Citect for Windows, Version 5.xx
Aquacom driver, User information

Contents

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1. User information

1.1 Device Application notes

	Detail
Manufacturer	ITT Flygt AB
	Box 2058
	SE-291 02 Kristianstad
	Sweden
Device name	Aquacom
Communications method	The connection between the central system and the RTUs is utilized with a modem used with fixed or dialed line. A plain RS-232 interface may also be used.

1.1.1 Principles of Operation

1.1.1.1 General

The RTUs contents a configuration file with the extension .atf or .ctg. This file has information about the RTU itself and therefore the driver has to handle this file. This is done in such a way that the driver with a special command "CreateIniFile" has to create an .ini file for each RTU. The .ini file is also used for saving setpoint changes, alarm priorities, trendinformation etc and therefore each RTU has there own .ini file.

1.1.1.2 Dialed up lines

With a dialed line, Citect calls the RTU when communication is needed. When the communication is finished via Cicode etc the driver sends out a BRK message to the RTU.
The RTU calls Citect only when an alarm has occurred.
Note. Not all types of RTUs support a full V.23 standard.

1.1.1.3 Fixed lines

With a fixed line the RTUs are connected in a multi drop net where the media and the time it takes to poll the RTUs for alarms is the limit over how many RTU's there can be on one fixed line. A usual limit for a V.23 fixed line is a number between 5-10, the number is depending on modem vendor, length and quality of the line.

1.1.2 Versions

1.1.2.1 Citect

For Dial in the Citectversion has to be at least 5.42 Special build HF542R020489. "This hotfix provides a fix for BUG# 20489 - Remote Dialin in devices cannot have a NULL char in CallerID string. This hotfix has not been regression tested. Application of this hotfix is done at your own risk." The ACT message has NULL characters before the startcharacter STX.

1.1.2.2 MacTec

The driver will work with RTU which uses the program language Forth in the operating system. This is the case for all units today. The "Basic" language is used in very old units and it can still be some of them at the market. If you are unsure what kind of unit you have please verify with ITT Flygt that your RTU is supporting Forth language.
For Dial in the RTU needs to have a patch delivered from ITT Flygt. The RTU have to send the ACT message after approximately 5 seconds after the modems has established a contact. Please verify with ITT Flygt that your RTU are supporting the Citect dial functionality.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.3 Reference: Communications forms

1.1.3.1 Fixed lines

1.1.3.1.1 Boards form

Field	Default	Allowable values
Board Name	This field is user defined.	
Board Type	COMX	
Address	0	
I/O Port	BLANK	
Interrupt	BLANK	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

1.1.3.1.2 Ports form

Field	Default	Allowable values
Port Name	This field is user defined.	
Port number		
Board name	Refers to the board previously defined in ‘boards’ form.	
Baud rate	9600	
Data bits	8	
Stop bits	1	
Parity	NONE	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

1.1.3.1.3 I/O Devices form

Field	Default	Allowable values
Name	This field is user defined, and is not used by the driver.	
Number	Must be unique.	
Address	x[:y]/filename	x = NodeID 0-50, y = stationnumber, 1-999 filename = name of configuration file (.ctg or .atf)
Protocol	AQUACOM	
Port name	Refers to the port previously defined in ‘ports’ form.	
Comment	This field is user defined and is not used by the driver.	

The aquacom consists of two separate addressingmethods. NodeId is used for fixed lines and stationnumber is used for dialed lines. Often these two addresses are the same but they don’t need to be so.
The filename is the name of the configurationfile which is delivered from ITT Flygt. This file shall be saved in \Citect\Bin or if you specify an own path in your Citect.ini file e.g. CfgPath=E:\Aquacom\Cfg.

You have two possibilities how to write the address:
1) 3/C60005se.atf
2) 3:7/C60005se.atf

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.3.2 Dialed lines

1.1.3.2.1 Boards form

Field	Default	Allowable values
Board Name	This field is user defined.	
Board Type	COMX	
Address	0	
I/O Port	BLANK	
Interrupt	BLANK	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

1.1.3.2.2 Ports form

Field	Default	Allowable values
Port Name	This field is user defined.	
Port number	-1	
Board name	Refers to the board previously defined in ‘boards’ form.	
Baud rate	9600	
Data bits	8	
Stop bits	1	
Parity	NONE	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

1.1.3.2.3 I/O Devices form

Field	Default	Allowable values
Name	This field is user defined, and is not used by the driver.	
Number	Must be unique.	
Address	x[:y]/filename	x = NodeID 0-50, y = stationnumber, 1-999 filename = name of configuration file (.ctg or .atf)
Protocol	AQUACOM	
Port name	Refers to the port previously defined in ‘ports’ form.	
Comment	This field is user defined and is not used by the driver.	
Scheduled	TRUE	
Time	BLANK	
Period	08:00:00	
Connect Action	BLANK	
Disconnect Action	BLANK	
Phone Number	Refers to the phone number Citect will call to this IO Device	
Caller ID	Must be unique.	

The aquacom consists of two separate addressing methods. NodeId is used for fixed lines and station number is used for dialled lines. Often these two addresses are the same but they don’t need to be so.
The filename is the name of the configuration file which is delivered from ITT Flygt. This file shall be saved in \Citect\Bin or if you specify an own path in your Citect.ini file e.g. CfgPath=E:\Aquacom\Cfg.

You have two possibilities how to write the address:
1) 3/C60005se.atf
2) 3:7/C60005se.atf

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.3.2.4 Caller ID

The Caller ID is build up from the special ‘ACT’ message the unit will send after the modems has got contact. If the unit has station number 9 the CallerID will be 9ACT369. The digits 369 are the checksum for this message. You can calculate the checksum through the way to add the decimal ASCII station number with the digits 312. You will find more information about this message in the chapter 2.5.1.6

Number	Decimal ASCII
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57

Example 1: Stationnumber 9
312 + 57 = 369
Caller ID 9ACT369

Example 2: Stationnumber 26
312 + 50 + 54 = 416
Caller ID 26ACT416

Example 3: Stationnumber 143
312 + 49 + 52 + 54 = 467
Caller ID 143ACT467

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.4 Reference: Data types

IO Device Type	Citect data format	Citect data types	Description/Special Usage/Limitations/ Valid Ranges
Status	St*x*	LONG	Default picture 1
Status with picture number	St*x/y*	LONG	
Channel	Ch*z*	STRING	Read and write channels
Channel	Ch*z*	INT	Read and write channels
Channel	Ch*z*	REAL	Read and write channels
Periodical report	Rp*t/u/v*	LONG	Read report data
Trend request	TrReq	STRING	Request trenddata
Pump control	Pu*p*	STRING	Default picture 1
Pump control with picture number	Pu*p/y*	STRING	
Alarm	Al*w*	DIGITAL	
Alarm code priorities	AlarmCode	STRING	
CreateIniFile	CreateIniFile	INT	Create ini file from atf/cfg file
ReadSetpoint	ReadSetpoint	INT	Read setpoints from RTU
WriteSetpoint	WriteSetpoint	INT	Write setpoints to RTU
CompareSetpoint	CompareSetpoint	STRING	Compare setpoints between cache and .ini file

Where:

x	Measuring point, 1 - 128
y	Picture number, 1 - 25
p	Machine number, 0 - 127
z	Channel number, 1 - 500
t	Report number, 1 -128
u	Segment number, 1 - 5
v	Day number, 0 – 3 0, today (normally not used) 1, yesterday 2, the day before yesterday 3, 2 days before yesterday
w	Alarm number, 1 - 9999

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.5 Examples data types

All data types are explained in more details in chapter 2.

1.1.5.1 Status (read only)

Data Type	LONG
Address	St9
Comment	The long value for wagon 9 at picture 1
Example	

Data Type	LONG
Address	St9/2
Comment	The long value for wagon 9 at picture 2
Example	

1.1.5.2 Channel (read and write)

Data Type	STRING
Address	Ch3
Comment	Telephone number 1 to Citect
Example	0567123456

Data Type	INT
Address	Ch22
Comment	Alarm position
Example	0=Local, 1=Remote

Data Type	REAL
Address	Ch29
Comment	Max level
Example	3,27

1.1.5.3 Report (read only)

Data Type	LONG
Address	Rp2/5/1
Comment	2=Report2, 5=Sum, 1=yesterday
Example	Long value

1.1.5.4 Trend (write only)

Data Type	STRING
Address	TrReq
Write	0303180910
Comment	Will start a reading of trend data from a given time
Example	9.10am at 18March 2003

1.1.5.5 Pump control (write only)

Data Type	STRING
Address	Pu2
Write	F
Comment	Pump2 picture 1
Example	Forward, start machine
Data Type	STRING
Address	Pu2/5
Write	F
Comment	Pump2 picture 5

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

Example Forward, start machine

1.1.5.6 Alarm (read only)

Data Type DIGITAL
Address Al8114
Comment Read the alarm 8114
Example

1.1.5.7 AlarmCode Priorities (write only)

Data Type STRING
Address AlarmCode
Write 8114B
Comment Change alarmcode priority for alarm 8114 to B
Example

1.1.5.8 CreateIniFile (write only)

Data Type INT
Address CreateIniFile
Comment
Example

1.1.5.9 ReadSetpoint (write only)

Data Type INT
Address ReadSetpoint
Comment
Example

1.1.5.10 WriteSetpoint (write only)

Data Type INT
Address WriteSetpoint
Comment
Example

1.1.5.11 CompareSetpoint (write)

This command will compare the cache and .ini and write information to a logfile saved in the LogPath. An internal statusflag will be activated and can be readout with a read of CompareSetpoint.

Data Type STRING
Address CompareSetpoint
Comment
Example

1.1.5.12 CompareSetpoint (read)

Data Type STRING
Address CompareSetpoint
Comment The status will be presented from the preceding made CompareSetpoint write.
Example OK, Difference, Look in logfile, No Commando

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.1.6 Hints, Tips, and Frequently asked questions

- For a low baudrate and many telegrams from one meter it can be shown some #COM at the screen. The reason for this #COM depends very often of that the IOServer get a request and it takes to long time to answer the question. Increase the Citect [LAN] timeout parameter to e.g. 20000 (20 sec); default is 8000 (8 sec).
- In some cases when you use longdistance modem you may need to handle the RTS signals separate with eg the Special Options for the COMx Driver -tPRE,POST When transmitting a message the driver will raise RTS for PRE milliseconds, transmit message, wait for POST milliseconds then drop RTS.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.2 Driver reference

	Detail
Driver name	AQUACOM
Maximum array size[1]	256

1.2.1 Driver generated error codes

N/A

1.2.2 Parameters, options, and settings

1.2.2.1 Standard Parameters used for fixed lines

Parameter	Default	Allowable values	Desription
Block (bytes)	256		
Delay (mS)	100		
MaxPending	2	2	
Polltime (mS)	0	0	
TimeOut (mS)*	5000	2000 to 32000	
Retry	3		
WatchTime (Sec)	30		

* The messages for Channel, trend and report can have a very long answer string and therefore they need extra long timeout. An algorithm TimeOut * 9600L / BaudRate is used for this timeout. This algorithm works only for the default timeout 5000mS. The TimeOut can never be bigger than 32000mS with the algorithm. The calculated value can be shown with the statistic parameter Long Time out period (ms).

1.2.2.2 Standard Parameters used for dialed lines

Parameter	Default	Values during tests	Desription
WatchTime (Sec)	30	1	

1.2.2.3 Dial Parameters

Parameter	Default	Values during tests	Desription
CacheRefresh	1	0	Prevent the automatic cache refresh.
ReadThroughCache	0	1	
WatchTime (Sec)	20	1	

[1] Equivalent to 'Maximum Request Length'

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.2.2.4 Driver Specific Parameters

Parameter	Default	Allowable values	Description
FilterDebug	0	0,1 or 2	Set to 0 will show the received string without rubbish characters before STX and after CR . Set to 1 will show both the filtered and raw string after each other. Set to 2 will only show the raw string.
MainPath	0	0 or 1	As default the CfgPath will be the master for Ini, Log and TrendPath. MainPath=1 will set the path Cfg, Log and TrendPath individually.
CfgPath	Citect\Bin		Path for were the cfg/atf files shall be stored.
IniPath	Citect\Bin		Path for were the .ini files shall be stored.
LogPath	Citect\Bin		Path for were the log files shall be stored.
TrendPath	Citect\Bin		Path for were the trenddata files shall be stored.
DisconnectTimeout	30000		If no more transmit after this time. The driver will make a disconnect of the modem.
RegisterDial (explained in the capital software protection)	MsgBox	MsgBox Fixed Dial	Extra message box will shown Only fixed registration Only dial registration
DisconnectTimeout	30000		
DisconnectDelay	3		

Examples:
MainPath=0 Default
CfgPath=C:\Aquacom\Cfg
IniPath= C:\Aquacom\Cfg
LogPath= C:\Aquacom\Cfg
TrendPath= C:\Aquacom\Cfg

MainPath=1
CfgPath=C:\Aquacom\Cfg
IniPath= C:\Citect\Bin or userdefined
LogPath= C:\ Citect\Bin or userdefined
TrendPath= C:\ Citect\Bin or userdefined

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.2.2.5 Driver Specific Parameters which can be applied to device/grouped

Parameter	Default	Allowable values	Description
StatusLiveTime	30000 mS		How often the driver shall make a new statusrequest.
ChannelLiveTime	30000 mS		How often the driver shall make a new channelrequest.
AlarmLiveTime	5000 mS		How often the driver shall make a new alarmrequest.
ReportLiveTime	30000 mS		How often the driver shall make a new reportrequest.

1.2.2.6 Driver Specific Parameters which can be applied only to channel (port)

Parameter	Default	Allowable values	Description
CitDial	0	0 or 1	Set to 0 will enable the fixed communication Set to 1 will enable special handling for Citect dial part

1.2.2.7 Aquacom Device/Group-specific Parameters

The AQUACOM driver supports the capability to apply different initialisation parameter values to specific I/O devices or groups of I/O devices. This means the user can specify:

a) Global parameters that apply to all devices
b) Channel (port) level parameters that apply to all devices on the specified port
c) Group level parameters that apply to all devices in a specified group
d) Device level parameters that apply only to the specified device

This feature can be implemented in the Citect.INI file for the following AQUACOM parameters:

StatusLiveTime
ChannelLiveTime
AlarmLiveTime
ReportLiveTime

To set parameter values for a particular group or device, you put a full stop (period) immediately after the name of the driver where it appears in the Citect.INI file, followed by the name of the particular port or group you want to specify a parameter setting for. For example:

[AQUACOM.<Port_Name>] applies the parameter settings to the specified port
[AQUACOM.<Group_Name>] applies the parameter settings to the specified group
[AQUACOM.<Port_Name>.<IODevice_Name>] applies to the specified device
Any parameters you then define in the following section of the Citect.INI file will relate only to the specified device or device group.

Example
The following Citect INI file format is an example of how the ' StatusLiveTime ' parameter could be specified differently for different I/O devices communicating using the AQUACOM Protocol.
Assume that two ports are used: PORT1 and PORT2.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

PORT1 has three I/O devices attached: DEV1A DEV1B DEV1C
PORT2 also has three devices:DEV2A DEV2B DEV2C

Assume that the user has specified that DEV1C and DEV2C belong to GROUPZ.
The Citect INI file contains the following entries:

```
[AQUACOM]
StatusLiveTime =10000
[AQUACOM.PORT1]
StatusLiveTime =20000
[AQUACOM.PORT2]
StatusLiveTime =20000
[AQUACOM.GROUPZ]
StatusLiveTime =30000
[AQUACOM.PORT1.DEV1A]
StatusLiveTime =10000
[AQUACOM.PORT2.DEV2B]
StatusLiveTime =40000
```

The resultant StatusLiveTime for the IO Devices will be as follows:

DEV1A:10000 as a result of [AQUACOM.PORT1.DEV1A]
DEV1B:20000 as a result of [AQUACOM.PORT1]
DEV1C:30000 as a result of [AQUACOM.GROUPZ]
DEV2A:20000 as a result of [AQUACOM.PORT2]
DEV2B:40000 as a result of [AQUACOM.PORT2.DEV2B]
DEV2C:30000 as a result of [AQUACOM.GROUPZ]

NOTE: As the above example shows, there is a hierarchy that determines the outcome of such settings. In simple terms, specific parameter settings overwrite general level settings. Therefore, parameters written in the scope of I/O devices will overwrite those set for groups; parameters set for groups will overwrite global settings, etc.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

1.2.3 Advanced

1.2.3.1 Driver generated statistics

Number	Label	Description
0	Frames Transmitted	Frames Transmitted
1	Frames Received	Normal frames received
2	Write Requests	Write Requests
3	Read Requests	Read Requests
4	Frames Accepted	Frames Accepted
5	Frames With Error	Frames With Error
6	Bad Check Sum	Bad Check Sum
7	Communication Error	Errors as parity error, break etc
8	Bad Message Length	A lot of rubbish in message
9	Bad Stx Position	How many STX before the correct STX
10	Bad Beginning	If not FF 00 00 before STX
11		
12		
13		
14		
15		
16		
17	Long Time out period (ms)	Long Time out used for Channel, Trend, Report
18	Dialed Registered	1 = Dial license registered
19	Fixed Registered	1 = Fixed license registered

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2. Analysis

2.1 Initialising the IO Device

2.1.1 Fixed line

When fixed line is chosen a standard WCA (Alarmrequest) is transmitted to control if the IODevice are there. If it should be an alarm saved in the unit the driver doesn't send any ALO (Acknowledge) why the alarm will be kept in the RTU until a normal alarmrequest will be sent out.

2.1.2 Dialed line

When dialed line is chosen via the driver specific parameter CitDial no WCA is transmitted as initrequest.

2.1.2.1 Dial out

As soon the modems have got contact the driver is waiting of the OK message from RTU. After this a normal alarm request has to be sent out before approximately 5 seconds after the driver has received the OK message. This time is the same as the driver specific parameter AlarmLiveTime.

2.1.2.2 Dial in

The CallerId is a standard ACT message. A typical CallerId can be eg 1ACT361.
As soon the CallerId has been accepted the driver will send an OK message and the RTU will answer with a normal alarm response. After the first alarm the driver will continue to send a normal WCA alarm request.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.2 Creation of the .ini file

An .ini file has to be created for each RTU. This .ini file is used for to know which channels/setpoints is used by the RTU etc.

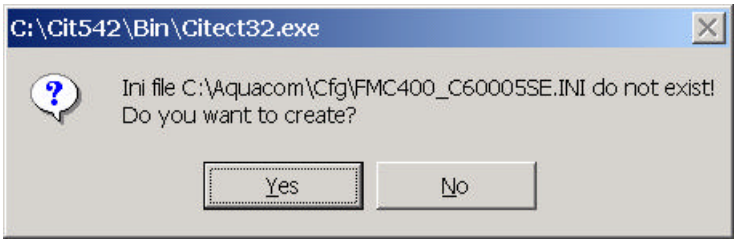
The creation of the .ini file is made from the RTUs configuration file. This file can have .atf or .cfg as an extension. The configuration file has to be stored in citect\bin or in the path which can be set via the driver specific parameter CfgPath in the group [AQUACOM] in citect.ini.
E.g. CfgPath=C:\Aquacom\Cfg

The name of the created .ini file will be IODevicename + _ + configurationfilename without extension + .ini

E.g. IODevicename FMC400 and configuration filename C60005se.atf gives the .ini filename FMC400_C60005SE.INI. This name guarantees it will be a unique .ini name even if several RTU uses the same configuration file.
The .ini file is saved in citect\bin or in the path which can be set via the driver specific parameter IniPath in the group [AQUACOM] in citect.ini.
E.g. IniPath=C:\Aquacom\Cfg

The .ini file can be created in two different ways.

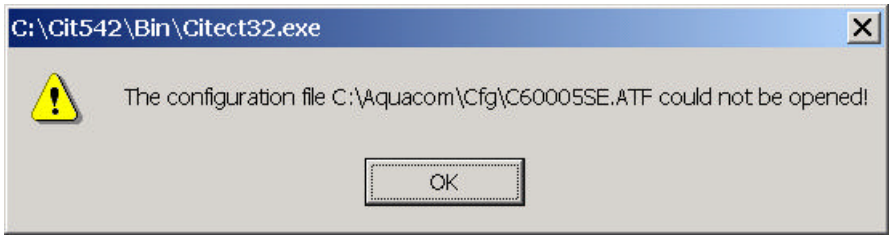
- The driver iteself will give you a message if it can't find any .ini file which don't maches the IODevicename and the filename in the address field in the IODeviceformular.



  In this case is only to press the Yes button and the file will be created.

- You can make a tag with address CreateIniFile and make a write command and a new .ini file will be created. A tag is always connected to the IODevice information and therefore the driver can recognize the information it needs for creating the file.

If the configuration file can't be found you will have the following message.



The .ini file contents the groups GENERAL, CHANNEL, TREND, ALARM, ALARMND and ALARMSTATUS which will be explained further in the following capitals.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.2.1 General

Some basic information is saved in this group.
[GENERAL]
CfgPath=C:\Aquacom\Cfg\
IniPath=C:\Aquacom\Cfg\
Filename=FMC400_C60005SE.INI
Name=MPC 6.00.05 SE
TypeNr=101
TypeName=Station built with Maestro 1.10+ (events)

2.2.2 Channel

The channel type is converted and saved direct in the .ini file in correct protocol format for to avoid further conversions.

CFG type	.ini type	Citect Datatype
C	T	STRING
D	D	INT
I	D	INT
S	D	INT
0	0	INT, REAL
1	1	REAL
2	2	REAL
3	3	REAL
4	4	REAL

2.2.3 Trend

See 2.5.8 TrendRequests

2.2.4 Alarm

This group is made for alarm priorities only. When an alarm priority has been changed via tagtype AlarmCode that alarms priority will be updated in the .ini file.

2.2.5 AlarmND

Some alarms have no down flank and therefore it will be no signal from the RTU who can acknowledge this type of alarms. The only way is to have the driver to set this alarm to passiv by itself after Citect has read out this alarm.

2.2.6 AlarmStatus

[ALARMSTATUS]
2=A

As soon an alarm has been activated the alarm will also be written to the alarm status group in the .ini file. As soon an alarm has been deactivated or passive the alarm will be deleted from the .ini file. Alarms with no down flank will be saved in the .ini file for a very short time. The first time after it has been read out by the Alarm Server it will be deleted direct by the driver itself. See also chapter 2.2.5.
In the above example alarm 2 (normally low level) is active. If Citect or the computer should be in shutdown state the .ini file will keep the status for the alarm. When Citect is started up again the driver will update the drivers alarm cache with the low-level alarm as active. Without this possibility the cache should be empty and the alarm should be deactivated in the alarm list even if it's still active in the unit.
If alarms has been read out by another software the unit will never send any new down flank to the driver and the alarm will never been deactivated. If this case should appear you can always delete this raw manually from the .ini file. The alarm will now be deactivated.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.3 The telegram construction

2.3.1 General

An AquaCom telegram is an ASCII-string with support of IBM extended codes. E.g. codes 01H to FFH is supported.

2.3.2 The syntax of a telegram

2.3.2.1 Telegram from Central System (CS)

Telegram												
			Message									
SOH		DEST		TYPE		DATA		#		CHK		CR

SOH Start of telegram 1 byte (01H)
DEST Destination 1 to 3 bytes. *Note that the ALO telegram differs.*
TYPE Type of message 3 bytes
DATA Message data 1 to 4000 bytes
End of message 1 byte (23H)
CHK Checksum 1 to 5 bytes
CR Carriage Return 1 byte (0DH)

Between every "cell" there is an ASCII space character (20H).

2.3.2.2 Telegram from Remote Terminal Unit (RTU)

Telegram								
	Answer							
STX		DATA		#		CHK		CR

STX Start of telegram 1 byte (02H)
DATA Answer data 1 to 4000 bytes
End of data 1 byte (23H)
CHK Checksum 1 to 5 bytes
CR Carriage Return 1 byte (0DH)

Between every "cell" there is an ASCII space character (20H).

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.4 Checksum calculation.

The checksum is calculated in following way:
Add together all characters after the STX or SOH to the '#' (Do not add this character), in a 16-bit signed integer.
Make the result from the absolute value of this sum.
Add a space (20H) to the string after the '#' and the characters from the checksum.

Example: The telegram SCO.

STX		SCO		#		293		CR

Detailed view

<STX>	***Space***	***S***	***C***	***O***	***Space***	**#**	**Space**	**2**	**9**	**3**	**Space**	**<CR>**
	32	*83*	*67*	*79*	*32*							

32+83+67+79+32=293

(The digits are in decimal)

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5 Data types

2.5.1 Alarm

2.5.1.1 General

The alarm consists of two types of alarms. The most common alarms can send Active or Passive status but the alarms with {ND} No Downflank can only send the status Active. This means the driver has to know when it is a {ND} alarm so the driver itself can acknowledge the alarm. Depending of this the group [ALARM_ND] is created in the .ini file.
The [ALARMSTATUS] group in the .ini file is created so the alarm can be saved if the computer or Citect should be shut down. This is the only way to know if an alarm has an Active Status when the computer or Citect is starting and the cache and memory is empty. All alarms will be written in the .ini file even system alarms which are not implemented in the original configuration file.

2.5.1.2 Alarm request (WCA)

<dest>	WCA	<set>	<YY>	<MM>	<DD>	<hh>	<mm>

dest	1-3 bytes,	Logical plant number, (1-999), if dialed or Node ID, (0-RTU dependant), if fixed
WCA	3 bytes,	Text
set	1 byte,	Set time and date (0=normal, 2=Set)
YY	2 bytes,	Year (00-99)
MM	1-2 bytes,	Month (1-12)
DD	1-2 bytes,	Day (1-31)
hh	1-2 bytes,	Hour (0-24)
mm	1-2 bytes,	Minutes (0-59)

NOTE! When the value for parameter **set** is 2, the RTU will adjust the real time clock in the RTU to the received parameters. (The driver uses set 0=normal)

2.5.1.3 Alarm answer - no alarms (ALN).

ALN	T	<v1>	<v2>	<v3>	<v4>

ALN	3 bytes,	text
T	1 byte,	text
v1	1-5 bytes,	AquaTrain wagon 1 (default zero)
v2	1-5 bytes,	AquaTrain wagon 2 (default zero)
v3	1-5 bytes,	AquaTrain wagon 3 (default zero)
v4	1-5 bytes,	AquaTrain wagon 4 (default zero)

This message is the default message when no alarm condition occurs.
The AquaTrain are functions only available with ITT Flygt MacTec software.

2.5.1.4 Alarm answer - acknowledge.

<dest>	ALO

dest	1-3 bytes,	0 if dialled or Node ID, (0-RTU dependant), if fixed
ALO	3 bytes,	text.

After a received ALR-message, the alarm must be acknowledged by transmitting an ALO-message. The alarm is then removed from the database in the RTU.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

If the RTU is placed on a fixed line, the destination will hold the node ID. In a dial up connection this destination must be set to zero, (0).
Some older versions of RTUs will go ON-Hook after a received ALO. Newer versions will be waiting for a new telegram (e.g. WCA).

2.5.1.5 Alarm answer - alarm is pending (ALR).

Note: Forth system version 1.?? or greater only

ALR	<lsn>	<code>	<YY>	<MM>	<DD>	<hh>	<mm>	<flank>	

	T	<V1>	<V2>	<V3>	<V4>	<pri>	<ss>

ALR	3 bytes,	text
lsn	1-5 bytes,	logical plant number (0-999)
code	1-5 bytes,	alarm code (0-8999)
YY	2 bytes,	year (00-99)
MM	1-2 bytes,	month (1-12)
DD	1-2 bytes,	day (1-31)
hh	1-2 bytes,	hour (0-23)
mm	1-2 bytes,	minute (0-59)
flank	1 bytes	the state of the event / alarm. Valid values are: A = Active / On / Up-flank P = Passive / Off / Down-flank
T	1 byte,	text
v1	1-5 bytes,	AquaTrain wagon 1 (default zero)
v2	1-5 bytes,	AquaTrain wagon 2 (default zero)
v3	1-5 bytes,	AquaTrain wagon 3 (default zero)
v4	1-5 bytes,	AquaTrain wagon 4 (default zero)
pri	1 byte,	Priority
ss	1-2 bytes,	second (0-59)

2.5.1.6 Dialled lined.

When the RTU calls, and CS has sent OK, the RTU transmits the alarm message immediately.

2.5.1.6.1 Caller identification

Citect require that the RTU can identify itself with a unique ID. This is used as CallerId in the IODevice formulary. The CallerId itself consist of the DEST, ACT and CHK part from the ACT message. Citect is filtering out the non ASCII characters. A typical CallerId is 9ACT369.
An ACT message can never be seen in Syslog.dat depending that the ACT message is arriving to Citect before the driver itself has been starting.

STX	DEST	ACT	#	CHK	CR

STX	Start of telegram	1 byte (02H)
DEST	Destination	1 to 3 bytes.
ACT	Type of message	3 bytes
#	End of message	1 byte (23H)
CHK	Checksum	1 to 5 bytes
CR	Carriage Return	1 byte (0DH)

Please observe that some RTU's does not have this caller identification feature. If this is the case, the central system must answer incoming calls with an OK-telegram to start the conversation.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.1.6.2 Caller identification - detailed example

Dialled RTU with plant number 9 dials in an alarm with code 12.

SOH	Start of telegram 1 byte (01H)	= <SOH> in this section
STX	Start of telegram. 1 byte (02H)	= <STX> in this section
CR	Carriage Return. 1 byte (0DH)	= <CR> in this section
Space	Space 1 byte (20H)	= _ in this section

RTU Dialing
RTU Connect
CS ← RTU **<STX>**_9_ACT_#_369_**<CR>**
CS → RTU **<SOH>**_OK_#_218_**<CR>**
CS ← RTU **<STX>**_ALR_9_12_3_6_17_13_58_P_T_0_0_0_0_A_27_#_1867_**<CR>**
CS → RTU **<SOH>**_0_ALO_#_364_**<CR>**
CS → RTU **<SOH>**_9_WCA_0_3_6_17_14_0_#_970_**<CR>**
CS ← RTU **<STX>**_ALN_T_0_0_0_0_#_719_**<CR>**
CS → RTU **<SOH>**_9_BRK_#_376_**<CR>**

2.5.1.7 Debug messages

Typical dial in alarm sequence:

```
Wed Sep 03 10:30:07 2003 114:29:46.332 Transmit Length 12
01 20 4F 4B 20 23 20 32 31 38 20 0D                  . OK # 218 .
Wed Sep 03 10:30:07 2003 114:29:46.651 Received Length 46
02 20 41 4C 52 20 31 20 32 20 33 20 39 20 33 20      . ALR 1 2 3 9 3
31 30 20 32 37 20 41 20 54 20 30 20 30 20 30 20      10 27 A T 0 0 0
30 20 42 20 35 20 23 20 31 36 38 37 20 0D            0 B 5 # 1687 .
Wed Sep 03 10:30:08 2003 114:29:47.171 Transmit ALO Length 15
01 20 30 20 41 4C 4F 20 23 20 33 36 34 20 0D         . 0 ALO # 364 .
Wed Sep 03 10:30:12 2003 114:29:51.333 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20      . 1 WCA 0 03 09
30 33 20 31 30 20 33 30 20 23 20 31 31 30 33 20      03 10 30 # 1103
0D                                                   .
Wed Sep 03 10:30:12 2003 114:29:51.729 Received Length 46
02 20 41 4C 52 20 31 20 32 20 33 20 39 20 33 20      . ALR 1 2 3 9 3
31 30 20 32 38 20 50 20 54 20 30 20 30 20 30 20      10 28 P T 0 0 0
30 20 42 20 32 20 23 20 31 37 30 30 20 0D            0 B 2 # 1700 .
Wed Sep 03 10:30:13 2003 114:29:52.231 Transmit ALO Length 15
01 20 30 20 41 4C 4F 20 23 20 33 36 34 20 0D         . 0 ALO # 364 .
Wed Sep 03 10:30:17 2003 114:29:56.833 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20      . 1 WCA 0 03 09
30 33 20 31 30 20 33 30 20 23 20 31 31 30 33 20      03 10 30 # 1103
0D                                                   .
Wed Sep 03 10:30:18 2003 114:29:57.166 Received Length 47
02 20 41 4C 52 20 31 20 31 20 33 20 39 20 33 20      . ALR 1 1 3 9 3
31 30 20 32 38 20 41 20 54 20 30 20 30 20 30 20      10 28 A T 0 0 0
30 20 42 20 31 37 20 23 20 31 37 33 38 20 0D         0 B 17 # 1738 .
Wed Sep 03 10:30:18 2003 114:29:57.668 Transmit ALO Length 15
01 20 30 20 41 4C 4F 20 23 20 33 36 34 20 0D         . 0 ALO # 364 .
Wed Sep 03 10:30:23 2003 114:30:02.333 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20      . 1 WCA 0 03 09
30 33 20 31 30 20 33 30 20 23 20 31 31 30 33 20      03 10 30 # 1103
0D                                                   .
Wed Sep 03 10:30:23 2003 114:30:02.682 Received Length 23
02 20 41 4C 4E 20 54 20 30 20 30 20 30 20 30 20      . ALN T 0 0 0 0
23 20 37 31 39 20 0D                                 # 719 .
Wed Sep 03 10:30:27 2003 114:30:06.541 Transmit BRK Length 15
01 20 31 20 42 52 4B 20 23 20 33 36 38 20 0D         . 1 BRK # 368 .
```

Typical dial out alarm sequence:

```
Wed Sep 10 10:28:30 2003 282:28:08.145 (OK detected)
Wed Sep 10 10:28:35 2003 282:28:12.837 Transmit Length 33
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

```
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20       . 1 WCA 0 03 09
31 30 20 31 30 20 32 38 20 23 20 31 31 30 38 20       10 10 28 # 1108
0D                                                    .
Wed Sep 10 10:28:35 2003 282:28:13.089 Received Length 23
02 20 41 4C 4E 20 54 20 30 20 30 20 30 20 30 20       . ALN T 0 0 0 0
23 20 37 31 39 20 0D                                  # 719 .
.
.
.
.
Wed Sep 10 11:53:00 2003 283:52:37.583 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20       . 1 WCA 0 03 09
31 30 20 31 31 20 35 33 20 23 20 31 31 30 37 20       10 11 53 # 1107
0D                                                    .
Wed Sep 10 11:53:00 2003 283:52:37.871 Received Length 48
02 20 41 4C 52 20 31 20 32 20 33 20 39 20 31 30       . ALR 1 2 3 9 10
20 31 31 20 35 31 20 50 20 54 20 30 20 30 20 30        11 51 P T 0 0 0
20 30 20 41 20 32 37 20 23 20 31 37 39 37 20 0D       0 A 27 # 1797 .
Wed Sep 10 11:53:01 2003 283:52:38.373 Transmit ALO Length 15
01 20 30 20 41 4C 4F 20 23 20 33 36 34 20 0D          . 0 ALO # 364 .
Wed Sep 10 11:53:05 2003 283:52:43.083 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20       . 1 WCA 0 03 09
31 30 20 31 31 20 35 33 20 23 20 31 31 30 37 20       10 11 53 # 1107
0D                                                    .
Wed Sep 10 11:53:06 2003 283:52:43.339 Received Length 23
02 20 41 4C 4E 20 54 20 30 20 30 20 30 20 30 20       . ALN T 0 0 0 0
23 20 37 31 39 20 0D                                  # 719 .
Wed Sep 10 11:53:11 2003 283:52:48.583 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20       . 1 WCA 0 03 09
31 30 20 31 31 20 35 33 20 23 20 31 31 30 37 20       10 11 53 # 1107
0D                                                    .
Wed Sep 10 11:53:11 2003 283:52:48.917 Received Length 48
02 20 41 4C 52 20 31 20 31 20 33 20 39 20 31 30       . ALR 1 1 3 9 10
20 31 31 20 35 31 20 41 20 54 20 30 20 30 20 30        11 51 A T 0 0 0
20 30 20 42 20 33 39 20 23 20 31 37 38 35 20 0D       0 B 39 # 1785 .
Wed Sep 10 11:53:12 2003 283:52:49.419 Transmit ALO Length 15
01 20 30 20 41 4C 4F 20 23 20 33 36 34 20 0D          . 0 ALO # 364 .
Wed Sep 10 11:53:16 2003 283:52:54.083 Transmit Length 33
01 20 31 20 57 43 41 20 30 20 30 33 20 30 39 20       . 1 WCA 0 03 09
31 30 20 31 31 20 35 33 20 23 20 31 31 30 37 20       10 11 53 # 1107
0D                                                    .
Wed Sep 10 11:53:17 2003 283:52:54.355 Received Length 23
02 20 41 4C 4E 20 54 20 30 20 30 20 30 20 30 20       . ALN T 0 0 0 0
23 20 37 31 39 20 0D                                  # 719 .
.
.
.
Wed Sep 10 12:01:57 2003 284:01:34.787 Transmit BRK Length 15
01 20 31 20 42 52 4B 20 23 20 33 36 38 20 0D          . 1 BRK # 368 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.2 Channels Write

2.5.2.1 General

When a setpoint is written to a channel, depending of the nature of RTU, all channels will be written in the same message. To avoid a flashing value at the Citect screen the value is first written to a driver cache and the original value are saved as a backup. If the writing has been successful and the RTU answer SCO the value will be written to the .ini file. If the writing shouldn't work properly and the RTU answer SCE the old value will be read back to the cache and a hardware error will appear.
Before a value is written to the RTU the format is compared with the .ini file. If the .ini file says this channel have a real with two decimals and you have made an input with three decimals the driver will adjust your value to two decimals with correct rounding before it will be sent to the RTU.

2.5.2.2 Function channels Write (MCC Forth)

dest>	MCC	<lsn>	<YY>	<MM>	<DD>	<hh>	<mm>	<ss>	

	<chan>	<type>	<value

----- Repeated (<= 200) ------

HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
prod	3 bytes,	Name of setpoint telegram.
lsn	1-5 bytes,	logical plant number (0-999).
YY	2 bytes,	year (00-99).
MM	2 bytes,	month (01-12).
DD	2 bytes,	day (01-31).
hh	2 bytes,	hour (00-23).
mm	2 bytes,	minute (00-59).
ss	2 bytes,	second (00-59).

DATA

chan	1-3 bytes,	channel number, (1-255).
type	1 byte,	type of data that follow. A = Analogue (0-200). D = Integer (0-32767). T = Text (Data between character ' (27H)). 0-4= Float (Data transmitted with 0-4 decimals).
value	1-128 bytes,	data as described above.

2.5.2.3 Function channels answer

SCO	#	293

Comment: The result could be a SCE telegram as well if the RTU does not accept the contents in the send telegram.
A SCE answer for a MCC telegram might contain the channel number the RTU rejected.
Exampel: Channel 48 didn't work

SCE	48	#	423

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.2.4 Debug messages

Example writing to high level limit. Channel 31, 2 decimals, value 2.27. For one value all channels will be written depending of the nature of the protocol.

```
Wed Sep 10 16:05:09 2003 288:04:46.410 Transmit Length 819
01 20 31 20 4D 43 43 20 31 20 30 33 20 30 39 20      . 1 MCC 1 03 09
31 30 20 31 36 20 30 35 20 30 39 20 33 20 54 20      10 16 05 09 3 T
27 34 27 20 34 20 54 20 27 5F 27 20 37 20 44 20      '4' 4 T '_' 7 D
30 20 38 20 54 20 27 5F 27 20 39 20 54 20 27 5F      0 8 T '_' 9 T '_
27 20 31 30 20 54 20 27 5F 27 20 31 31 20 54 20      ' 10 T '_' 11 T
27 5F 27 20 31 32 20 44 20 30 20 31 38 20 44 20      '_' 12 D 0 18 D
30 20 32 31 20 44 20 31 33 20 32 32 20 44 20 31      0 21 D 13 22 D 1
20 32 33 20 44 20 31 30 20 32 34 20 44 20 31 30       23 D 10 24 D 10
20 32 35 20 44 20 33 30 20 32 36 20 44 20 30 20       25 D 30 26 D 0
32 37 20 44 20 30 20 32 39 20 32 20 33 2E 30 30      27 D 0 29 2 3.00
20 33 30 20 32 20 30 2E 30 30 20 33 31 20 32 20       30 2 0.00 31 2
32 2E 32 37 20 33 32 20 32 20 31 2E 30 30 20 33      2.27 32 2 1.00 3
33 20 32 20 30 2E 33 30 20 33 34 20 44 20 30 20      3 2 0.30 34 D 0
33 36 20 31 20 30 2E 30 20 33 37 20 31 20 30 2E      36 1 0.0 37 1 0.
30 20 33 38 20 31 20 30 2E 30 20 33 39 20 31 20      0 38 1 0.0 39 1
30 2E 30 20 34 31 20 31 20 30 2E 30 20 34 32 20      0.0 41 1 0.0 42
31 20 30 2E 30 20 34 33 20 31 20 30 2E 30 20 34      1 0.0 43 1 0.0 4
34 20 31 20 30 2E 30 20 34 36 20 33 20 31 2E 30      4 1 0.0 46 3 1.0
30 30 20 34 38 20 31 20 31 2E 30 20 34 39 20 31      00 48 1 1.0 49 1
20 30 2E 30 20 35 31 20 30 20 32 20 35 32 20 44       0.0 51 0 2 52 D
20 30 20 35 33 20 32 20 30 2E 30 30 20 35 34 20       0 53 2 0.00 54
32 20 32 33 2E 30 30 20 35 35 20 32 20 30 2E 30      2 23.00 55 2 0.0
30 20 35 36 20 32 20 30 2E 30 30 20 35 37 20 32      0 56 2 0.00 57 2
20 30 2E 30 30 20 35 38 20 44 20 31 30 20 35 39       0.00 58 D 10 59
20 44 20 35 20 36 30 20 44 20 30 20 36 31 20 30       D 5 60 D 0 61 0
20 30 20 36 32 20 44 20 30 20 36 34 20 32 20 30       0 62 D 0 64 2 0
2E 30 30 20 36 35 20 32 20 30 2E 30 30 20 36 36      .00 65 2 0.00 66
20 30 20 35 20 36 37 20 32 20 31 2E 30 30 20 36       0 5 67 2 1.00 6
38 20 32 20 30 2E 30 30 20 36 39 20 32 20 30 2E      8 2 0.00 69 2 0.
30 30 20 37 30 20 32 20 30 2E 30 30 20 37 31 20      00 70 2 0.00 71
32 20 30 2E 30 30 20 37 32 20 32 20 30 2E 30 30      2 0.00 72 2 0.00
20 37 33 20 32 20 30 2E 30 30 20 37 34 20 32 20       73 2 0.00 74 2
30 2E 30 30 20 37 35 20 32 20 30 2E 30 30 20 37      0.00 75 2 0.00 7
36 20 32 20 30 2E 30 30 20 37 37 20 32 20 30 2E      6 2 0.00 77 2 0.
30 30 20 37 39 20 31 20 30 2E 30 20 38 30 20 31      00 79 1 0.0 80 1
20 30 2E 30 20 38 32 20 31 20 30 2E 30 20 38 33       0.0 82 1 0.0 83
20 31 20 30 2E 30 20 39 32 20 32 20 30 2E 30 30       1 0.0 92 2 0.00
20 39 33 20 31 20 30 2E 30 20 39 34 20 31 20 30       93 1 0.0 94 1 0
2E 30 20 39 35 20 31 20 30 2E 30 20 39 36 20 31      .0 95 1 0.0 96 1
20 30 2E 30 20 39 37 20 31 20 30 2E 30 20 39 38       0.0 97 1 0.0 98
20 31 20 30 2E 30 20 39 39 20 31 20 30 2E 30 20       1 0.0 99 1 0.0
31 30 30 20 31 20 30 2E 30 20 31 30 31 20 31 20      100 1 0.0 101 1
30 2E 30 20 31 30 32 20 31 20 30 2E 30 20 31 30      0.0 102 1 0.0 10
33 20 31 20 30 2E 30 20 31 30 34 20 31 20 30 2E      3 1 0.0 104 1 0.
30 20 31 30 35 20 31 20 30 2E 30 20 31 30 36 20      0 105 1 0.0 106
31 20 30 2E 30 20 31 30 37 20 31 20 30 2E 30 20      1 0.0 107 1 0.0
31 30 38 20 31 20 30 2E 30 20 31 30 39 20 31 20      108 1 0.0 109 1
30 2E 30 20 31 31 30 20 31 20 30 2E 30 20 31 31      0.0 110 1 0.0 11
31 20 31 20 30 2E 30 20 31 31 32 20 31 20 30 2E      1 1 0.0 112 1 0.
30 20 31 31 39 20 30 20 33 20 31 32 31 20 30 20      0 119 0 3 121 0
33 20 31 32 32 20 30 20 30 20 23 20 32 39 35 30      3 122 0 0 # 2950
36 20 0D                                             6 .
Wed Sep 10 16:05:10 2003 288:04:48.293 Received Length 13
02 20 53 43 4F 20 23 20 32 39 33 20 0D               . SCO # 293 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.3 Channels Read

2.5.3.1 General

Before you can start to read you have to first make a write. See the comments below. The driver doesn't do this automatically depending of that the operator has to decide the values/setpoints in the .ini is correct before the writing is made. The writing can be done with the command 'WriteSetpoint' or you can choose to write a value as described in the previous chapter.

2.5.3.2 Function channels Read (MCC Forth)

<dest>	MCC	<lsn>	<YY>

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
prod	3 bytes,	Name of setpoint telegram.
lsn	1-5 bytes,	must be 0, (zero).
YY	2 bytes,	must be 1, (one).

2.5.3.3 Function channels answer

<chan>	<type>	<value

----- Repeated (<= 200) ------

chan	1-3 bytes,	channel number, (1-255).
type	1 byte,	type of data that follow. A = Analogue (0-200). D = Integer (0-32767). T = Text (Data between character ' (27H)). 0-4= Float (Data transmitted with 0-4 decimals).
value	1-128 bytes,	data as described above.

Comment: In order to fetch a MCC telegram from a Forth RTU you must first send one. This means that it is not possible to fetch settings from a cold started RTU.
Important 1: Please note that the RTU only sends the data part of the telegram.
Important 2: Up till now, (2002-08-30), it is only possible to fetch the last sent setpoint telegram from the RTU. This means that if we have 100 channels to send, but we only send the first 50 channels it is NOT possible to fetch the second half of the setpoint telegram.
This problem will hopefully be solved in the future, but today you must send what you would like to receive.

2.5.3.4 Debug messages

```
Wed Sep 10 16:00:52 2003 288:00:30.053 Transmit Length 19
01 20 31 20 4D 43 43 20 30 20 31 20 23 20 35 31      . 1 MCC 0 1 # 51
37 20 0D                                             7 .
Wed Sep 10 16:00:54 2003 288:00:31.480 Received Length 788
02 20 33 20 54 20 27 34 27 20 34 20 54 20 27 27      . 3 T '4' 4 T ''
20 37 20 44 20 30 20 38 20 54 20 27 27 20 39 20       7 D 0 8 T '' 9
54 20 27 27 20 31 30 20 54 20 27 27 20 31 31 20      T '' 10 T '' 11
54 20 27 27 20 31 32 20 44 20 30 20 31 38 20 44      T '' 12 D 0 18 D
20 30 20 32 31 20 44 20 31 33 20 32 32 20 44 20       0 21 D 13 22 D
31 20 32 33 20 44 20 31 30 20 32 34 20 44 20 31      1 23 D 10 24 D 1
30 20 32 35 20 44 20 33 30 20 32 36 20 44 20 30      0 25 D 30 26 D 0
20 32 37 20 44 20 30 20 32 39 20 32 20 33 2E 30       27 D 0 29 2 3.0
30 20 33 30 20 32 20 30 2E 30 30 20 33 31 20 32      0 30 2 0.00 31 2
20 32 2E 32 34 20 33 32 20 32 20 31 2E 30 30 20       2.24 32 2 1.00
33 33 20 32 20 30 2E 33 30 20 33 34 20 44 20 30      33 2 0.30 34 D 0
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

```
20 33 36 20 31 20 30 2E 30 20 33 37 20 31 20 30      36 1 0.0 37 1 0
2E 30 20 33 38 20 31 20 30 2E 30 20 33 39 20 31      .0 38 1 0.0 39 1
20 30 2E 30 20 34 31 20 31 20 30 2E 30 20 34 32       0.0 41 1 0.0 42
20 31 20 30 2E 30 20 34 33 20 31 20 30 2E 30 20       1 0.0 43 1 0.0
34 34 20 31 20 30 2E 30 20 34 36 20 33 20 31 2E      44 1 0.0 46 3 1.
30 30 30 20 34 38 20 31 20 31 2E 30 20 34 39 20      000 48 1 1.0 49
31 20 30 2E 30 20 35 31 20 30 20 32 20 35 32 20      1 0.0 51 0 2 52
44 20 30 20 35 33 20 32 20 30 2E 30 30 20 35 34      D 0 53 2 0.00 54
20 32 20 32 33 2E 30 30 20 35 35 20 32 20 30 2E       2 23.00 55 2 0.
30 30 20 35 36 20 32 20 30 2E 30 30 20 35 37 20      00 56 2 0.00 57
32 20 30 2E 30 30 20 35 38 20 44 20 31 30 20 35      2 0.00 58 D 10 5
39 20 44 20 35 20 36 30 20 44 20 30 20 36 31 20      9 D 5 60 D 0 61
30 20 30 20 36 32 20 44 20 30 20 36 34 20 32 20      0 0 62 D 0 64 2
30 2E 30 30 20 36 35 20 32 20 30 2E 30 30 20 36      0.00 65 2 0.00 6
36 20 30 20 35 20 36 37 20 32 20 31 2E 30 30 20      6 0 5 67 2 1.00
36 38 20 32 20 30 2E 30 30 20 36 39 20 32 20 30      68 2 0.00 69 2 0
2E 30 30 20 37 30 20 32 20 30 2E 30 30 20 37 31      .00 70 2 0.00 71
20 32 20 30 2E 30 30 20 37 32 20 32 20 30 2E 30       2 0.00 72 2 0.0
30 20 37 33 20 32 20 30 2E 30 30 20 37 34 20 32      0 73 2 0.00 74 2
20 30 2E 30 30 20 37 35 20 32 20 30 2E 30 30 20       0.00 75 2 0.00
37 36 20 32 20 30 2E 30 30 20 37 37 20 32 20 30      76 2 0.00 77 2 0
2E 30 30 20 37 39 20 31 20 30 2E 30 20 38 30 20      .00 79 1 0.0 80
31 20 30 2E 30 20 38 32 20 31 20 30 2E 30 20 38       1 0.0 82 1 0.0 8
33 20 31 20 30 2E 30 20 39 32 20 32 20 30 2E 30      3 1 0.0 92 2 0.0
30 20 39 33 20 31 20 30 2E 30 20 39 34 20 31 20      0 93 1 0.0 94 1
30 2E 30 20 39 35 20 31 20 30 2E 30 20 39 36 20       0.0 95 1 0.0 96
31 20 30 2E 30 20 39 37 20 31 20 30 2E 30 20 39       1 0.0 97 1 0.0 9
38 20 31 20 30 2E 30 20 39 39 20 31 20 30 2E 30      8 1 0.0 99 1 0.0
20 31 30 30 20 31 20 30 2E 30 20 31 30 31 20 31       100 1 0.0 101 1
20 30 2E 30 20 31 30 32 20 31 20 30 2E 30 20 31       0.0 102 1 0.0 1
30 33 20 31 20 30 2E 30 20 31 30 34 20 31 20 30      03 1 0.0 104 1 0
2E 30 20 31 30 35 20 31 20 30 2E 30 20 31 30 36      .0 105 1 0.0 106
20 31 20 30 2E 30 20 31 30 37 20 31 20 30 2E 30       1 0.0 107 1 0.0
20 31 30 38 20 31 20 30 2E 30 20 31 30 39 20 31       108 1 0.0 109 1
20 30 2E 30 20 31 31 30 20 31 20 30 2E 30 20 31       0.0 110 1 0.0 1
31 31 20 31 20 30 2E 30 20 31 31 32 20 31 20 30      11 1 0.0 112 1 0
2E 30 20 31 31 39 20 30 20 33 20 31 32 31 20 30      .0 119 0 3 121 0
20 33 20 31 32 32 20 30 20 30 20 23 20 33 31 31       3 122 0 0 # 311
39 31 20 0D                                          91 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.4 Alarm code priorities

2.5.4.1 General

When an alarm code priority is written the value is first saved in the driver. If the writing has been successful and the RTU has answered SCO the value will be written to the .ini file. If the writing shouldn't work properly and the RTU answer SCE the new value will not be saved in the .ini file instead a hardware error will appear.

2.5.4.2 Alarmcode priorities (SLT)

<dest>	SLT	<code>	<prio>

dest	1-3 bytes,	Logical plant number, (1-999), if dialed or Node ID, (0-RTU dependant), if fixed
SLT	3 bytes,	text.
code	1-4 bytes,	alarm code (1-9999).
prio	1 byte,	the alarm priority as A-Z. Ordinary priorities are: A = Alarms that will be sent to pager. B = Normal alarms. C = Alarms that you do not care about. D = Time controlled A-alarms H = Events are treated as an alarm, The RTU must be configured to handle events before this feature will work.

2.5.4.3 Alarmcode priorities answer

If the writing has succeed the rtu will answer with a SCO message.

SCO	#	293

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram. This result will create a Citect driver error "DRIVER_CANNOT_WRITE".

SCE	#	283

2.5.4.4 Debug messages

Changing of alarm 2 (normally low level) to priority A has succeeded and the .ini file has been updated.

```
Wed Sep 10 10:55:52 2003 282:55:29.479 Transmit Length 19
01 20 31 20 53 4C 54 20 32 20 41 20 23 20 35 36          . 1 SLT 2 A # 56
37 20 0D                                                 7 .
Wed Sep 10 10:55:52 2003 282:55:29.746 Received Length 13
02 20 53 43 4F 20 23 20 32 39 33 20 0D          .    SCO # 293 .
```

Changing of alarm 0 to priority A has not been accepted.

```
Wed Sep 10 11:00:59 2003 283:00:36.327 Transmit Length 19
01 20 31 20 53 4C 54 20 30 20 41 20 23 20 35 36          . 1 SLT 0 A # 56
35 20 0D                                                 5 .
Wed Sep 10 11:00:59 2003 283:00:36.636 Received Length 13
02 20 53 43 45 20 23 20 32 38 33 20 0D          .    SCE # 283 .
Wed Sep 10 11:00:59 2003 283:00:36.654 Error: General error
  WRITE    0008 PORT1_BOARD1      FMC400              AlarmCode0    1
  Generic 000008 Driver 00000036 (0x00000024)
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.5 Status

2.5.5.1 General

The status datatype are using the Aquacom message type “Status request (WGS)”. All tag information are coming in one message and the driver is saving this tags in an internal cache. When the driver makes a read request it’s first looking if the value is too old via the driver specific timer parameter StatusLiveTime. If the time has passed the driver makes a new request and saves this new value in the cache and sends the value to Citects dcb buffer in the same time. If the timer hasn’t passed the value will only be read out from the cache.
If the project uses several pictures the cache contents the information for all used pictures. Each page has a separate timer for the StatusLiveTime parameter.
For the datatype status the command !=Ignore is always used in the request.

2.5.5.2 Status request (WGS)

<dest>	WGS	P	<mash>	<dir>	<pict>

Dest	1-3 bytes,	Logical plant number, (1-999), if dialed or Node ID, (0-RTU dependant), if fixed
WGS	3 bytes,	text.
P	1 bytes,	text.
mash	1-3 bytes,	machine number to make a remote control of.
dir	1 byte,	command. F = Forward, start machine. B = Backward, reverse the machine. N = Zero, stop the machine. M = Motor protection ! = Ignore. RTU will ignore the command.
pict	1-3 bytes,	picture number from CS. If not used send 1, (one).

2.5.5.3 Status answer

<ver>	<prod>	<Nal>	<alp>	<rem>	<r1>	<r2>	<r3>	<r4>	

	<mp1>	<mp2>	……	<mp128>

ver	3-5 bytes,	Forth system version, some older basic RTUs send their version number * 100.
prod	1-3 bytes,	type of RTU or compile ID, sometimes not used.
Nal	1-3 bytes,	number of alarms following last measuring point.
alp	1-3 bytes,	position where the first alarm is to be found.
rem	1 byte,	indicates if RTU is controlled locally = 0 or is in remote control = 1.
r1 - r4	1-5 bytes,	reserved for future use.
mp1-mp128	1-12 bytes,	data from the measuring points 1 to 128 for requested picture.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.5.4 Debug messages

Example with picture 1:

```
Wed Sep 10 17:09:01 2003 289:08:38.505 Transmit Length 23
01 20 31 20 57 47 53 20 50 20 30 20 21 20 31 20      . 1 WGS P 0 ! 1
23 20 37 32 34 20 0D                                 # 724 .
Wed Sep 10 17:09:02 2003 289:08:39.246 Received Length 145
02 20 33 32 30 30 30 20 31 35 35 31 32 20 30 31      . 32000 15512 01
20 34 38 20 30 20 30 20 30 20 30 20 30 20 30 20       48 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 32 32 36 37 20 30 20 30 20 30 20 32 33 30      0 2267 0 0 0 230
30 30 20 30 20 30 20 30 20 30 20 30 20 30 20 34      00 0 0 0 0 0 0 4
39 35 32 33 20 31 20 31 20 30 20 30 20 30 20 30      9523 1 1 0 0 0 0
20 30 20 30 20 30 20 32 20 30 20 30 20 30 20 30       0 0 0 2 0 0 0 0
20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30       0 0 0 0 0 0 0 0
20 30 20 30 20 30 20 31 20 23 20 35 36 38 32 20       0 0 0 1 # 5682
0D                                                   .
```

Example with picture 2 (No picture 2 is configured in the rtu):

```
Wed Sep 10 17:09:02 2003 289:08:39.746 Transmit Length 23
01 20 31 20 57 47 53 20 50 20 30 20 21 20 32 20      . 1 WGS P 0 ! 2
23 20 37 32 35 20 0D                                 # 725 .
Wed Sep 10 17:09:03 2003 289:08:40.183 Received Length 39
02 20 33 32 30 30 30 20 31 35 35 31 32 20 30 31      . 32000 15512 01
20 31 20 30 20 30 20 30 20 30 20 30 20 31 20 23       1 0 0 0 0 0 1 #
20 31 32 38 36 20 0D                                  1286 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.6 Pump Control

2.5.6.1 General

A pump control telegram is exactly the same as status request (WGS). The only difference in the information flow is that the command is F, B, N or M. In a standard status request it's an ignore command '!' instead. In pump control you can also use address 0 for a pump and this is the same as resetting.
The status cache is also updated when the answer from a pump control request has received.

2.5.6.2 Debug messages

```
Tue Sep 16 11:58:04 2003 427:57:40.783 Transmit Length 23
01 20 31 20 57 47 53 20 50 20 32 20 4E 20 31 20      . 1 WGS P 2 N 1
23 20 37 37 31 20 0D                                 # 771 .
Tue Sep 16 11:58:05 2003 427:57:41.576 Received Length 148
02 20 33 32 30 30 30 20 31 35 35 31 32 20 30 31      . 32000 15512 01
20 34 38 20 31 20 30 20 30 20 30 20 30 20 30 20       48 1 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 31 38 38 39 20 30 20 30 20 30 20 32 33 30      0 1889 0 0 0 230
30 30 20 30 20 30 20 30 20 30 20 30 20 30 20 34      00 0 0 0 0 0 0 4
31 32 36 35 20 31 20 31 20 30 20 30 20 30 20 30      1265 1 1 0 0 0 0
20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30       0 0 0 0 0 0 0 0
20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30       0 0 0 0 0 0 0 0
20 30 20 30 20 30 20 38 30 35 30 20 23 20 35 38       0 0 0 8050 # 58
34 31 20 0D                                          41 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.7 Periodical report

2.5.7.1 General

The report datatype are using the Aquacom message type “Periodical report (WGD)”. All tag information are coming in one message and therefore has the driver to save this tags in an internal cache. When the driver makes a report read request it’s first looking if the value is too old via the driver specific timerparameter ReportLiveTime. If the time has passed the driver makes a new request and saves this new value in the cache and sends the value to Citects dcb buffer in the same time. If the timer hasn’t passed the value will only be read out from the cache.
If the project uses several reports eg from yesterday, the day before yesterday and two days before yesterday the cache contents the information for all used reports. Each report has a separate timer for the ReportLiveTime parameter.

2.5.7.2 Periodical report (WGD)

<dest>	WGD	<YY>	<MM>	<DD>	<hh>	<mm>	<aa>	<bb>	<cc>

dest	1-3 bytes,	Logical plant number, (1-999), if dialed or Node ID, (0-RTU dependant), if fixed
WGD	3 bytes,	text.
YY	2 bytes,	year to set the real time clock.
MM	2 bytes,	month to set the real time clock.
DD	2 bytes,	day to set the real time clock.
hh	2 bytes,	hour to set the real time clock.
mm	2 bytes,	minute to set the real time clock.
aa	2 bytes,	requested year for report.
bb	2 bytes,	requested month for report.
cc	2 bytes,	requested day for report.

The answer could be "periodical data" or NDA, (No Data Available).

2.5.7.3 Periodical data

<seg1>	<seg2>	<seg3>	<seg4>	<sum>

---------- Repeated for all measuring points (1-128) ----------
seg1 1-12 bytes, data from segment1 (default 00-06).
seg2 1-12 bytes, data from segment2 (default 06-09).
seg3 1-12 bytes, data from segment3 (default 09-16).
seg4 1-12 bytes, data from segment4 (default 16-00).
sum 1-12 bytes, total value from seg1 to seg4.

2.5.7.4 Debug messages

```
Tue Sep 16 12:06:20 2003 428:05:56.681 Transmit Length 40
01 20 31 20 57 47 44 20 30 33 20 30 39 20 31 36      . 1 WGD 03 09 16
20 31 32 20 30 36 20 30 33 20 30 39 20 31 34 20       12 06 03 09 14
23 20 31 34 34 30 20 0D                               # 1440 .
Tue Sep 16 12:06:21 2003 428:05:57.185 Received Length 110
02 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      . 0 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20      0 0 0 0 0 0 0 0
30 20 30 20 30 20 23 20 34 30 33 32 20 0D            0 0 0 # 4032 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.8 Trend request

2.5.8.1 General

The historical trend data is saved in the file Aquatrend.log. The size of the file is depending of the answer from the RTU. If you make a request there you want data from e.g. 0306271040 the file will start from 0306271045. The fileformat is explained below. The fileformat is optimized for merging the trenddata into the Citect trendsystem via cicode. See the codeexample in appendix 3.

2.5.8.2 Historical trend request (WGH)

<dest>	WGH	<YY>	<MM>	<DD>	<hh>	<mm>

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
WGH	3 bytes,	text.
YY	2 bytes,	year of last stored block.
MM	2 bytes,	month of last stored block.
DD	2 bytes,	day of last stored block.
hh	2 bytes,	hour of last stored block.
mm	2 bytes,	minute of last stored block.

The answer could be "historical trend data" or "No Data Available" (NDA).

2.5.8.3 Historical trend data answer

<YY>	<MM>	<DD>	<hh>	<mm>	<v1>	<v2>	<v3>	.....	<v32>	!

---------- repeated for every 5min segment ----------

YY	2 bytes,	year for the block.
MM	1-2 bytes,	month for the block.
DD	1-2 bytes,	day for the block.
hh	1-2 bytes,	hour for the block.
mm	1-2 bytes,	minute for the block.
v1-v32	1-5 bytes,	data from measuring point 1 to 32. The actual numbers is depending of the type of RTU and version.
!	1 bytes,	text. Separates the blocks.

The length of the telegram must be calculated as follow.
The time must not exceed 6 hours and the whole telegram must be smaller than 4000 characters.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.8.4 Trendinformation from configuration file

```
6
;------------------------------------------------------------------------
; Point Text_number Wagon Min Max Max_input_value
; Text number < 1000 located in RAPPTEXT.CFG
; Text number >=1000 located in ANL????.RFG
;------------------------------------------------------------------------

1   1011   1       0     0.01    10 0 ; {DC} TrW1Lev
2   1012   2       0      0.1     6 1 ; {DC} TrP1CapCalc
3   1013   3       0      0.1     6 1 ; {DC} TrP2CapCalc
4   1014   4       0      0.1     6 1 ; {DC} TrInFl
5   1015   5       0      0.1     6 1 ; {DC} TrW1OFlow
6   1016   6       0      0.1     1 1 ; {DC} TrRain
```

2.5.8.5 Trendinformation translated to .ini file

```
[TREND]
NrOfTrends=6
1=TrW1Lev
2=TrP1CapCalc
3=TrP2CapCalc
4=TrInFl
5=TrW1OFlow
6=TrRain
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.8.6 Trend logfile

The trendtag is builded up from the trendname in the configurationfile and the IODevicename. It's possible to use the same configurationfile for several rtu units in a project and the only way to separate the trendnames is to include the IODevice name.

2.5.8.6.1 Normal trendblock

(Start)
8 ***(Number of values for each trendtag)***
0306271045 ***(Starttime for trends)***
(Start of trendblock)
FMC400_TrW1Lev ***(First trendtag)***
1887 ***(Time 1045)***
1887 ***(Time 1050)***
1887 ***(Time 1055)***
1887 ***(Time 1100)***
1888 ***(Time 1105)***
1888 ***(Time 1110)***
1888 ***(Time 1115)***
1888 ***(Time 1120)***
#
FMC400_TrP1CapCalc
0
0
0
0
0
0
0
#
FMC400_TrP2CapCalc
0
0
0
0
0
0
0
#
FMC400_TrInFl
0
0
0
0
#
FMC400_TrW1OFlow
0
0
0
0
#
FMC400_TrRain
0
0
0
0
0
0
0
#
0306271120 ***(Endtime for trends, shall be used as next trendrequest)***

2.5.8.6.2 Last trendblock

(Start)
NDA ***(No more trenddata)***

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.8.7 Cicode example

See Appendix 3

2.5.8.8 Debug messages

```
Fri Jun 27 11:21:47 2003 03:29:09.475 Transmit Length 31
01 20 31 20 57 47 48 20 30 33 20 30 36 20 32 37       . 1 WGH 03 06 27
20 31 30 20 34 30 20 23 20 31 30 33 38 20 0D          10 40 # 1038 .
Fri Jun 27 11:21:48 2003 03:29:10.231 Received Length 249
02 20 33 20 36 20 32 37 20 31 30 20 34 35 20 31       . 3 6 27 10 45 1
38 38 37 20 30 20 30 20 30 20 30 20 30 20 21 20       887 0 0 0 0 0 !
33 20 36 20 32 37 20 31 30 20 35 30 20 31 38  38      3 6 27 10 50 188
37 20 30 20 30 20 30 20 30 20 30 20 21 20 33 20       7 0 0 0 0 0 ! 3
36 20 32 37 20 31 30 20 35 35 20 31 38 38 37 20       6 27 10 55 1887
30 20 30 20 30 20 30 20 30 20 21 20 33 20 36 20       0 0 0 0 0 ! 3 6
32 37 20 31  31 20 30 20 31 38 38 37 20 30 20 30      27 11 0 1887 0 0
20 30 20 30 20 30 20 21 20 33 20 36 20 32 37 20        0 0 0 ! 3 6 27
31 31 20 35 20 31 38 38 38 20 30 20 30 20 30 20       11 5 1888 0  0 0
30 20 30 20 21 20 33 20 36 20 32 37 20 31 31 20       0 0 ! 3 6 27 11
31 30 20 31 38 38 38 20 30 20 30 20 30 20 30 20       10 1888 0 0 0 0
30 20 21 20 33 20 36 20 32 37 20 31 31 20 31 35       0 ! 3 6 27 11 15
20 31 38 38 38 20 30 20 30 20 30 20 30 20 30 20        1888 0 0 0 0 0
21 20 33 20 36 20 32 37 20 31 31 20 32 30 20 31       ! 3 6 27 11 20 1
38 38 38 20 30 20 30 20 30 20 30 20 30 20 21 20       888 0 0 0 0 0 !
23 20 31 30 31 39 31 20 0D                            # 10191 .
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.5.9 CreateIniFile

2.5.9.1 General

2.5.10 ReadSetpoint

2.5.10.1 General

This datatype is designed if you need to make an update of all channel data in the .ini file e.g. after that you have made some manually setting in the RTU.
This type uses the standard channel readfunction. The driver normally saves the data in the channelcache. When this function is used it also takes the cachedata and copies the data to the .ini file. Before it saves the channeldata it makes a copy of the .ini file to a .bak file. The file will be saved in the same path as the .ini file.

2.5.11 WriteSetpoint

2.5.11.1 General

This datatype is designed if you need to make an update of all channel data in the RTU e.g. after a cold start.
WriteSetpoint makes first a copy of the .ini channel data to the channel cache. When the data is copied it makes a normal write to the RTU.

2.5.12 CompareSetpoint

2.5.12.1 General

This datatype is designed for the situation there you want to verify that the channeldata in RTU is the same as you have in your .ini file.

It takes the data from the cache and compares this data with the data in the .ini file for the specific RTU. If you use this command direct after a startup it should always show the status OK and write "No difference" in the logfile depending of that the cache is always updated from .ini when Citect startup.
Make an update of the cache through a page with some channeldata or make a manual ReadSetpoint. After this the cache should be updated with the RTU data. After a CompareSetpoint command it will show you the text "Difference, Look in logfile" or OK. In the logfile you will see the date, time, channelnumber and the datadifference between the RTU/cache and the the .ini file
The log file will be saved under the same path as the .ini file itself and the file extension will be changed to .log from .ini. Eg file FMC400_C60005SE.INI will be changed to FMC400_C60005SE.LOG

2.5.12.2 Example of a logfile

030916 15:09:12
No difference
030916 15:09:36

Ch22 Rtu D 1	Ini D 0
Ch31 Rtu 2 2.29	Ini 2 2.27
Ch32 Rtu 2 1.00	Ini 2 0.00
Ch33 Rtu 2 0.30	Ini 2 0.12
Ch54 Rtu 2 23.00	Ini 2 0.00

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.6 DBFs

2.6.1 Help.dbf

TYPE	DATA	FILTER
PROTOCOL	AQUA COM	

2.6.2 Aquacom.dbf Entries

TEMPLATE	UNIT_TYPE	RAW_TYPE	BIT_WIDTH	LOW	HIGH	COMMENT
St%<16U(0,1,128)[/%+U(0,1,25)%*256]	0x00000001	4	32	0	0	Status
Ch%<16U(0,1,500)	0x00000002	7	256	0	0	Channel - string
Ch%<16U(0,1,500)	0x00000002	2	32	0	0	Channel - real
Ch%<16U(0,1,500)	0x00000002	1	16	0	0	Channel - integer
Rp%<16U(0,1,128)/%+U(0,1,5)%*256[/%+U(0,0,3)%*256]	0x00000003	4	32	0	0	Periodic report
TrReq	0x00000004	7	88	0	0	Trendrequest eg 0303180910
Pu%<16U(0,1,128)[/%+U(0,1,25)%*256]	0x00000005	7	256	0	0	Pump control
Al%<16U(0,1,9999)	0x00000006	0	1	0	0	Alarm
AlarmCode	0x00000007	7	48	0	0	Alarmcode priorities
CreateIniFile	0x00000008	1	16	0	0	CreateIniFile
ReadSetpoint	0x00000009	1	16	0	0	ReadSetpoint
WriteSetpoint	0x0000000a	1	16	0	0	WriteSetpoint
CompareSetpoint	0x0000000b	7	256	0	0	CompareSetpoint

2.6.3 Protdir.dbf

TAG	FILE	BIT_BLOCK	MAX_LENGTH	OPTIONS
ACUACOM	AQUACOM	256	256	0x10004f

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

2.7 Development resources

2.7.1 Contacts

2.7.1.1 Technical support

Autic System AB
Box 426
SE-201 24 Malmö
support@autic.se
www.autic.com

ITT Flygt AB
Box 2058
SE-291 02 Kristianstad
+46 (0)44-18 78 00

2.7.1.2 General questions

PiiGAB Processinformation
Anders Carlssons gata 14
S-417 55 Göteborg
info@piigab.se
www.piigab.se

2.7.2 Documents

AquaCom Technical description
October-2002 Rev 4

2.7.3 Driver Version History

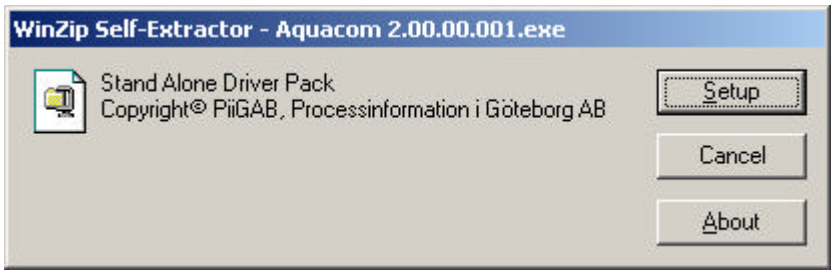
Driver Version	Comments
2.00.00.001	Initial release
2.00.00.002	Channel range changed from 1-255 to 1-500

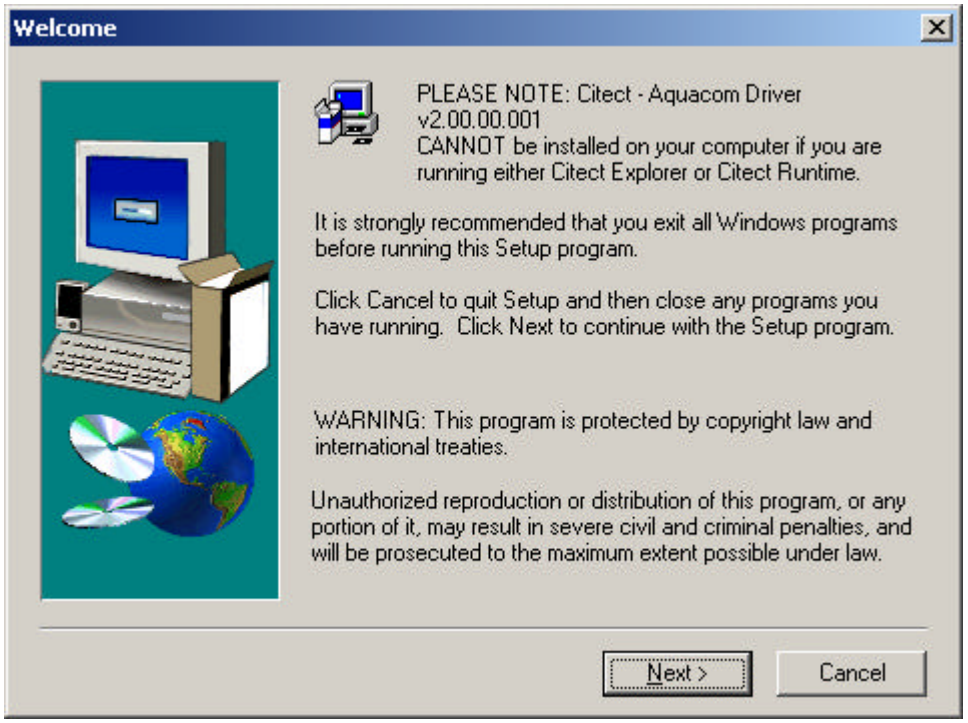
© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

3. Appendix 1 - Installation

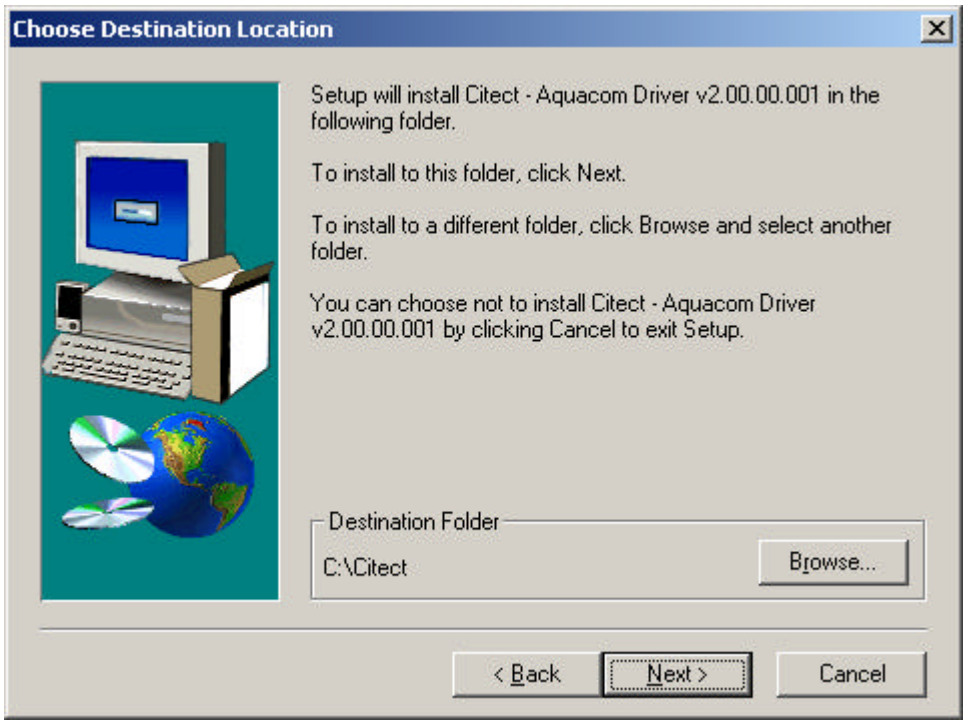
3.1 Beginning of installation

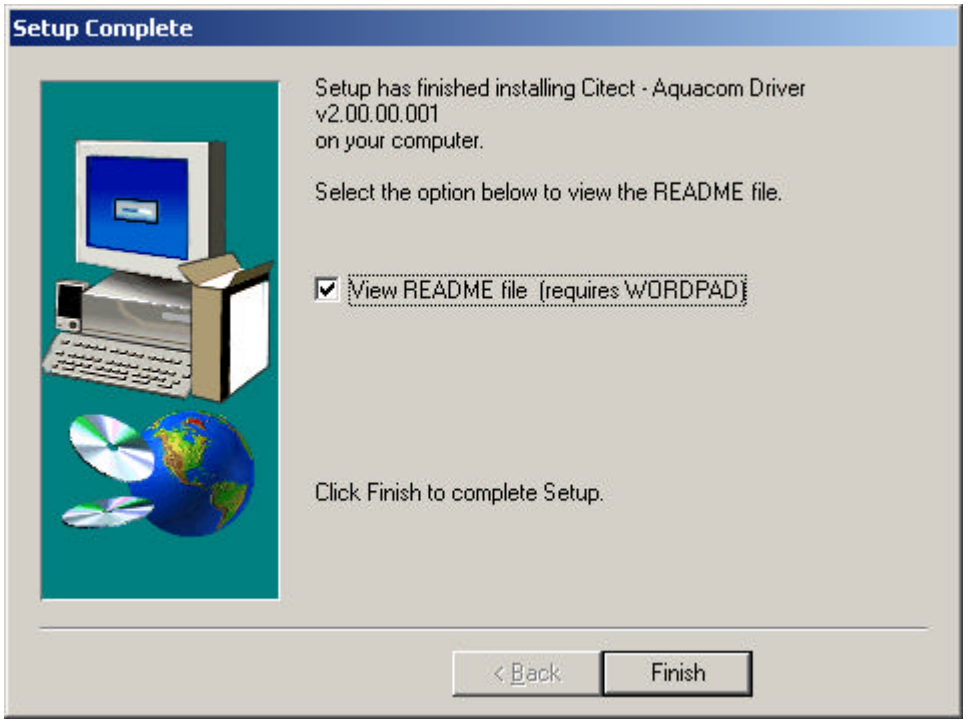
You are starting the installation by activating the self-extracting .exe file Mbuscit1.00.00.001.exe. The information should be self-explaining during the installation process. But anyway here is little short information about it.





© Copyright 2003 PiiGAB, Processinformation i Göteborg AB





© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

3.2 Files to copy

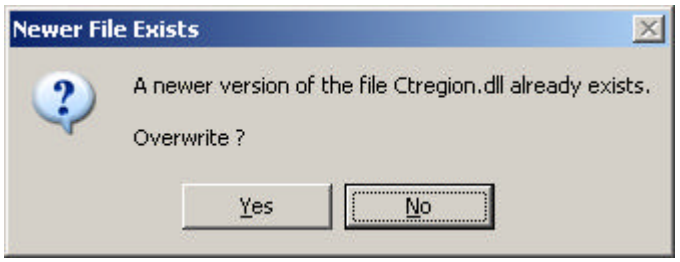
Before the files starting to be copied you will have a dialog box with similar information as below.

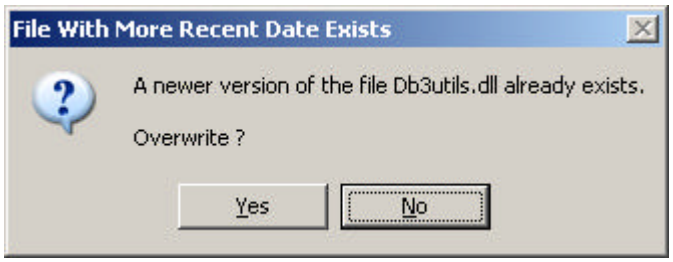
Destination Folder:
c:\cit542
Required disk space:
587 Kb

The information below is required by advanced users only.
The following files will be added:
AQUACOM.DBF will be updated in Bin
Aquacom.dll will be updated in Bin
Ctregion.dll will be updated in Bin and backed up to DrvBack
Db3utils.dll will be updated in Bin
EXPRBRD.DBF will be updated in Bin and backed up to DrvBack
EXPRWIZ.DBF will be updated in Bin and backed up to DrvBack
HELP.DBF will be updated in Bin and backed up to DrvBack
PROTDIR.DBF will be merged in Bin, backed up to DrvBack and copied to User\Include
PROTERR.DBF will be updated in Bin and backed up to DrvBack
Readme.wri will be added to DrvDoc

3.3 File With More Recent Data Exists

If these two dialog boxes appear at the screen you should normally press the Nej or No button.

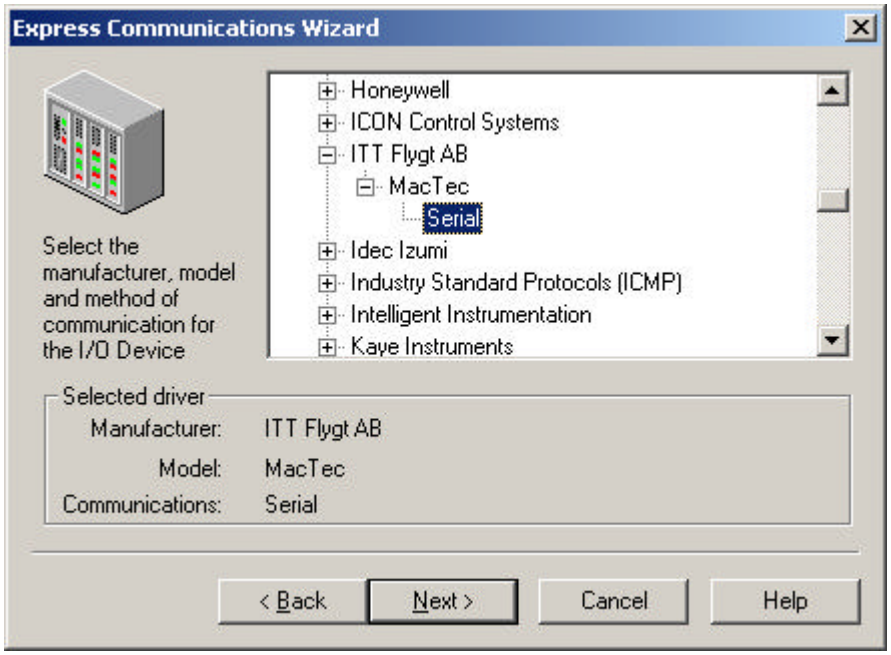




© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

3.4 Installation in Citect project

When the driver is installed in this way you can choose ITT Flygt AB from the standard Express Communication Wizard when you shall make a new project in Citect.



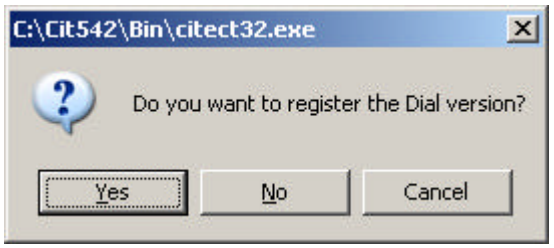
© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

4. Appendix 2 - Software Protection

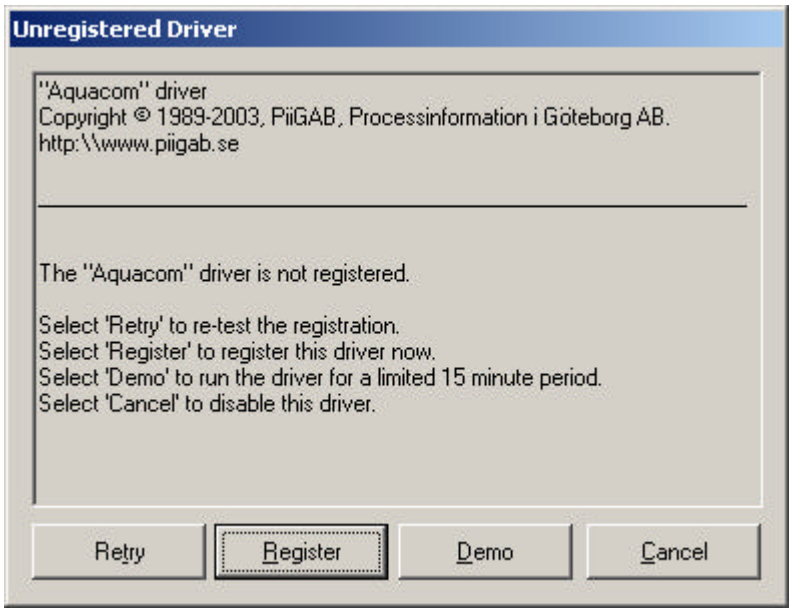
4.1 Unregistered driver

When a driver is run for the first time, or each time an unregistered driver runs, an 'Unregistered driver' dialog box will be displayed.
In the Aquacom driver a small message box will appear as default before the main registration dialog boxes. This message box will tell the system if it shall register a dial or fixed key. If you chose 'Yes' it will look for the dial key and if you chose 'No' it will look for the fixed key. Cancel will give you the possibility to chose Demo mode.

If you don't want this small message box shall be shown you can change the driver specific parameter 'Dial registration' to Dial or Fixed. See that capital for more information



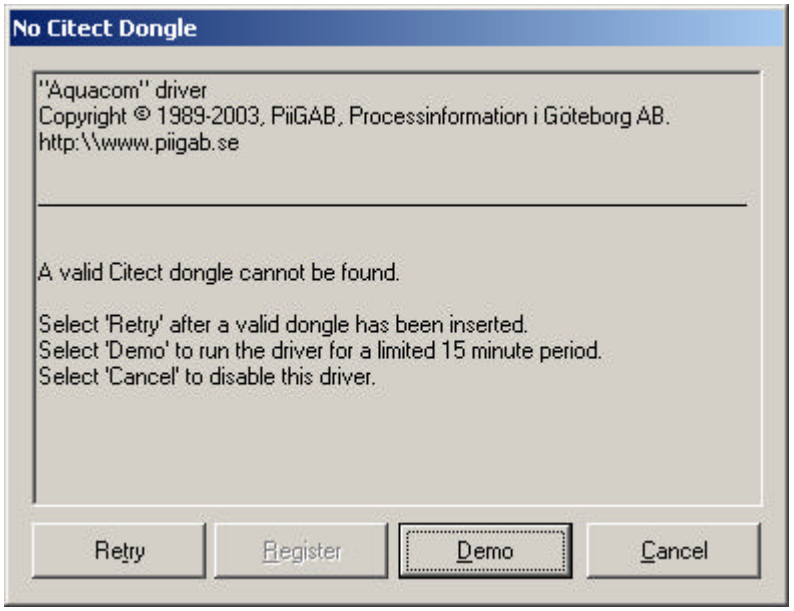
This dialog box will prompt you to either:
a) Let the driver re-test the registration settings
b) Register the driver
c) Run in a demo mode
d) Disable the driver



© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

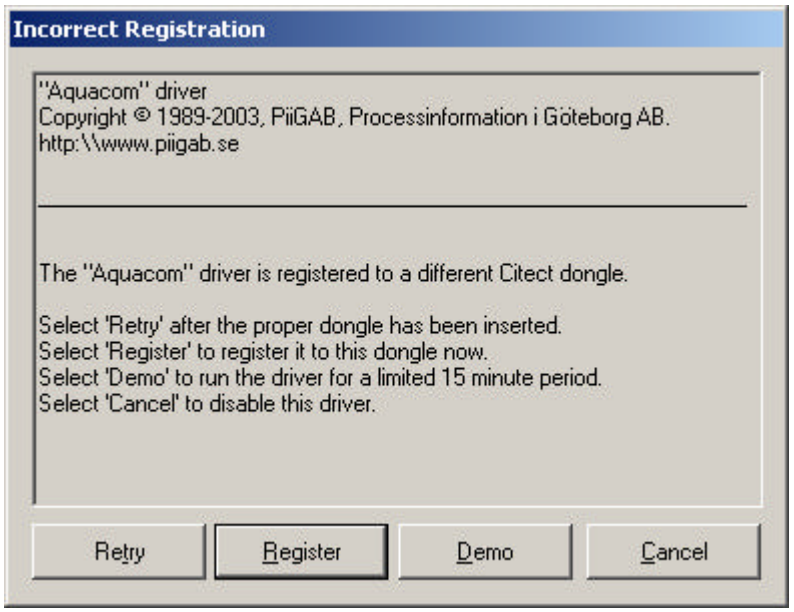
4.2 Citect Dongle not found

If there is no Citect dongle attached to local parallel port, or the dongle cannot be detected, then the 'No Citect Dongle' dialog box will be displayed. This dialog box will prompt you to either:
a) Insert a valid Citect dongle and retry
b) Run in a demo mode
c) Disable the driver



4.3 Incorrect registration

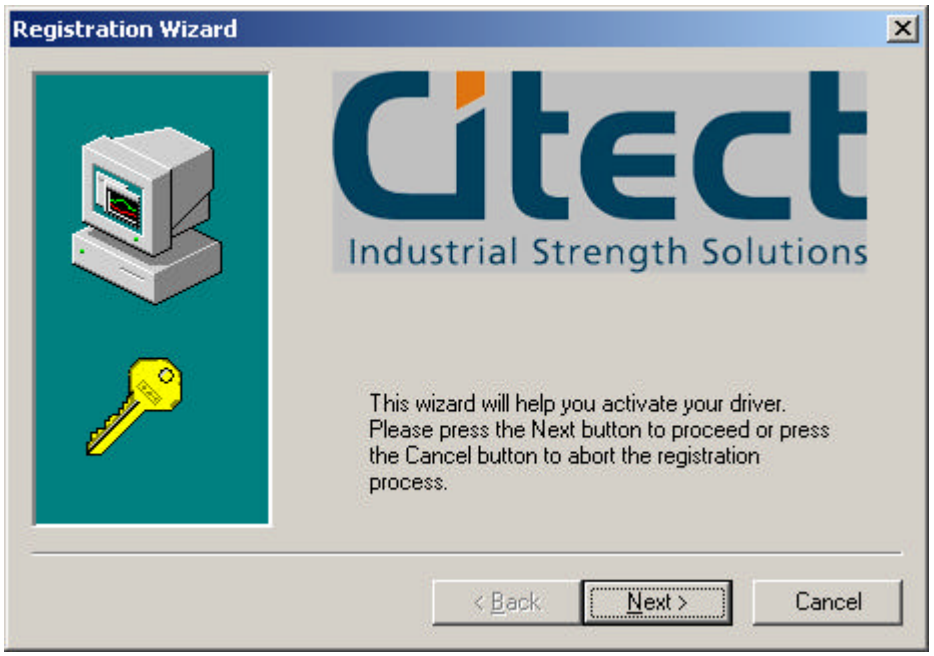
If the Citect dongle does not match the registration key, then the 'Incorrect registration' dialog box will be displayed. This dialog box will prompt you to either:
a) Replace the dongle and retry
b) Register the driver to the current attached dongle
c) Run in a demo mode
d) Disable the driver

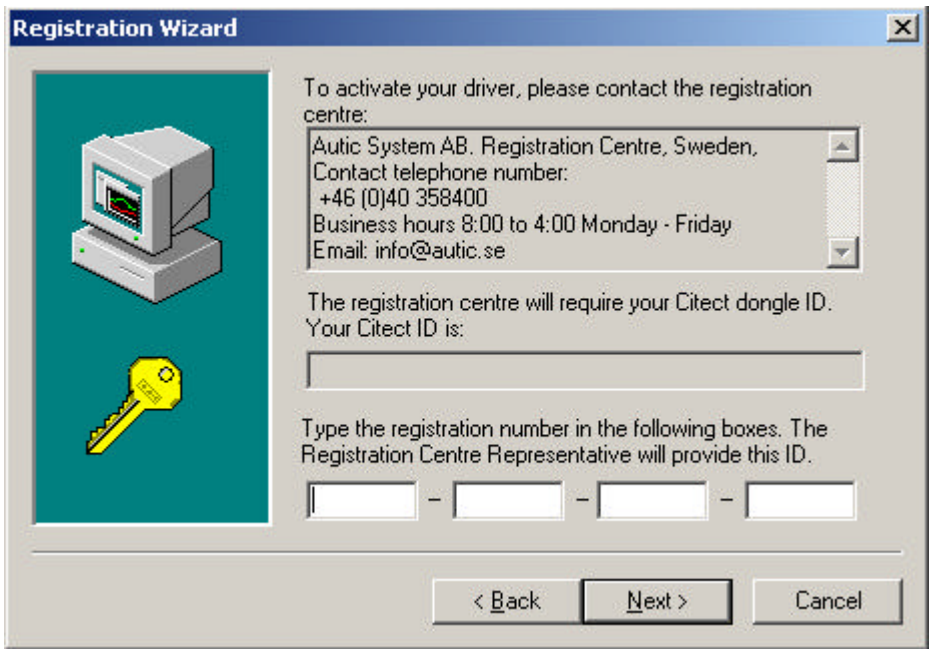


© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

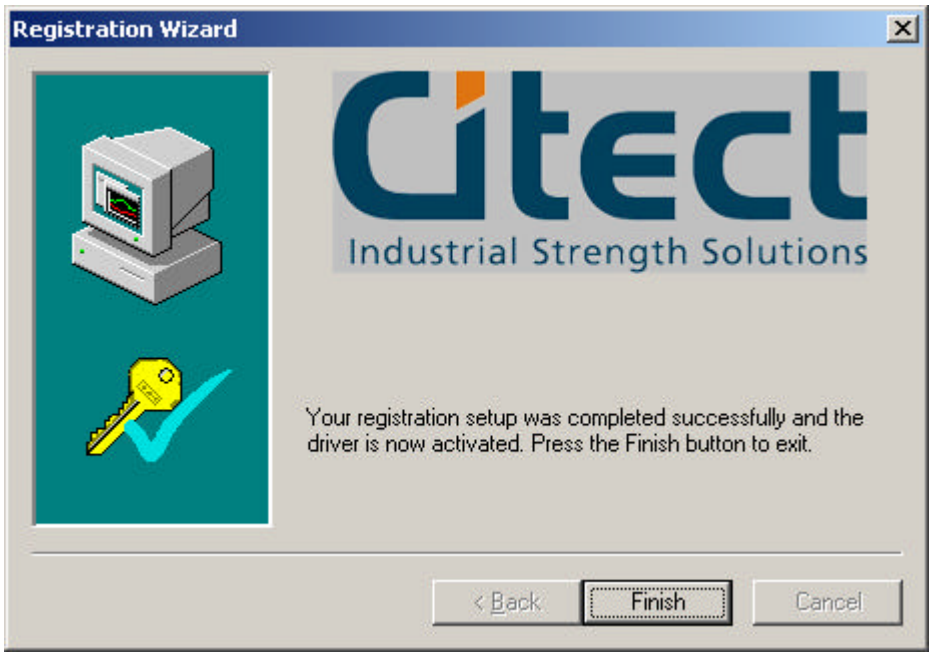
4.4 Registration wizard

If the user selects the 'Register' button on the dialog box, the 'Registration Wizard' will be activated which guides you through the registration process. If you have entered an invalid registration number, then a fail message will be displayed as shown, otherwise a successful message will be displayed and the registration number will be saved in the group [DRIVER_REGISTRATION] in Citect.ini file.





© Copyright 2003 PiiGAB, Processinformation i Göteborg AB



If you cancel the registration process, the 'Unregistered Driver' dialog box will remain open. If the 'Finish' button is pressed (process finished successfully), the dialog box will close automatically.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

4.5 Changing from fixed to dial registration

As soon you have finished a registration with successfully results the registration dialog boxes will never appear again. To change a registration you have to first disable the old
Aquacom registration number from the group [DRIVER_REGISTRATION] in Citect.ini file.

4.6 Information of what is registered

If you need to know if fixed or dial is registered so this is shown as a driver generated statistics parameter. If Fixed is registered the parameter Fixed Registered = 1 and if Dial is registered the parameter Dialed Registered = 1.

4.7 Demo Mode

If you choose the 'Demo' button from the dialogue, then the driver will run in demo mode for period of time.

4.8 Disable the Driver

If you choose the 'Cancel' button from the dialog, the driver will be put in 'Channel Offline' mode. This means the driver will report the 'Channel Offline' hardware alarm whenever the 'WatchTime' parameter triggers, which is normally every 30 seconds.

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

5. Appendix 3 – Cicode example for Trends

This example shown a way to read the logfile Aquatrend.log and merge the data in Citects trendsystem. This code is written by Autic System AB.

```
MODULE
REAL TrendArray[100];   //

INT hTrenddata;                  // File handle
INT iNoOfTrend;
INT iSample;                      // No of Trend samples

STRING Trendname;
STRING StartDate;
STRING TrendPath;

INT
FUNCTION Store_trend_from_FMC(STRING DateTime)
INT hFilData;
INT x,y;
       // Initial values
       StartDate=DateTime;
       TrendPath=ParameterGet("AQUACOM","TrendPath","[BIN]:");

       ! If no \ is set at the end of the path. put it there
       IF (StrRight(TrendPath,1)<>"\") THEN
              TrendPath= TrendPath+ "\";
       END

       TrendPath=TrendPath+"Aquatrend.log";

       WHILE Y < 1000 DO
              IF Fetch_trend_file(StartDate)= 9 THEN
                     FileClose(hTrenddata);
                     Prompt("@(Trend is ready)");
                     RETURN 0;
              END
              FOR x=0 TO iNoOfTrend-1 DO
                     Read_and_store();
              END
              FileClose(hTrenddata);
              Prompt("@(More trends to fetch)");
              Y=Y+1;
       END
END

PRIVATE
INT
FUNCTION Fetch_trend_file(STRING DateTime)
STRING sSecondRow;
STRING EndDate,sLine;
INT RowCounter,i;


       IF FileFind(TrendPath,0)<>"" THEN
              FileDelete(TrendPath);
       END

       Trend_Req=DateTime;
       Prompt("@(Waiting for trends from RTU)");

       WHILE FileFind(TrendPath,0)="" AND i < 100 DO
              Sleep(1);
              i=i+1;
       END

       hTrenddata=FileOpen(TrendPath,"r");
       IF hTrenddata < 0 THEN
              ErrLog("Could not open the trend log file
"+TrendPath+"Aquatrend.log");
              RETURN -1;
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB

```
       END
       IF StrSearch(0,FileReadLn(hTrenddata),"#")<>-1  THEN
                     sSecondRow=StrLeft(FileReadLn(hTrenddata),4)
                     IF sSecondRow="NDA" THEN
                            ErrLog("NDA Found");
                            RETURN 9;
                     ELSE
                            iSample=StrToInt(sSecondRow);
                     END
                     EndDate=StrLeft(FileReadLn(hTrenddata),10);
       ELSE
                     ErrLog("Gick ej att hitta styrtecken");
                     RETURN -1;
       END
       RowCounter=0;
       WHILE NOT FileEOF(hTrenddata) DO
              RowCounter=RowCounter+1;
              StartDate=FileReadLn(hTrenddata);
       END

       ParameterPut("AQUACOM", "LastTrendTime", StartDate);

       iNoOfTrend=(RowCounter-5)/(iSample+2)+1;

       FileSeek(hTrenddata,0);
    sLine=FileReadLn(hTrenddata);
    sLine=FileReadLn(hTrenddata);
       sLine=FileReadLn(hTrenddata);

       Prompt("@(Createing Trends)");
       RETURN 1;
END

PRIVATE
FUNCTION Read_and_store()
STRING sline;
INT x;  //Loop index
INT iSampleCheck,iDate;
STRING sTime;
INT iTimevar;

       IF StrSearch(0,FileReadLn(hTrenddata),"#")<>-1 THEN
              sline=FileReadLn(hTrenddata);
              Trendname=StrTrim(sLine);
              FOR x = 0 TO iSample-1 DO
             // Store in a backward order to get it right in the trenddatabase
            sLine=FileReadLn(hTrenddata);
                     TrendArray[iSample-x-1]=StrToReal(sline);
              END
       ELSE
                     ErrLog("Could not find control characters");
       END


       iDate=StrToDate(StrLeft(StartDate,2)+"-"+StrMid(StartDate,2,2)+"-
"+StrMid(StartDate,4,2));
       sTime=StrMid(StartDate,6,2)+":"+StrRight(StartDate,2);
       iTimevar=iDate+StrToTime(sTime);
       ErrLog(" Store Trend "+TrendName+" Time " + StartDate +" No of samples "
+ IntToStr(iSample));
       //TrnSetTable(Tag, Time, Period, Length, Table, Milliseconds)
       ErrSet(1);
       iSampleCheck = TrnSetTable(TrendName,iTimevar,0,iSample,TrendArray[0],0)
       ErrSet(0);

       IF iSampleCheck <> iSample THEN
              ErrLog("Trend "+TrendName+" is not defined in the Citect
application ");
       END
END
```

© Copyright 2003 PiiGAB, Processinformation i Göteborg AB