then initiate the remote access.
If the number is correct, the gateway:

- Off-hooks,
- Authenticates the user's identification information,
- Attributes the 192.168.2.13 IP address to the user.

The connection at the IP level is thus set up and the administrator can access the IP services present on the gateway knowing that the gateway address is 192.168.2.12

For example, the administrator can access the Web server of the gateway from a browser and the following URL http://192.168.2.12.

A remote connection of the gateway to an IAP can be summarized as comprising the following actions:

- The gateway and the modem initiate a telephone call to the IAP.
- The IAP off-hooks.
- The IAP authenticates the gateway's id information.
- The IAP attributes an official IP address (address accessible to any device connected on the Internet).
- The gateway connects to the SMTP server and sends its e-mail.
- The gateway on-hooks and returns to the WAIT mode.

If an error is detected (for example the line is busy), all the preceding actions are reiterated three times.

5.6.2 IPCallBack WAIT mode

When they use Internet technologies, (TCP/IP and other Web-based tools) users of Modbus Webdyngate gateways have access to new perspectives.

The basic idea is that through an Internet connection, one can control any device using standard tools and protocols.

Using this architecture the aim is to be able to control devices remotely using standard and sustainable tools (Internet browser, messaging) and protocols (TCP/IP, HTTP, FTP, SMTP, SNMP).

One of the major advantages of the Internet is to be able to use the world network as an infrastructure. The interest of using the Internet as a communication medium is basically economical. Thanks to the Internet, the cost of communication is limited to the cost of a local call.

To achieve this objective, the Webdyngate gateway must have an official IP address and the link between the gateway and the Internet must be permanent.

This type of architecture can be set up over an ADSL link, over a leased line or over cable.

However, it is not possible to keep a permanent link open on the STN at low cost.

In the previous hypothesis (WAIT incoming call mode) the connection was set up on a temporary point-to-point link (thus private) with the cost of communication depending on the distance between the two devices.

To use an Internet link and thus pay the local rates, the only way is to use an IAP.

If the Webdyngate is connected to its IAP via an STN modem, the problem is as follows:

When an administrator wants to reach the remote device, the device must already be connected on Internet and the administrator must know its official IP address.

The IPCallBack procedure is the function used to reach this status.

Through its link with the modem, the gateway can detect if an incoming call is in progress (ringing detection).

The principle of IPCallBack is to ''page'' the gateway when one wants to connect to the device.

When the gateway receives the page, it knows that a connection with it is sought but it does not off-hook.

When the ringing has ceased, the gateway connects to the IAP and retrieves an official IP address.

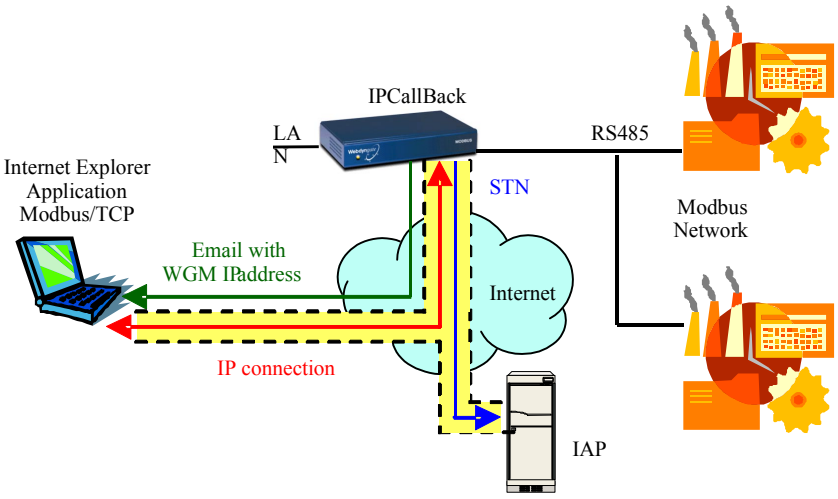
Once it has a valid IP address, the gateway maintains the connection and sends an e-mail to the programmed addressee with its IP address it has just retrieved in the body of its e-mail.

Thanks to this principle, the previously mentioned problem is solved.

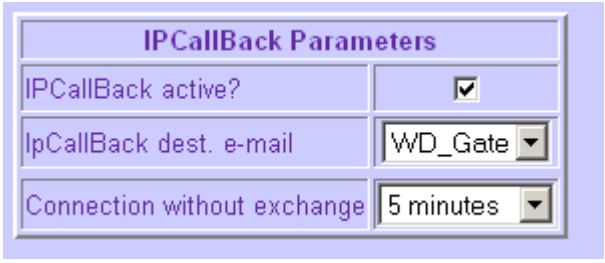
When the user receives the e-mail, he/she has the IP address of the remote gateway and since that gateway is connected, the IP link over the Internet can be set up.

Sending an e-mail is not deterministic, one never knows in advance how long it will take to be received by the user. To avoid overly long connection waiting time and high costs, once the e-mail is sent, the gateway initiates a timer. If at the end of a programmable amount of time the user is not connected, the modem on-hooks and the link with the IAP is disconnected.

Likewise, if the user forgets to disconnect after having communicated with the gateway, the latter detects the absence of dialogue and disconnects.



To set up the IPCallBack option, the configuration of e-mails and the messaging system must be performed beforehand and the following parameters must be filled in:



- option IPCallback active validated
- The e-mail address corresponds to the address of the user who wants to login. This address is selected from among the aliases already registered.
- The ''dialogue-free connection'' corresponds to the time during which the gateway waits for connection after sending the e-mail. At the end of this period, if the initiator of the ''page'' is still not connected or if no dialogue is present on the line, the gateway hangs-up. This option is also valid for the modem in mode.
-

On ''page'' reception, the e-mail sent by the gateway is of the type:

where the 62.147.157.203 address is the public IP address supplied by the ISP.

Once the connection is set up, a user can disconnect at any time.

This disconnection is activated using an HTML button, ''Modem Disconnection'' that appears dynamically in all the menus.

Pressing this button stops the call in progress and restores the Webdyngate to its initial status: waiting for a new outside call (WAIT mode).

6 Webdyngate maintenance

6.1 Introduction

The Modbus protocol is a master-slave protocol. Only one device can emit on the line at once. The master manages the exchanges and only it has the initiative. It successively interrogates each of the slaves. No slave can send messages itself without having been invited to do so.

On a Modbus network there can only be **a single Master.** However this master can interrogate **255** slaves.

On a Modbus network, the Webdyngate gateway always behaves like a master. Its role is to be able, spontaneously or after a request from a remote user, to collect or update a slave's variable.

To reach a variable, the gateway must thus have the address of the slave and the location of the Modbus variable in the slave's database.

A standard transaction is composed of a request sent by the master, followed by a response from the slave.

The message in either direction comprises the following information:

Address of peripheral	Function code	Data	Error detection data

- Each slave has a unique ''peripheral address''. This address always falls between 1 and 255.
- The Webdyngate gateway handles a subset of Modbus function codes.
- The data includes the parameters of the devices referred to by a 'parameters address'
- Initiating a call with a unique peripheral address triggers a response from the only peripheral with this address. This peripheral searches for errors, performs the task requested, then responds by giving its own address, the data and a check total.
- An error checking code is included at the end of each frame and enables the integrity of the data transmitted in both directions between a master and a slave to be verified.

The Modbus protocol specifies a set of functions that govern the dialogue between a master and a slave.

Each function has a code describing an action to execute.

The main Modbus communication functions are summarized in the table below:

Function code	Function
01 or 02	Read n bits
03 or 04	Read n words
05	Write one bit
06	Write one word
07	Quick status read
08	Looping
16	Write n words

Depending on the slave retained, all or some of these functions are supported. Before configuring the Webdyngate gateway, it is important to check which functions are supported by a slave.

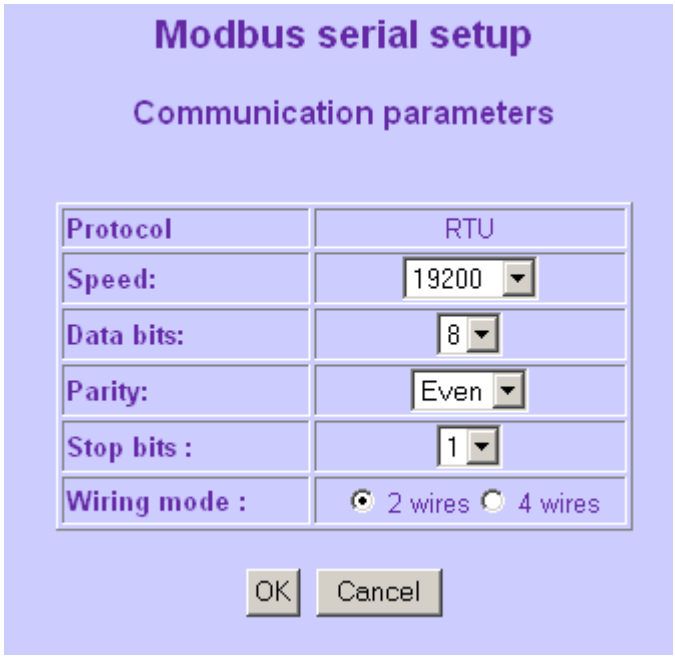
6.2 Configuration of the Modbus interface

At the physical level, the Modbus protocol uses an asynchronous serial port.

Different parameters must be set:

- Line speed (bauds)
- Number of data bits
- Parity
- Number of stop bits.

For two devices to be able to dialogue, these parameters must be identical on each side.



The speed may fall between 2400 bauds and 115200 bauds (by default 19200 bauds).

The number of data bits may be set to 7 or 8 (by default 8).

The parity can be activated or deactivated and if it is activated the parity can be odd or even (by default deactivated).

The number of stop bits can be set to 1 or 2 (by default 2).

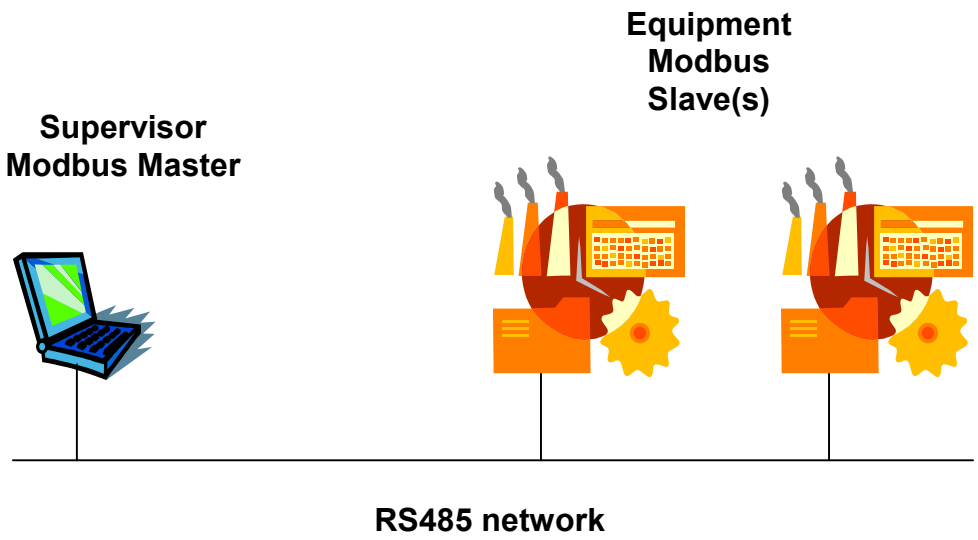
The cabling method allows the administrator to choose whether the gateway will operate in Modbus 4 wire or 2 wire mode.

These parameters are active for all the slaves connected on the same bus. The first step involves checking the coherence of all the devices.

6.3 Redundance

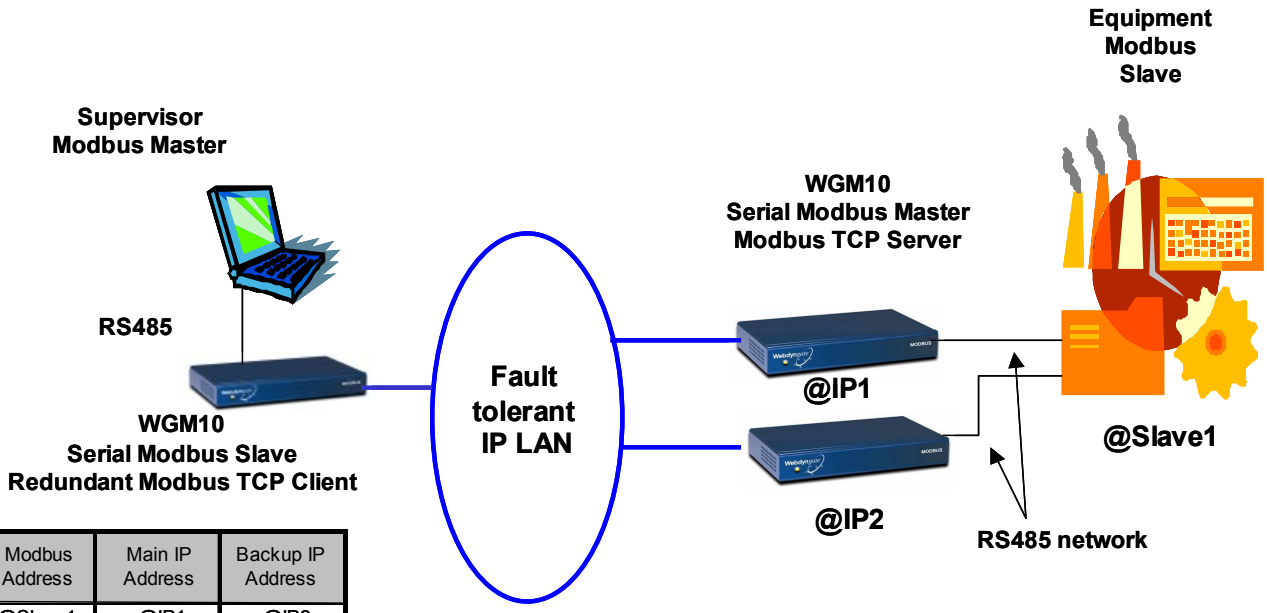
Process control using industrial networks automatically implies an extremely high degree of availability together with some degree of fault tolerence or redundance if possible. In order to provide this level of availability it may be necessary to duplicate some of the elements of the solution architecture. The use of a fault tolerant IP network (usually fiber ring) together with the redundant option for the gateway model WGM10 allows the creation of a totally fault resilient network infrastructure.

The initial Modbus control application looks like the image below :



A PC is configured as the Modbus master and is connected to the RS485 using a suitable network card. The Modbus slaves are all connected to this RS485 bus. Any failure of this bus (disconnected cable, bad connection …) or of the PC or its network adapter will cause the application to be inoperational. This is unacceptable for many industrial applications.
The Webdyngate redundant mode objective is to enable existing Modbus RS485 applications (without any PC application modification) to be able to be configured for

use over fault tolerent IP network infrastructures. The basic architecture and iits elements is illustrated below :



Modbus Address	Main IP Address	Backup IP Address
@Slave1	@IP1	@IP2

In the above configuration the WGM10 gateway that is connected to the PC based supervisor is configured in Modbus serial slave and Modbus TCP client mode. The WGM10 gateways that are connected to the Modbus equipment are configured in serial Modbus master and Modbus TCP server mode.

In addition to the standard gateway function the WGM10 also supports a special feature enabling the Modbus TCP client gateway to define a primary and secondary network "path" to reach an equipment.

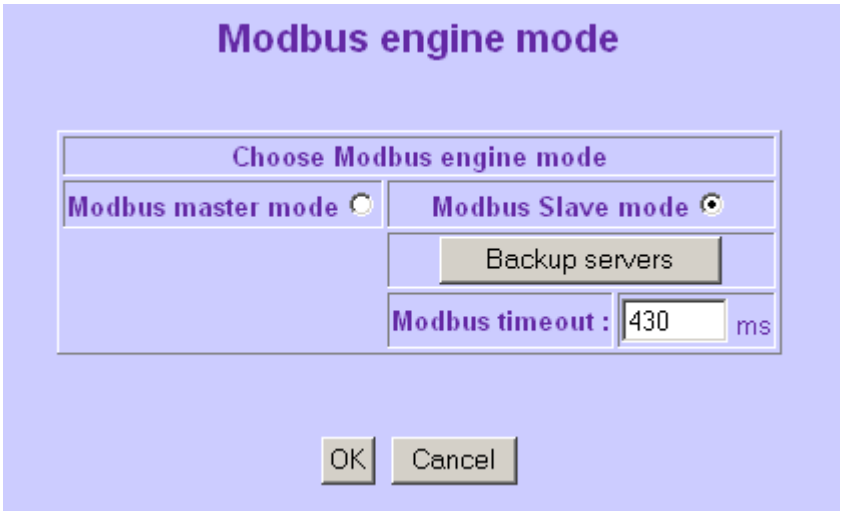
The Modbus connected equipment must be equipped with two serial Modbus interfaces that will be configured with the same Modbus slave address "@slave1". The redundancy feature operates as follows :

When the Modbus serial slave WGM10 receives a data frame, the contents of the frame are analysed. Based on the destination Modbus slave address, the gateway is able to identify two IP addresses : the main WGM10 Modbus TCP server and the backup WGM10 server.

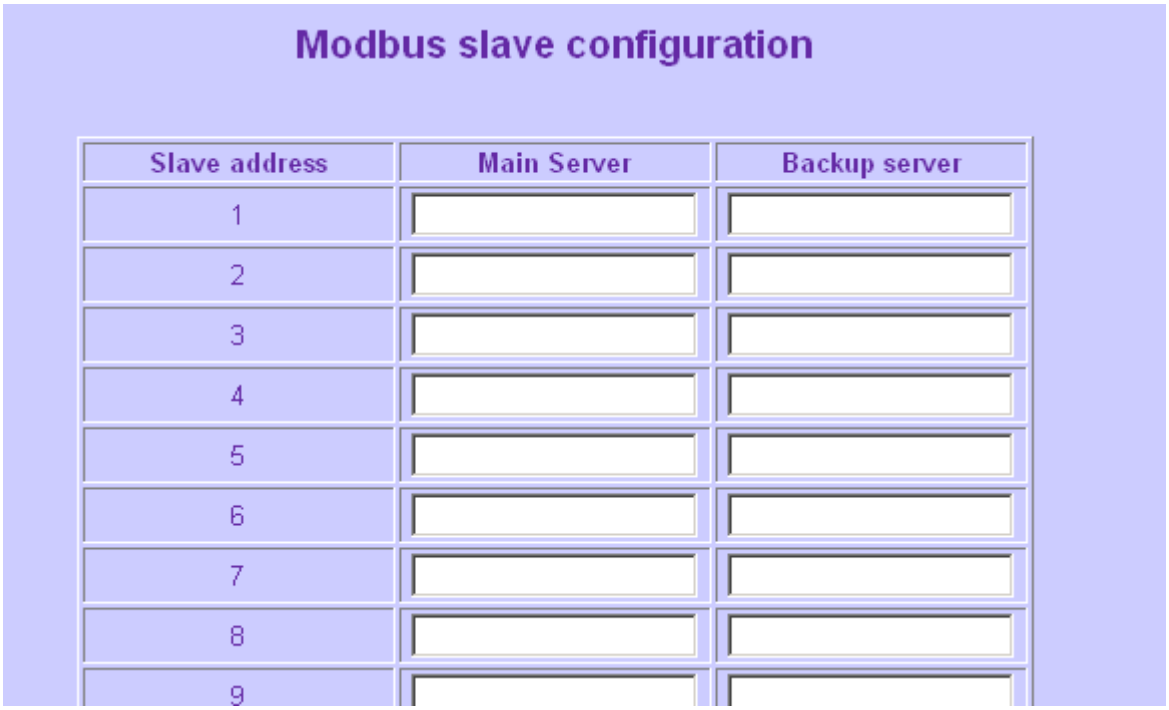
The decision as to which WGM10 server to use is made by the WGM10 configured as the Modbus TCP client. An example of how this mechanism works is described here : Imagine that the frame sent by the supervisor application to the @slave1 address doesn't respond within the time limit. In this case the WGM10 client automatically tries to reach the @slave1 target by using the WGM10 Modbus TCP server configured as the backup. Assuming that this "route" is successful then this becomes the normal route and the old backup Modbus TCP server is marked as

being the main server. This "toggle" effect enables both routes to be exploited without any manual intervention.

The Modbus mode menu is used to configure whether the WGM10 is to be configured in serial slave or serial master mode. If slave mode is selected then you will have access to the redundance configuration menu (shown as Backup servers). Note : If you change the configuration from slave to master or vice versa then you must reboot the gateway.



Clicking on the Backup servers button gives access to the following screen :



This table is where the the association is made between a serial Modbus slave address and the main Modbus TCP server and the backup Modbus TCP server. The information requested for each of the fields is the IP address of the WGM10 gateway.

6.4 Identification of Modbus variables

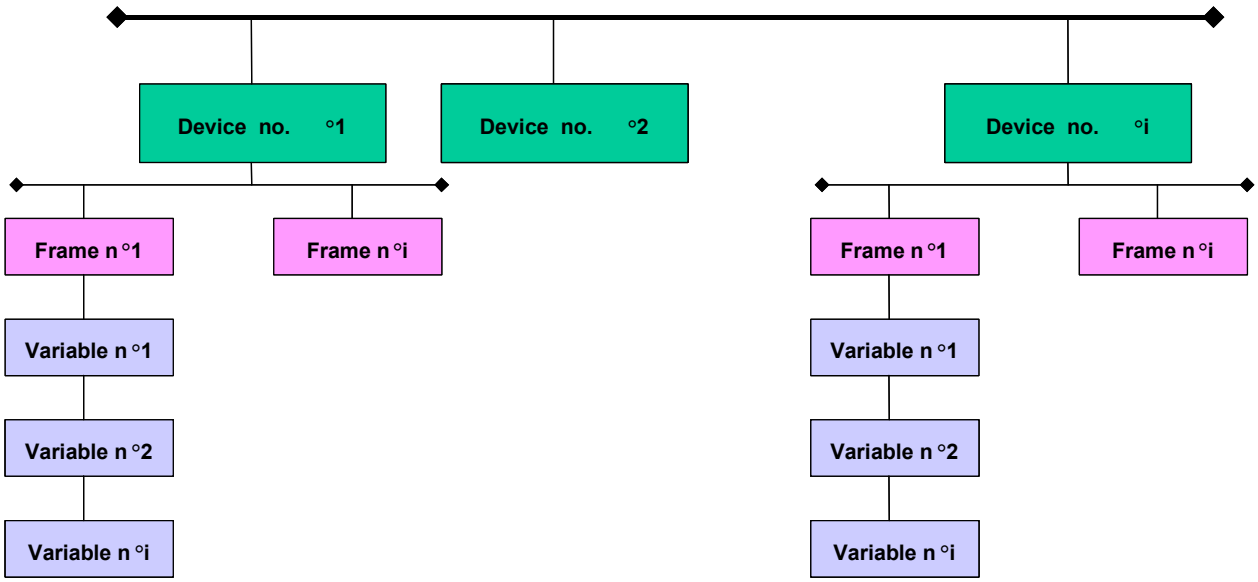
A serial Modbus network can be composed of N number of devices, each device represented by a database in which the variables that can be read or changed (writing cycle) are stored.

The role of the master is to unite all these bases into a single base. This single base must be created on a master by identifying each variable entering the base.

The variables are identified using the address of the device in which the variables are stored as well as the positioning of the variable in the database of the concerned device.

To facilitate the identification of each variable, the Webdyngate database is configured in three steps:

- Identification of the device.
- Identification of the Modbus frames associated with the device.
- Identification of the variables associated with each frame format.



The notion of ''frame'' was added to optimize traffic on the Modbus serial link. In slave Modbus equipment, the database is represented by a set of memory cells (a cell is equivalent to one word or one bit), each cell being associated with an offset (its address).

The address of a memory cell (coded on 16 bits) is between 1 and 65536.

Depending on its type, a variable can use several cells and several cells can be stored consecutively in the base.

When a master Modbus sends a request to a slave, the request (whatever the function, read or write) is composed of:

- The address of the slave,

- The offset of the memory cell in the base,
- The number of memory cells concerned by the request.

In each read frame (bits or words), the data field contains the address of the first bit or word to read and the number of bits or words to read.

For each write frame, the data field contains the address of the first bit or word to write, the number of bits or words to write as well as the values of the bits or words to write.

The Modbus protocol ignores the notion of variables. Its decoding is based on a set of memory cells identified by an offset and a quantity of memory cells.

A Modbus request can thus point to a variable or set of consecutive variables.

It is up to the user to manage his/her variables singly (one variable equals one frame) or grouped (one frame equals n number of consecutive variables).

When there is a significant number of devices and variables, it is important to use the frame option.

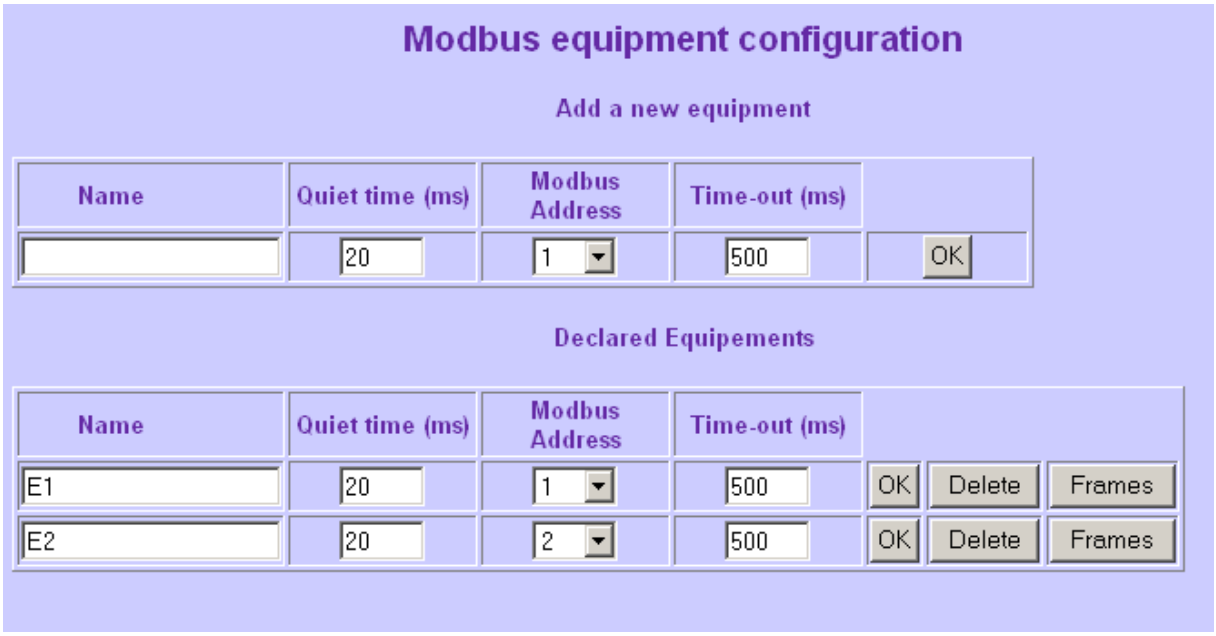
Some devices do not allow variables to be grouped. This can be checked by analyzing the functions supported by a slave.

Function code	Function
01 or 02	Read n bits
03 or 04	Read n words
05	Write one bit
06	Write one word
07	Quick status read
08	Looping
16	Write n words

In writing for example, if function code 16 is not supported, and only function 06 is possible, a frame will necessarily be associated with a single variable in the write/read mode.
All the variables of a frame configured in the gateway have the same read or read/write rights.

The first step involves identifying the device in which the variable is stored.

The '' **Modbus equipment configuration''** HTML page allows all the devices already registered by the gateway to be visualized and also allows modifying, adding or deleting a device.



Modbus equipment is identified by its name and Modbus address.

The Modbus address, between 1 and 255, is a parameter internal to the device. Depending on the equipment involved, this parameter can be set manually (code wheel) or using a computer terminal. This configuration is independent of the Webdyngate gateway. The Modbus network administrator must perform it beforehand. It is important to check that two devices do not have the same Modbus address.

There is no constraint on the attribution of addresses as far as the Webdyngate is concerned. It is possible on a network with two devices that the first have address 32 and the second address 254.

The administrator is free to choose the **Name** of the device. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that is used to identify the device.
Two devices with the same name will be accepted but displayed in red to signal the risk of error.

The **Quiet time** and **Time-out** are two technical characteristics related to the device.

The **Quiet time** is the minimum time a master must wait between two requests to the same device (by default 20ms). By definition a slave Modbus device allocates a certain amount of time to processing the Modbus requests it received from a master. If these requests are too close to each other in time the Modbus network can become a bottleneck for this device. The turnaround time is thus set up to allow a slave Modbus device to process all its tasks equally. To handle this time, the Webdyngate has a queue in which all the requests addressed to a slave are stored. The queue is emptied with respect to the turnaround time.

The **time-out** (by default 500ms) is the maximum amount of time in which the slave must respond to a request from master. If this time is exceeded, the master considers

that the slave is absent. It is highly recommended to keep this value above 50 ms or else the equipment may not have the time to respond before being considered as having timed out.

Each new or old device is linked to a line of the form and only one line can be modified at a time.

To validate a line, you must click on the OK button and only the line concerned by that button will be taken into account by the gateway.

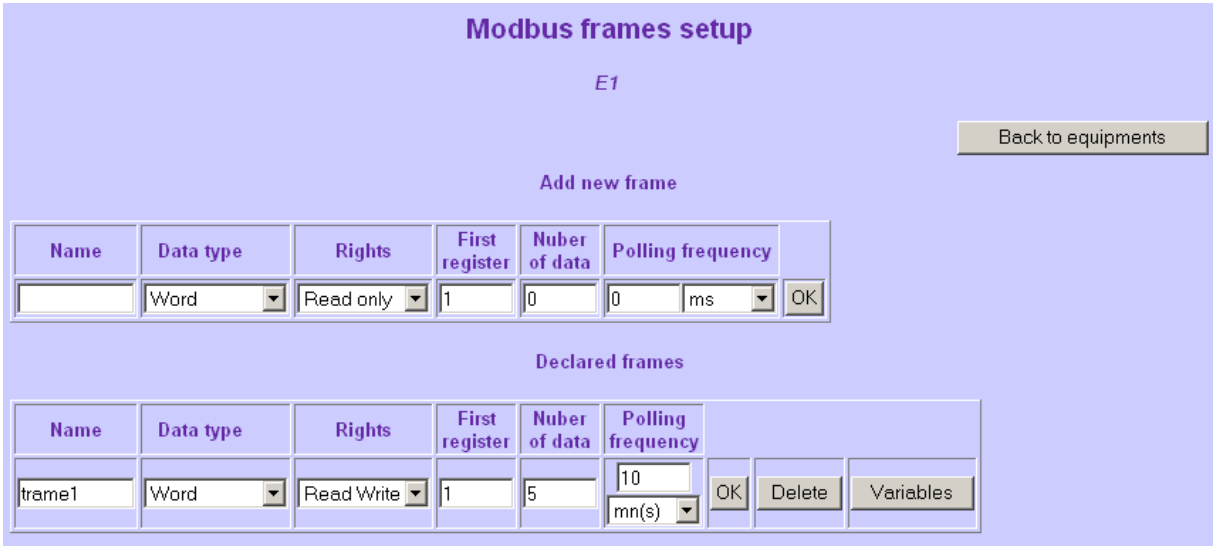
Two identical names are accepted but displayed in red to signal the risk of error.

If a modification is necessary on several devices, the operation must be repeated as many times as there are devices being modified.

The addition of a new device (first part of the form) causes the registration of the device. Once registered, the device appears in the list of declared devices.

The passage to the **second step** is achieved device per device.

The **"Modbus frames configuration''** HTML page is used to visualize all the frames already registered by the gateway and to modify, add or delete a frame.



Pressing on the ''**Frames''** button causes the opening of a new page where all the frames related to a device and their parameters are recorded.

All the frames declared in this page are associated with the device selected in the first step. The name of the device appears on the second line of the page. In our example, the device being modified is E1.

Each new or old frame is linked to a line of the form and only one line can be modified at a time.

To validate a line, you must click on the OK button and only the line concerned by that button will be taken into account by the gateway.

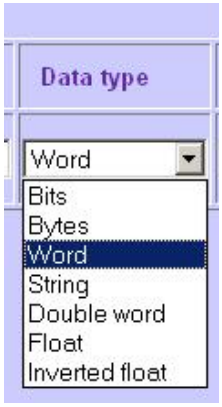
If a modification is necessary on several frames, the operation must be repeated as many times as there are frames being modified.

The parameters related to a frame are :

- the **Name** of the frame
- the **Type of data** contained in the frame
- the **rights** associated with these data
- the address of the **First Register** where the data is stored
- the **number of data items** processed
- the data **scanning frequency**

The **Name** of the frame is left up to the administrator. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that is used to identify the frame.

The **Type of data,** in accordance with the Modbus protocol, can by of the bit, byte, word, character, double word or floating type.



The "float" type corresponds to a double word of 32 bits in the Intel format.
The "inverted float" type is the same as the floating type except that the order of the 2 words is changed. The order of bits inside each word is not changed.
The "string" type allows the creation of a frame that contains a single variable that itself contains as many characters as have been configured in the "data length" setting.

The **Rights** associated with a data item can be read only or read/write.

The **First Register** identifies the offset of the first memory cell in the device database that will be dealt with. This offset can be between 1 and 65535.

The **Number of data items** counts the number of memory cells processed. If the type of data is bit, this field counts the number of bits processed, if the type is byte, this field counts the number bytes processed.

It is important to note that a frame is always composed of a set of data that are all of the same type and that have the same rights.

The maximum size of the gateway database is 1500 variables.

Each frame may contain upto 256 bytes (i.e. 2048 bits, 128 words or 64 double words).

The master unites all the variables from the Modbus slaves. When an administrator displays the status of a variable using an Internet browser, it is the master database that is interrogated. It is thus important to know the frequency of variable status or content refreshing.

It is the **Scanning frequency** (in seconds) that sets this frequency. It can be between 0 ms and 24 days.
If this frequency is worth 0, with each user request the variable pointed to is refreshed and the value displayed is indicated in real time.

The choice of frequency is a compromise between:

- the occupation of the Modbus bus. The lower frequency, the greater the number of variables and slaves and the more the occupation of the bus is great. Depending on the response time of a slave, a Modbus request (interrogation and response) takes between 30ms and 200ms. In the worst cases, we see that the interrogation of 5 variables with a frequency of 1 second saturates the bus.
- The display time of an HTML page. If the scanning frequency of the variables is nil, with the same assumptions as above, the display of an HTML page including N number of variables will take n*200ms.

The ''Modbus frame configuration'' HTML page allows all the frames of a device to be visualized by the gateway and also allows modifying, adding, or deleting a frame.

Each new or old frame is linked to a line of the form and only one line can be modified at a time.

To validate a line, you must click on the OK button and only the line concerned by that button will be taken into account by the gateway.

If a modification is necessary on several frames, the operation must be repeated as many times as there are frames being modified.

The third and last step involves specifying the parameters of a variable in a frame of a given device.

The passage to this step is achieved using the ''**Variables''** button in the ''**Modbus frames configuration''** page.

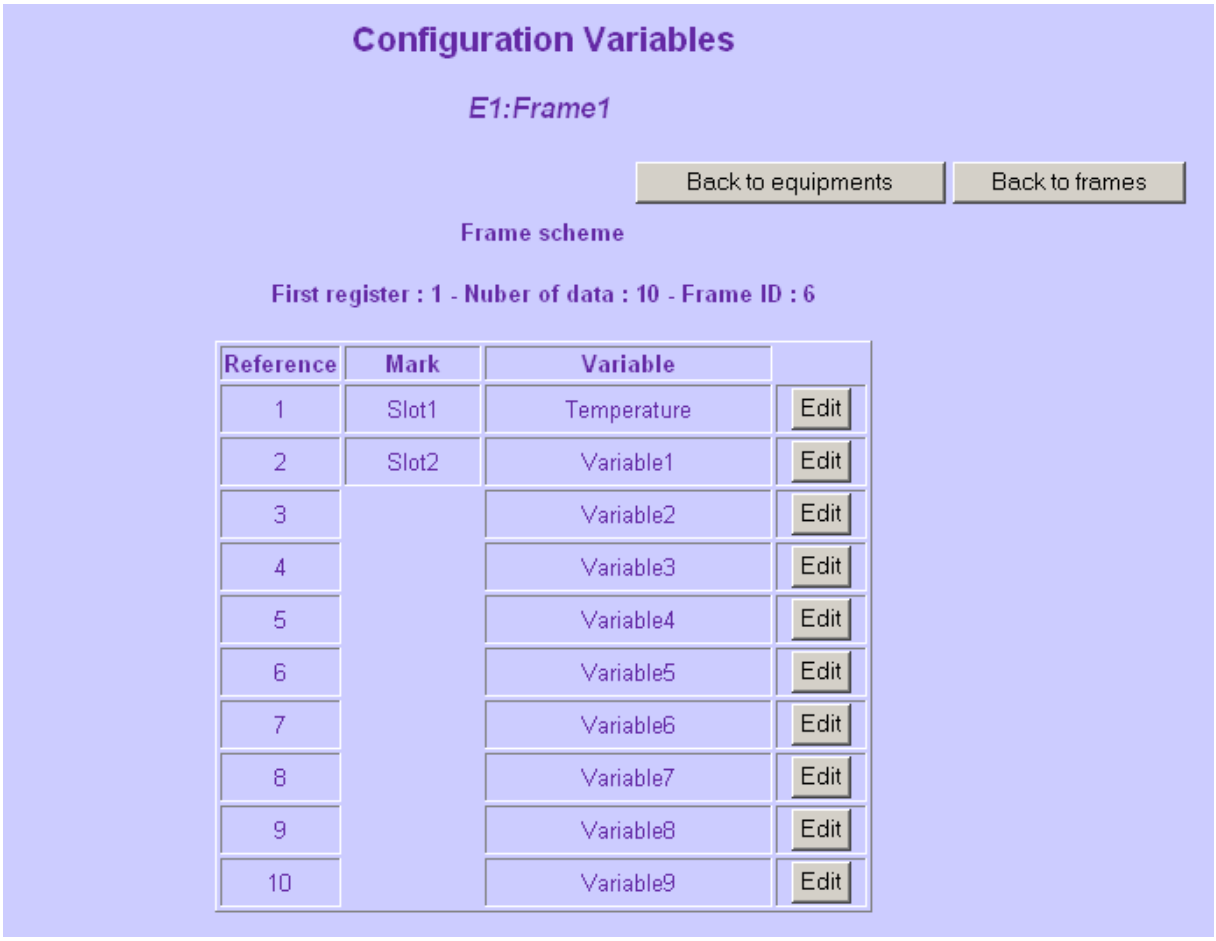
This page lists all the variables contained in a frame being modified.

Based on the specification of a frame, the gateway knows the number of variables composing the frame.

In example, ''Frame1'' is composed of 10 variables.

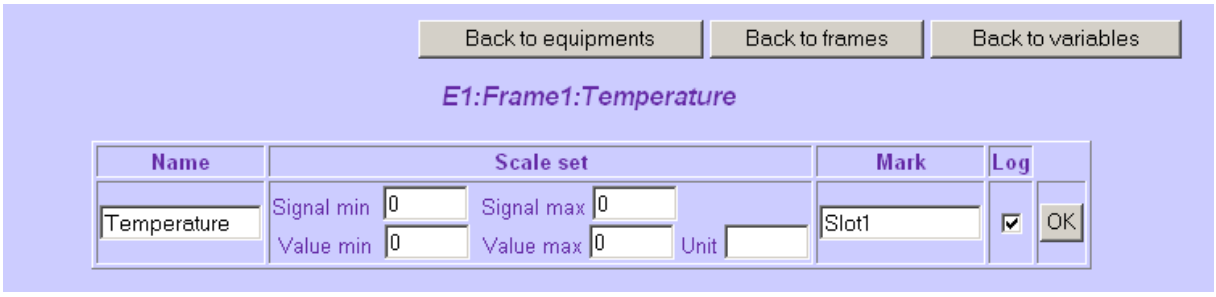
The '' Variables Configuration'' page is used to specify each of these ten variables.

By default, a variable has a factory configuration and can keep it.



In the example presented, of the ten variables, the first and second were modified: name and mark for the first, mark for the second. All the other variables (variable2 to variable9) are the default variables.

A variable is modified using the **Edit button.**



The **Name** of the variable is left up to the user. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that will be used to identify the variable.

The **unit** of the variable is a string of 5 characters that allows the content of a variable to be specified. This data is not used by the gateway and it is only called up in the operation pages. This field is optional and can be left empty.

Scale set is used to scale the data from a device.

- Signal is the data read directly on the device.
- Value is the data scaled.

By definition, scaling is done linearly in the form of an y=ax+b equation.

The assumptions used to calculate the slope and origin of this line are:

- If x= min Signal y= min Value
- If x= max Signal y= max Value

We infer that:

- a=(Max value - Min value)/(Max signal - Max signal)
- b=(Min value) – a* (Min signal) or b=(Max value) – a*(Max signal)

Let us suppose for example that the Webdyngate gateway is connected to a Modbus thermocouple that delivers a value between 0 and 5 according to the temperature.

With this assumption: ''Min signal=0'' and ''Max signal=5''.

Based on the thermocouple calibration curve, we know that the signal 0 corresponds to a temperature of 20°C and that signal 5 corresponds to 120°C on this curve, we thus infer that: ''Min value=20'' and ''Max value=120''.

Based on these assumptions, we can infer that the slope and origin of the line:

- a=(120-20)/(5-0)=20
- b=20-(20*0)=20

Based on a Modbus request:

Scaled value = (value read) * 20 + 20 (y=ax+b).

If the data read = 2 we infer that the scaled value = 20 * 2 + 20 = 60°C

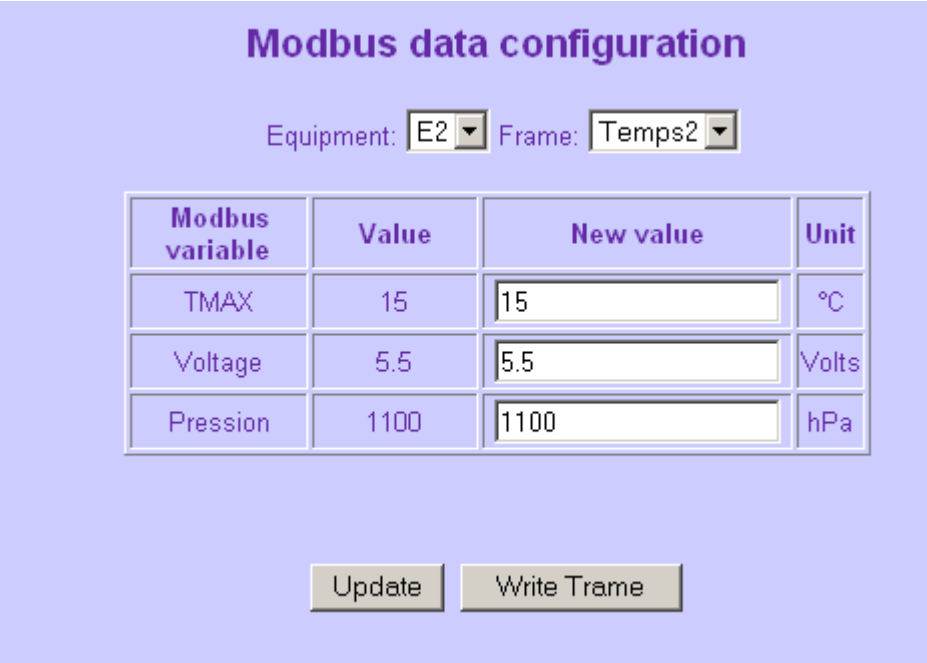
If the data read = 0 we infer that the scaled value = 20 * 0 + 20 = 20°C (Min)

If the data read = 5 we infer that the scaled value = 20 * 5 + 20 = 120°C (Max)

The **Log** box allows a trace or the name of a variable to be kept.

If this box is checked, periodically, according to the scanning frequency (frame configuration) the name of the variable, the time and date and the value of the variable are stored in the log.

6.5 Modbus Data



Depending on its rights, a Modbus variable can be ''read only'' or ''read/write''.

By definition, the variables are united in a frame, a frame including variables of the same type and with the same rights.

In a first step, it is thus compulsory to select the frame in which one or more variables must be modified. This selection is achieved first by choosing the device associated with the frame concerned. Finally, when the frame is selected, a table including all the variables associated with the frame is displayed.

This table lists:

- The name of the **Modbus Variable**
- Its **current value**, scaled
- An entry box in which the **New value** of the variable, scaled, can be entered if the frame has write rights.
- The **unit.**

One or more variables can be modified at the same time. If a variable is not modified, its current value is preserved.

Pressing on the ''**write frame''** button instantaneously sends the frame to the selected device. Beforehand, if scaling is necessary, the Webdyngate gateway converts the values entered to the device format. This button only appears if the frame has write rights.

6.6 Alarms Configuration

6.6.1 Introduction

In Webdyngate terminology, an alarm is linked to an event (the alarm trigger) and to an action (the consequence of the alarm).

Two types of triggers can be implemented:

- Triggers related to a Modbus variable.
- Triggers linked to the gateways dry contact inputs.

6.6.2 Configuring the alarms

On the first page we find the exhaustive list of alarms declared as well as two buttons used to add new alarms whose trigger is a Modbus variable or a dry contact input.

An alarm and its description correspond to each line in the list. Three buttons are used to handle these alarms:

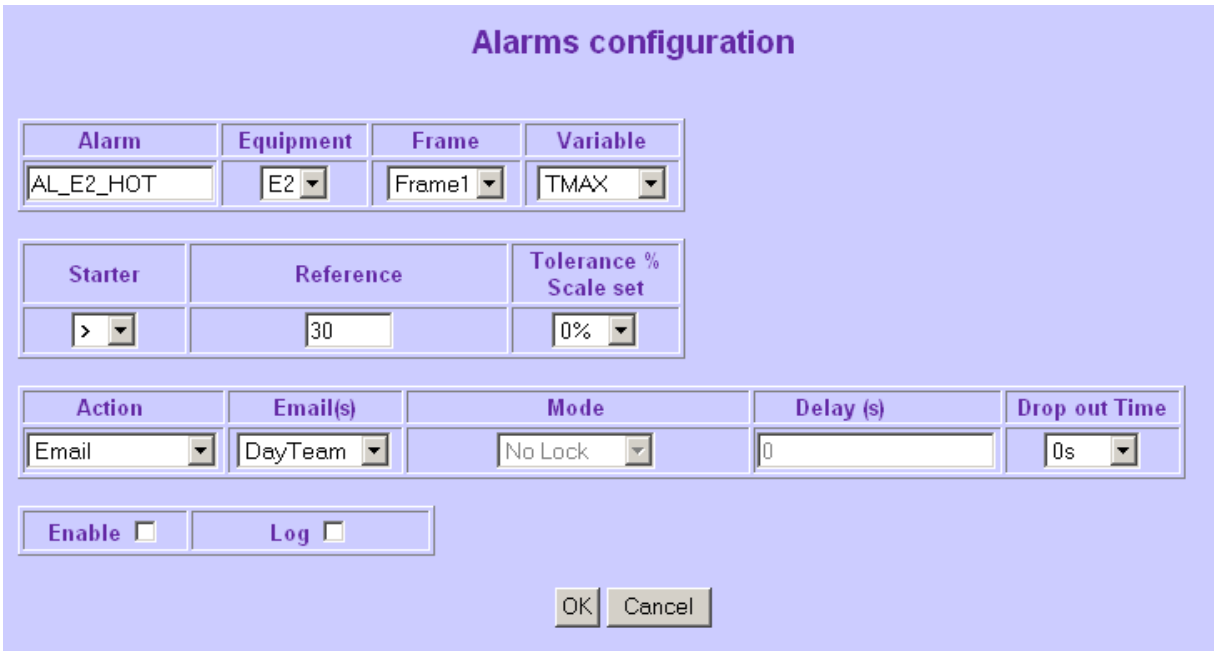
- The OK button validates any modification in the activation or deactivation of the alarm. If the box is checked, the alarm is active. If the box is empty the alarm is deactivated but is still part of the Webdyngate database.
- The Erase button is used to delete an alarm.
- The Edit button is used to modify the components of an alarm.

An alarm is configured (edition or addition) according to the trigger: Modbus variable or dry contact input.

Pressing the Edit button opens a new page containing all the parameters of the alarm being edited.

6.6.3 Modbus Variable as an Alarm Trigger

Addition (Add new alarm on device button) or edition (Edit button pointing an alarm with a Modbus variable as a trigger) opens a new page listing all the parameters related to an alarm.



An alarm is identified by its name.

The name of the **Alarm** is left up to the user. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that is used to identify an alarm.

First, a variable associated with the alarm must be selected.

If the ''**Variable''** comes from the Modbus base, the ''**Device''** and ''**frame''** selection lists must be filled in before hand.

Each selection updates the following selection.

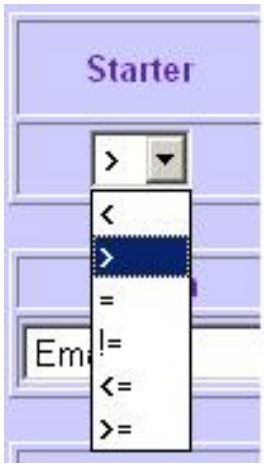
For each device, only the associated frames will be selected and for each frame only the associated variables will be selected.

Once the variable is selected, the alarm triggering conditions (operation and reference) must be specified.

In any case, the operation executed is the comparison of the variable selected with a reference.

Six operations are possible:

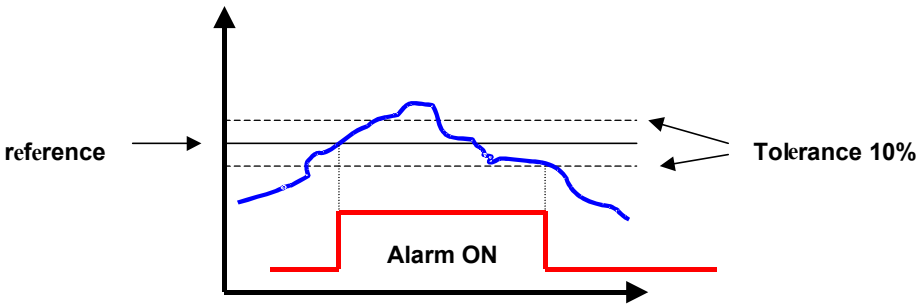
- Less than (<)
- Greater than (>)
- Equal (=)
- Different (!=)
- Less than or equal (<=)
- Greater than or equal (>=).



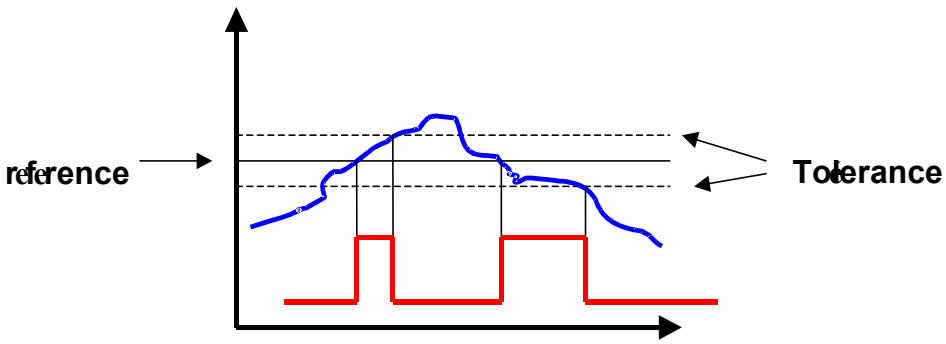
The **Reference** is the value the variable will be compared to. This reference must be of the same type as the variable and scaled.

The **Tolerance** parameter filters the oscillations of a variable around its reference thus avoiding bursts of alarms.

Two examples are given below to show the tolerance effect:



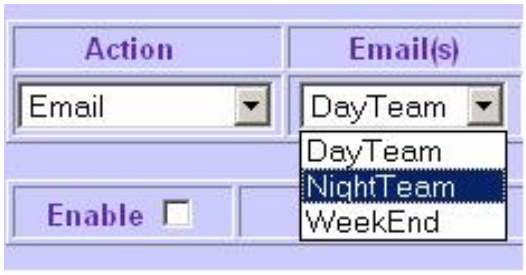
Alarm if variable>reference with a tolerance of 10%



Alarm if variable=reference with a tolerance of 10%

Four types of action are possible with an alarm:

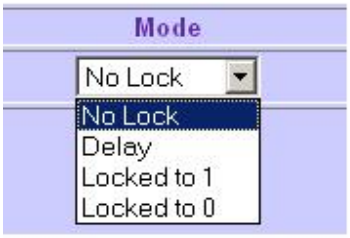
Sending one or more **E-mail(s)** (mail) causes the connection to the messaging server, the sending of the e-mail with in its body the name of the alarm, the date and time and the value of the variable having triggered the alarm.



The destination e-mail of the alarm is selected in the ''e-mail(s)'' list. This lists includes all the declared aliases.

The **Switch Output 1 (on-off)** and **Switch Output 2 (on-off)** activate one output among the two available.

This activation can follow several ''**Modes''**:



Contact not locked
The output is active as long as the alarm is active. In this configuration the status of the output selected reflects the status of the alarm.

Delay
In this configuration the output is active for a period given in the time box.

Locked at 1
The output is activated and remains so whatever the time and status of the alarm.

Locked at 0
The output is deactivated and remains so whatever the time and status of the alarm.

Among other possibilities, the lock mode allows an output to be managed on two different alarms configured with two different triggers and an identical action.

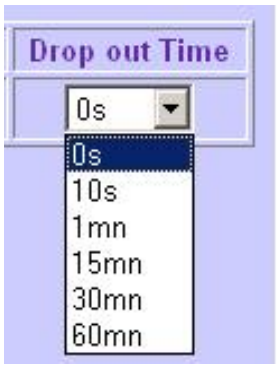
The only difference concerning the action is that one alarm uses the locked at 1 mode and the other alarm uses the locked at 0 mode.

In this configuration the status of the output will always be the reflection of the last of the two alarms triggered.

The **IPCallBack** action will cause an email to be sent to a predefined list of destination addresses (see chapter 4.6.2). This is the case even if IPCallBack is not acvtivated in the "modem in" menu.

In this particular case, the body of the e-mail includes, in addition to the alarm description, the public IP address supplied by the IAP and once the e-mail is sent the connection with the IAP is maintained.

The **Drop out time** is the time during which an alarm is deactivated once triggered.



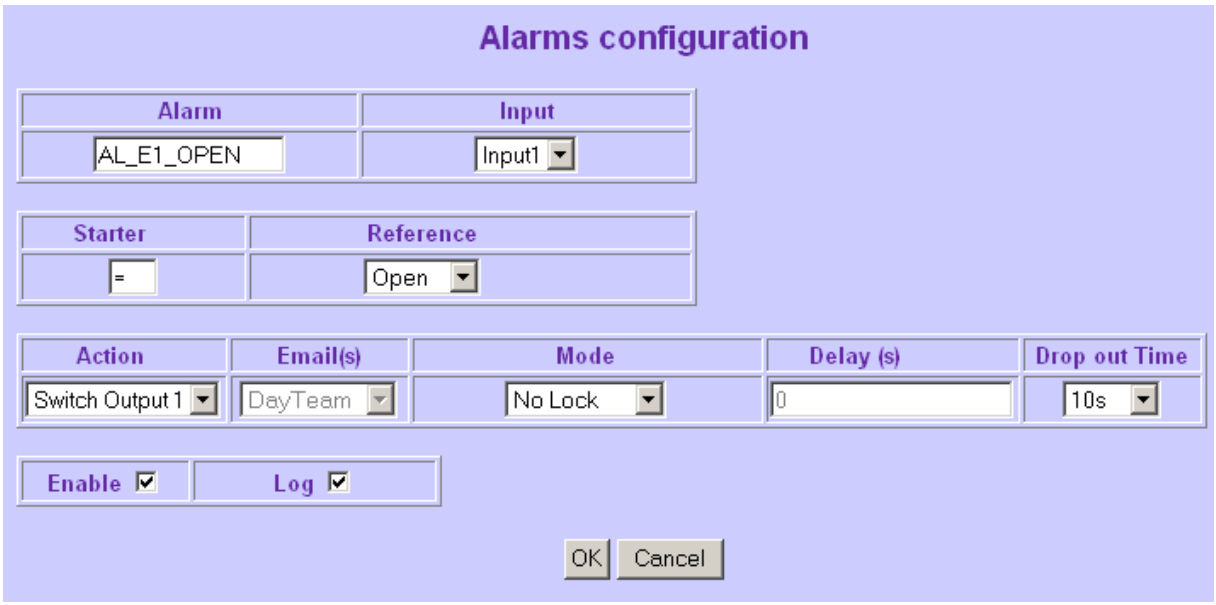
This delay allows alarms arriving in bursts to be filtered.

The last two check boxes specify:

- If the alarm is active
- If the alarm must be traced in the alarms log.

6.6.4 Webdyngate input as the alarm trigger

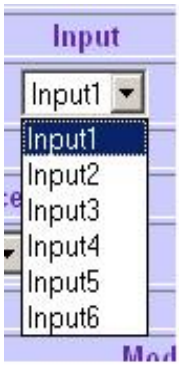
Addition (Add new input alarm button) or edition (Edit pointing an alarm with a Webdyngate input as a trigger button) opens a new page listing all the parameters related to an alarm.



An alarm is identified by its name.

The name of the **Alarm** is left up to the user. It can be composed of a string of at most 30 characters. In all the other configuration and operation pages, it is this name that is used to identify an alarm.

The Webdyngate gateway has 6 dry contact inputs. The input is selected in the **''input''** list:

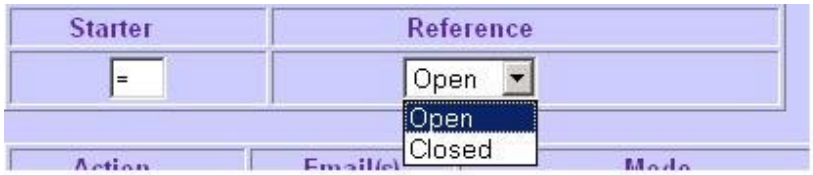


Once the input is selected, the alarm trigger conditions (operation and reference) must be determined.

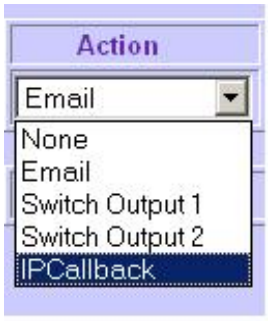
In any case, the operation executed is the comparison of the input selected to a reference.

Two cases are possible:

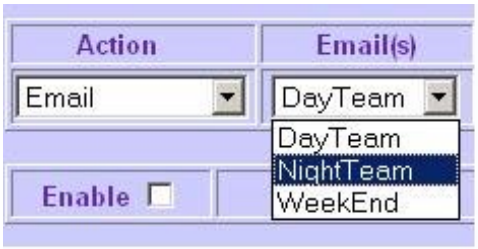
- The input is **Open**
- The input is **Closed**



Four types of action are possible with an alarm:



Sending one or more **E-mail(s)** (mail) causes the connection to the messaging server and the sending of the e-mail with in its body the name of the alarm, the date and time and the value of the variable having triggered the alarm.



The destination e-mail of the alarm is selected in the ''e-mail(s)'' list. This lists includes all the declared aliases.

The **Switch output 1(on-off)** and **Switch output 2 (on-off)** activate one output among the two available.

This activation can follow several ''**Modes''**:

No lock
The output is active as long as the alarm is active. In this configuration the status of the output selected reflects the status of the alarm.

Delay
In this configuration the output is active for a period given in the time box.

Locked to 1
The output is activated and remains so whatever the time and status of the alarm.

Locked to 0
The output is deactivated and remains so whatever the time and status of the alarm.

Among other possibilities, the lock mode allows an output to be managed on two different alarms configured with two different triggers and an identical action.

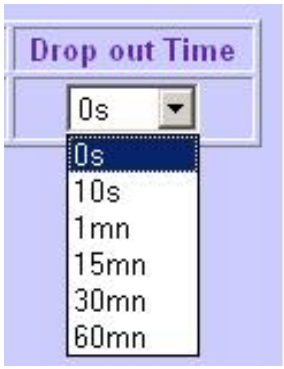
The only difference concerning the action is that one alarm uses the locked at 1 mode and the other alarm uses the locked at 0 mode.

In this configuration the status of the output will always be the reflection of the last of the two alarms triggered.

The **IPCallBack** action will cause an email to be sent to a predefined list of destination addresses (see chapter 4.6.2). This is the case even if IPCallBack is not activated in the "modem in" menu.

In this particular case, the body of the e-mail includes, in addition to the alarm description, the public IP address supplied by the IAP and once the e-mail is sent the connection with the ISP is maintained.

The **Drop out time** is the time during which an alarm is deactivated once triggered.



This delay allows alarms arriving in bursts to be filtered.

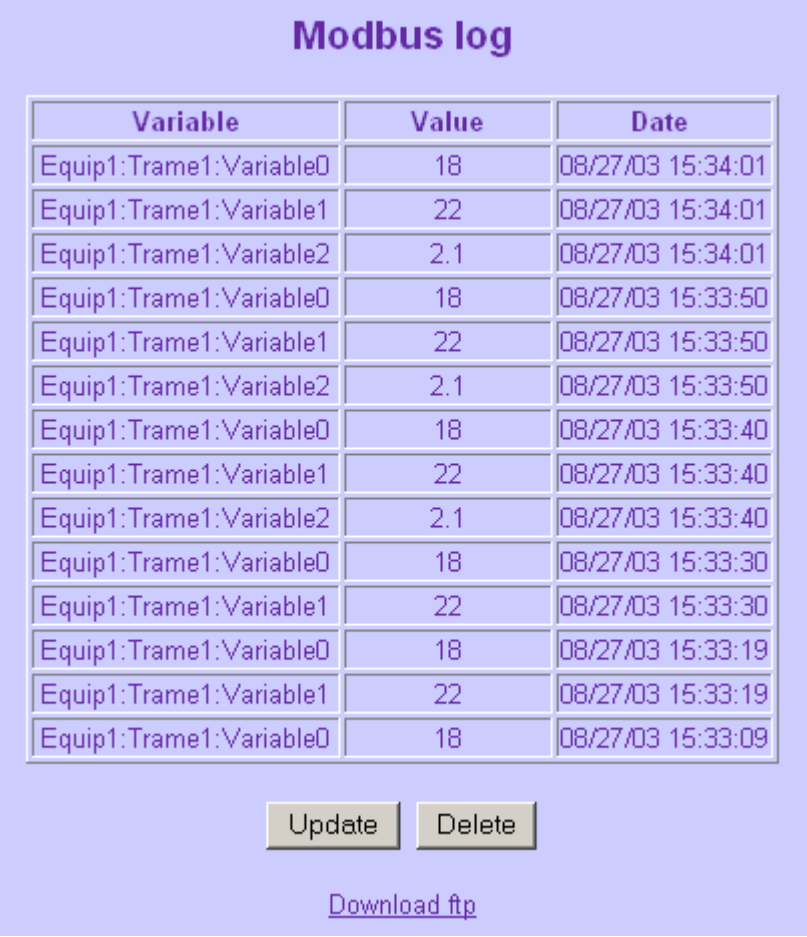
The last to check boxes specify:

- If the alarm is **active**
- If the alarm should be traced in the alarms log (**''Logging'')**

6.7 Configuration of logs

Three logs are managed by the Webdyngate gateway:

- The Modbus variables log.
- The alarms log.
- The email log.



All the Modbus variables can be traced in the log if their scanning frequency is different from 0. When the Modbus variables are configured, the user is free to incorporate a variable in the log or not. The Modbus log is a synchronous log: events are always traced at a set date.

The log is saved on the flash disk every 3 minutes or every 1000 lines. When it reaches its maximum size (2MB), it is compressed to the gz format and sent by e-mail to the life signal addressee if the ''log as an attachment'' box is checked. Once sent, the Modbus.log file is emptied.

A "**Delete**" button purges the log.

Le lien « Télécharger ftp » permet de télécharger directement le fichier sur le pc sans avoir à faire de manipulations sur la flashdisk.

The alarms log traces all the events (appearance and disappearance) occurring in relation to the alarms. The alarms log is an asynchronous log. A recording is written at the same moment as the alarm occurs.

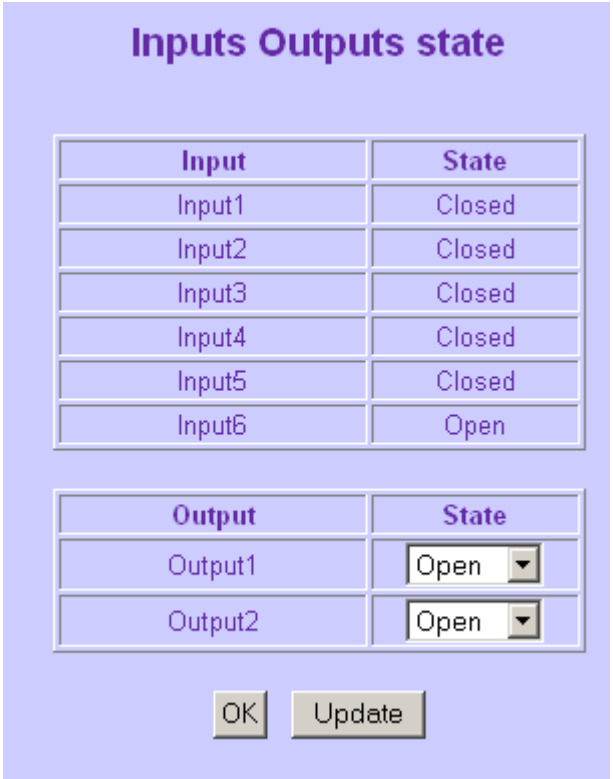
The alarm and email logs are managed on a FIFO basis. When they reach their maximum size (100 lines/entries long) then the oldest entries are deleted and replaced by any newer events that must be logged.

6.8 Input – Output configuration

The gateways : WGM20(M) & WGM30(M) each have 6 digital inputs and 2 relay based outputs.

It is possible to visualize the status (open or closed) of each input.
It is also possible to both visualize and change the status of the outputs.

This Web page will be refreshed every minute. You must click on the "OK" button in order to validate the change of state.



7 Utilisation

7.1 Modbus Status

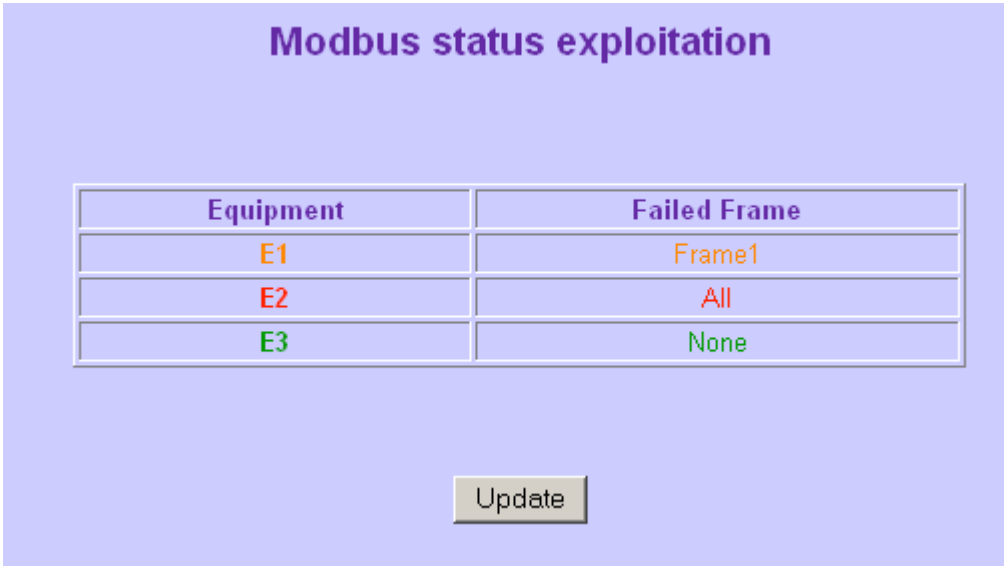
The Modbus status page indicates the status of the communication (calls) between the master (Webdyngate) and the slaves (Modbus devices).

All the calls between the master and a slave are formatted in frame form. Each frame groups a certain number of variables.

The quality of a call between a master and a slave can be measured on the basis of the frames exchanged between both units.

Three quality levels are possible, device per device or slave per slave:

- Quality OK (green): all the frames declared for a device are recognized and processed by the slave.
- Quality mediocre (orange): some frames declared on the master are not recognized by the slave.
- Quality not OK (red): none of the frames declared on the master are recognized by the slave.



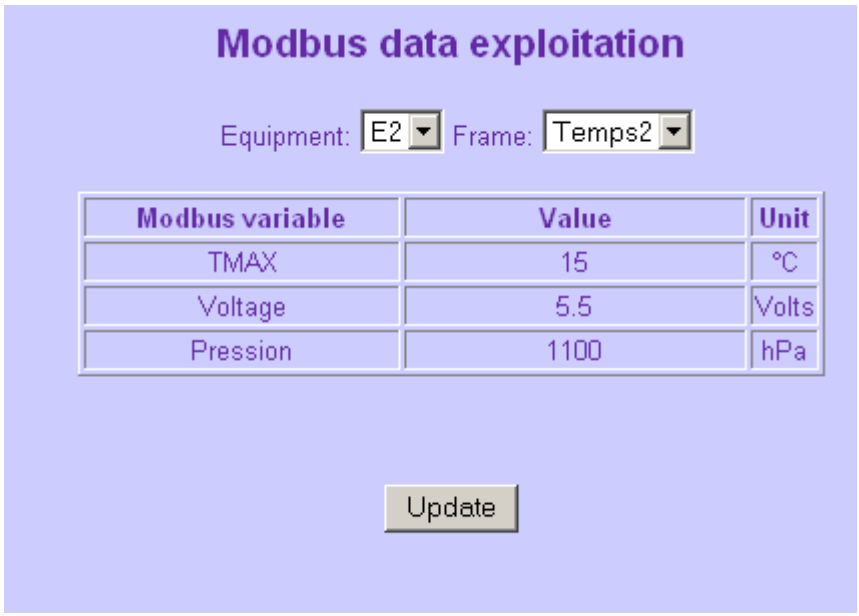
For a device showing mediocre quality (orange) the HTML page specifies which frames are defective.

The Modbus status page was created to help an administrator fine tune his/her gateway. In particular, it allows the errors related to the addresses of the variables and devices to be corrected.

A "Refresh'' button updates the data.

7.2 Modbus Data

The value of each variable for each device can be visualized on the "**Modbus data exploitation''** page.



This visualization occurs for each device, frame per frame.

The selection at the top of the page is used to select the device and frame to be processed.

A table is generated dynamically. It includes all the variables associated with the device and the frame.

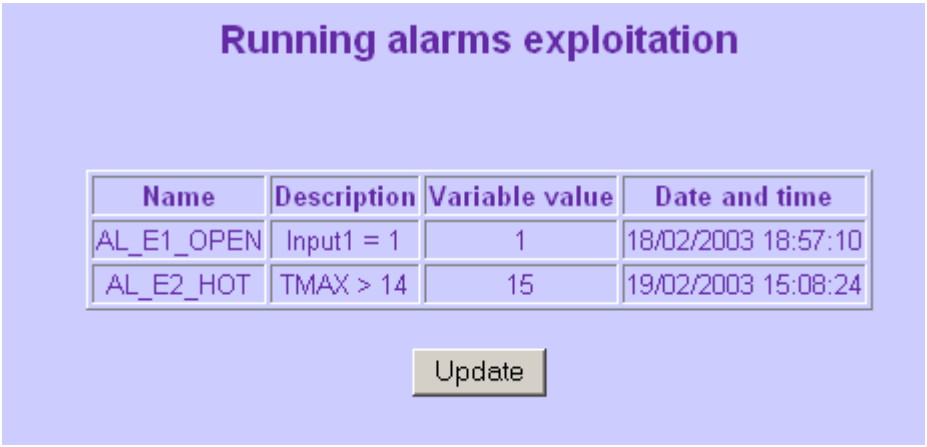
No writing is allowed in the processing menus.

An automatic refresh occurs every minute.

An ''**Update''** button updates the data in real time.

7.3 Running alarms

The ''**Running alarms exploitation''** page lists the alarms that are triggered and that are still in the triggered state.



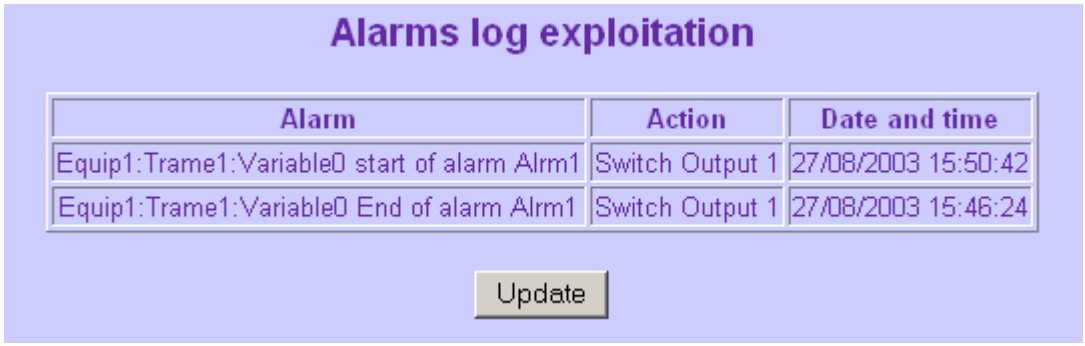
The date and time and trigger conditions are indicated for each alarm.

An automatic refresh occurs every minute.

An ''**Update''** button updates the data in real time.

7.4 Alarms Log

The alarms log traces the successive states of the alarms chronologically.



Two states are possible for each alarm: ''start'' and ''end''. These states are date/time stamped.

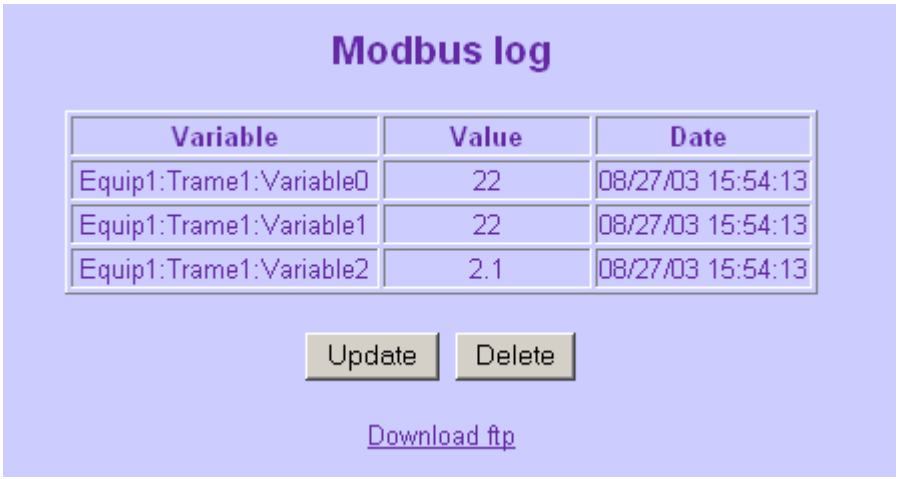
An automatic refresh occurs every minute.

An ''**Update''** button updates the data in real time.

7.5 Modbus Log

The Modbus log chronologically traces the successive states of the variables declared as traceable.

A variable is identified by its name and by the device and frame with which it is associated.



An automatic refresh occurs every minute.

An ''**Update''** button updates the data in real time.

The size of an HTML page is limited. Only the last 100 recordings are displayed. To obtain the entire file, it must be retrieved on the flash disk with the FTP protocol.

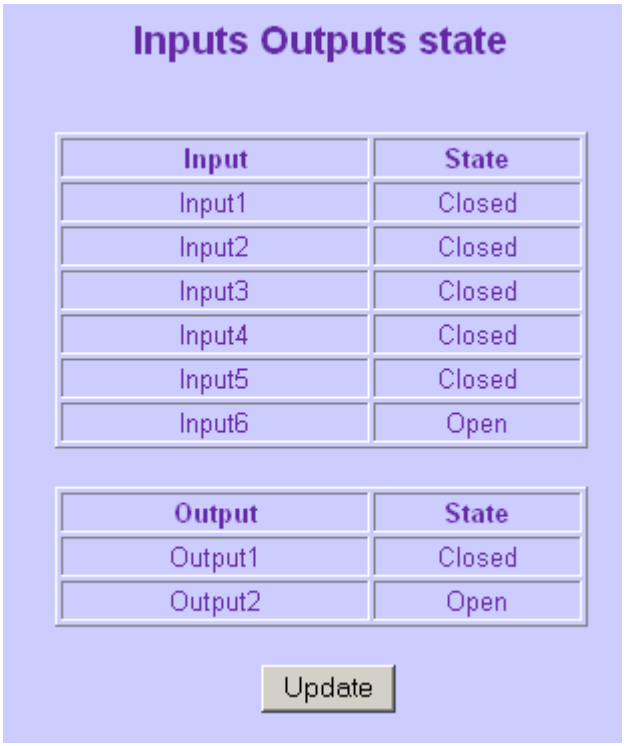
The "**Download ftp**" URL allows the user to download the log file diirectly from the Web page.

The ‘’**Delete’’** button only appears if the user is connected under the maintenance or administration level, but not under the operation level.

7.6 Inputs & Outputs

The gateways : WGM20(M) & WGM30(M) each have 6 digital inputs and 2 relay based outputs.

If a user is connected with utilization rights then he/she is able to visualize the current status of the inputs & outputs (open or closed).



7.7 Email log

Emails may be sent under three different conditions :

- IPCallBack
- Alarms
- Life sign

The email log lists the details of the last 100 emails iincluding their status (sent, pending or error).

8 User

This option is only accessible on the WGM30 and WGM30-M models. It allows the user to create his/her own html pages and to place them directly on the gateway in the special storage unit.

Information on this functionality and its use may be found in the Webdyngate Customization Guide.