



EZForecourt Developers Manual



Table of contents

1.	Version.....	7
1.1.	History	Error! Bookmark not defined.
2.	Introduction	7
3.	Concepts.....	8
4.	System Architecture	11
5.	Installation.....	13
6.	INI file configurations	13
6.1.	EZServer INI file	15
6.2.	EZClient INI file	17
6.3.	EZDriver INI file	17
6.4.	EZDB INI file.....	19
6.5.	EZBackOffice INI file	21
6.6.	EZLoader INI file	22
6.7.	EZLicense INI file	22
7.	Developer's Kit contents.....	23
7.1.	The EZSim pump simulator.....	23
7.1.1.	EZSim INI file	24
7.2.	Sample applications	25
7.2.1.	EZDemoPos.....	25
7.2.2.	EZClientVBTest	25
7.2.3.	EZDBVBClient	25
8.	API reference.....	26
8.1.	API Components	26
8.2.	Data types	26
8.3.	API Definition.....	27
8.3.1.	ClientLogon/ClientLogonEx	27
8.3.2.	ClientLogoff.....	30
8.3.3.	ClientStatus.....	31
8.3.4.	TestConnection.....	32
8.3.5.	LicenseStatus	33
8.3.6.	GetIniValue	34
8.3.7.	SetIniValue.....	35
8.3.8.	ProcessEvents	36
8.3.9.	GetEventsCount.....	37
8.3.10.	GetNextEventType	38
8.3.11.	DiscardNextEvent.....	39
8.3.12.	GetNextPumpEvent / PumpStatusEvent / StatusEvent	40
8.3.13.	GetNextDeliveryEvent[Ex] / DeliveryEvent[Ex]	43
8.3.14.	GetNextServerEvent / ServerEvent.....	46
8.3.15.	GetNextClientEvent/ClientEvent	47
8.3.16.	FireClientEvent	48
8.3.17.	GetNextDBLogEvent/DBLogEvent.....	49
8.3.18.	GetNextDBLogDeliveryEvent/DBLogDeliveryEvent.....	51
8.3.19.	GetNextDBCclearDeliveryEvent/DBCclearDeliveryEvent	53
8.3.20.	GetNextDBStackDeliveryEvent/DBStackDeliveryEvent.....	55
8.3.21.	GetNextDBHoseETotalsEvent/DBHoseETotalsEvent	56
8.3.22.	GetNextDBTriggerEvent/DBTriggerEvent.....	58
8.3.23.	GetNextDBAttendantLogonEvent / DBAttendantLogonEvent.....	59
8.3.24.	GetNextDBAttendantLogoffEvent / DBAttendantLogonEvent.....	60

8.3.25.	GetNextDBTankStatusEvent / DBTankStatusEvent	61
8.3.26.	GetPumpsCount	63
8.3.27.	GetPumpByName.....	64
8.3.28.	GetPumpByNumber	65
8.3.29.	GetPumpByOrdinal.....	66
8.3.30.	GetPumpProperties	67
8.3.31.	SetPumpProperties	70
8.3.32.	DeletePump	74
8.3.33.	GetPumpHosesCount/GetHosesCount.....	75
8.3.34.	GetPumpHoseByNumber/GetHoseByNumber.....	76
8.3.35.	GetPumpStatus	77
8.3.36.	GetAllPumpStatuses	79
8.3.37.	GetPumpDeliveryProperties[Ex]/GetDeliveryProperties[Ex]	81
8.3.38.	PrepayReserve.....	84
8.3.39.	PrepayCancel.....	86
8.3.40.	PrepayAuthorise	88
8.3.41.	PreauthReserve.....	90
8.3.42.	PreauthCancel.....	92
8.3.43.	PreauthAuthorise.....	93
8.3.44.	LoadPreset	95
8.3.45.	AttendantAuthorise	97
8.3.46.	Authorise.....	99
8.3.47.	CancelAuthorise	101
8.3.48.	TempStop	102
8.3.49.	ReAuthorise	103
8.3.50.	TerminateDelivery	104
8.3.51.	StackCurrentDelivery.....	105
8.3.52.	GetHosesCount	107
8.3.53.	GetHoseByOrdinal.....	108
8.3.54.	GetHoseProperties	109
8.3.55.	SetHoseProperties	111
8.3.56.	DeleteHose	113
8.3.57.	GetDeliveriesCount	114
8.3.58.	GetDeliveryByOrdinal	115
8.3.59.	GetDeliveryProperties/GetDeliveryPropertiesEx.....	116
8.3.60.	SetDeliveryProperties/SetDeliveryPropertiesEx	118
8.3.61.	LockDelivery	121
8.3.62.	UnlockDelivery.....	122
8.3.63.	ClearDelivery	123
8.3.64.	LockAndClearDelivery.....	125
8.3.65.	GetGradesCount	127
8.3.66.	GetGradeByNumber.....	128
8.3.67.	GetGradeByName	129
8.3.68.	GetGradeByOrdinal	130
8.3.69.	GetGradeProperties	131
8.3.70.	SetGradeProperties.....	132
8.3.71.	DeleteGrade	134
8.3.72.	SetGradePrice	135
8.3.73.	GetGradePrice.....	137
8.3.74.	GetTanksCount	138
8.3.75.	GetTankByNumber.....	139
8.3.76.	GetTankByName	140
8.3.77.	GetTankByOrdinal	141
8.3.78.	GetTankProperties	142



8.3.79.	SetTankProperties.....	144
8.3.80.	DeleteTank	147
8.3.81.	GetPortsCount.....	148
8.3.82.	GetPortByNumber	149
8.3.83.	GetPortByName	150
8.3.84.	GetPortByOrdinal	151
8.3.85.	GetPortProperties.....	152
8.3.86.	SetPortProperties	154
8.3.87.	DeletePort.....	156
8.3.88.	GetAttendantsCount.....	157
8.3.89.	GetAttendantByNumber	158
8.3.90.	GetAttendantByName.....	159
8.3.91.	GetAttendantByOrdinal.....	160
8.3.92.	GetAttendantProperties.....	161
8.3.93.	SetAttendantProperties	162
8.3.94.	DeleteAttendant.....	164
8.3.95.	AttendantLogon	165
8.3.96.	AttendantLogoff	167
8.3.97.	AllStop	168
8.3.98.	AllStopfIdle	169
8.3.99.	AllAuthorise	170
8.3.100.	AllReAuthorise.....	171
8.3.101.	SetNextDeliveryID	172
8.3.102.	ResultString	173
8.3.103.	PumpStateString	174
8.3.104.	DeliveryStateString.....	175
8.3.105.	DeliveryTypeString	176
8.3.106.	ReserveTypeString.....	177
9.	Database reference	178
9.1.	Introduction.....	178
9.2.	Data Model	179
9.3.	Tables reference	180
9.3.1.	AttendantHoseReports.....	181
9.3.2.	AttendantReports	182
9.3.3.	Attendants	183
9.3.4.	DeviceTypes	184
9.3.5.	EventTypes.....	185
9.3.6.	Events	186
9.3.7.	GradePriceLevels	187
9.3.8.	GradePrices	188
9.3.9.	GradeProfiles	189
9.3.10.	Grades.....	190
9.3.11.	HoseDeliveries	191
9.3.12.	HoseReports.....	192
9.3.13.	Hoses.....	195
9.3.14.	Ports	196
9.3.15.	PriceSignTypes	197
9.3.16.	PriceSigns	198
9.3.17.	Protocols.....	199
9.3.18.	PumpGroups	200
9.3.19.	PumpProfiles	201
9.3.20.	PumpTypes	202
9.3.21.	Pumps.....	203
9.3.22.	ReportTypes	204



9.3.23.	Reports	205
9.3.24.	TankDeliveries	206
9.3.25.	TankGaugeTypes	207
9.3.26.	TankGauges	208
9.3.27.	TankReports	209
9.3.28.	Tanks	210
9.4.	Stored procedures reference	211
9.4.1.	ezspAddHoseReport	212
9.4.2.	ezspAttendantLogoff	213
9.4.3.	ezspAttendantLogon	214
9.4.4.	ezspClearHoseDelivery	215
9.4.5.	ezspCloseReport	216
9.4.6.	ezspGradeProfileChange	217
9.4.7.	ezspInitDB	218
9.4.8.	ezspLogHoseDelivery	219
9.4.9.	ezspNightly	220
9.4.10.	ezspProcessHoseETotals	221
9.4.11.	ezspProcessMeterReadings	222
9.4.12.	ezspProcessTankDip	223
9.4.13.	ezspProcessTankDrop	224
9.4.14.	ezspPumpGroupChange	225
9.4.15.	ezspPumpProfileChange	226
9.4.16.	ezspStackHoseDelivery	227
10.	Appendices	228
10.1.	Appendix A – Pump states	228
10.2.	Appendix B – Pump reserves	230
10.3.	Appendix C – Delivery types	231
10.4.	Appendix D – Delivery states	232
10.5.	Appendix E – Event types	233
10.6.	Appendix F – Pump display formats	235
10.7.	Appendix G – Pump authorization modes	236
10.8.	Appendix H – Pump delivery stack (memory) modes	237
10.9.	Appendix I – Pump limit types	238
10.10.	Appendix J – Permitted hoses mask	239
10.11.	Appendix K – Tank Types	240
10.12.	Appendix L – Device Types	241
10.13.	Appendix M – Error messages	242
10.14.	Appendix N – Client event types	246



**EZForecourt Developers Manual
Version 1.3**

© Copyright EZTech Ltd.
2005 – 2006
all rights reserved
www.eztech.com.br



Version History

Version 1.00	August 2005
Version 1.20	November 2005
Version 1.30	July 2006

1. Introduction

The EZForecourt product is a forecourt controller designed to work in conjunction with third party windows XP/2000 based POS systems, and possesses the following characteristics.

USB interface.

It utilizes a simple and small, USB based, forecourt interface module (EZMod) to easily connect a windows XP/2000 based server to common gas station forecourt devices. The use of the USB standard does away with the need to install any additional proprietary hardware in the host server.

Self reliant device.

The EZMod module is a self powered USB device which has the ability to buffer up to 2950 deliveries if the connection to the host is lost, hence providing redundancy and independence from the server.

Generic high level interface.

It can control and manage many common types and makes of Petrol pumps in a transparent fashion, all the host application sees is a generic pump device.

Electronic tank gauge support.

It also interfaces to many different types and makes of electronic tank gauges, providing a generic and simple interface for both alarms and tank statistics.

ActiveX, .NET and DLL client interfaces.

The interface between EZForecourt and the host system/application is done via two ActiveX controls, EZClient.OCX and EZPump.OCX, a .NET package EZTech.dll which contains EZTech.EZClient and EZTech.EZPump, or a standard windows a DLL, EZClient.DLL. Which of the three options is utilized will depend on the capabilities and requirements of the host system.

TCP/IP client server interface.

The interface between the EZForecourt server and client applications is done via TCP/IP sockets, hence the clients and the server need not be running on the same physical machine, but can be anywhere on the same local TCP/IP based network.

Optional back office and database support.

EZForecourt comes with an optional back office and database, which when used, can dramatically reduce the amount of work and time required to bring a fully featured forecourt system to market. However if the host system already has the necessary support applications and database structure then these can be easily utilized, preserving a consistent look and feel across the entire back office, and reducing the cost of EZForecourt.

2. Concepts

Prior to starting an integration of EZForecourt, it is necessary that a few of the basic concepts be clarified first, to ensure that the more technical information be interpreted correctly. The following terms will be used many times in the API and data base sections later on in this document.

Postpay delivery

Deliveries on a forecourt generally follow the Post pay sequence of events, the hose is pulled on the pump and the pump is either authorized automatically or manually by a shop or forecourt attendant. When the delivery is completed the customer either pays the attendant for the delivery or pays for it in the shop. This is referred to as a post pay delivery, i.e. a delivery that is paid for after it is taken.

Prepay delivery

An alternative to paying for the delivery after it is taken is paying for it before. In this case the customer goes into the shop, pays a predetermined amount for fuel on a specified pump, this is usually an estimate on behalf of the customer, who then goes back to his car and takes the delivery. When the hose is pulled from the specified pump, it is automatically authorized to delivery up to the prepaid amount. If the delivery is terminated prior to the limit being reached, a refund for the difference is automatically generated and the customer is then required to return to the shop to collect the refund. If however there is no refund the customer is free to leave when the hose is returned.

Preauth delivery

These days many systems give the customer the option of paying for the fuel delivery at the pump, to do this, the pump must be fitted with a card terminal. The sequence of events in this situation is, the customer passes their card thru the card terminal on the respective pump, and then the card terminal starts a pre-authorization with the respective bank. Once this pre-authorization is granted, the pump is authorized to delivery up to the limit returned with the pre-authorization response. When the customer has completed the delivery the actual value of the delivery is sent as part of a pre-authorization advice message billing the customer's card with the correct value. This is known as a preauth delivery.

Drive off delivery

Unfortunately there are also customers, who take a delivery at a pump and leave without paying, this only occurs in post pay environments. These are politely referred to as drive off deliveries.

Test delivery

At times it is necessary for the various certification authorities to carry out tests on the pumps to verify that their volume measurement conforms to the relevant regulations. When these deliveries are completed the fuel is returned manually returned to the tank from which it came. These deliveries must be treated differently and are referred to as test deliveries.

Monitor deliveries

In a lot of countries gas station forecourts are run fully attended and the customer does not even have to exit the car to fill up the tank. A forecourt attendant takes the delivery on behalf of the customer and the customer pays the attendant directly after the delivery is completed. In this mode the forecourt controller is in essence only monitoring the forecourt activity, and is usually used to prevent fraud or theft by the forecourt attendants. These deliveries are referred to as monitor deliveries.

Offline deliveries

At times the communications with the pumps are lost; this could be because the pumps are turned off, switched into local mode or because of technical faults. When the pumps are returned to normal function and communications are re-established, an offline delivery is generated which is the sum of all of the deliveries which were done while the pump was offline. This total is difference between the last known electronic totals before it stopped responding and the electronic totals retrieved immediately after the pump started to respond. The purpose of this delivery type is to ensure that the electronic and theoretical totals still tally.

Temp stop

Pump deliveries can be controlled remotely via EZForecourt, imagine a situation where some one is smoking near a pump, and due to safety reasons it is necessary to stop the fuel delivery, this is called a 'Temp stop'. A temp stop remains in place until the pump is re-authorized manually via EZForecourt. All of the pumps can be stopped in single operation; this is called an 'All Stop', and can be reversed by an 'All Re-authorize'.

Pump authorize

EZForecourt permits pumps to run in several authorization modes, the first and most simple is auto authorize. In this mode the pump will start delivering, without intervention, when the hose is pulled. When EZForecourt is running in monitor mode the pumps are also auto authorize. The other mode is compulsory authorize, in this mode when the hose is pulled, the pump will call for authorization, it will only start delivering when a forecourt or shop attendant manually authorizes the pump. If a limit is placed on the number of unpaid deliveries, auto authorize pumps will not automatically authorize, if this limit has been reached. One or more of the unpaid deliveries will need to be processed first. This does not however apply to monitor mode.

Electronic totals

Most pump types keep electronic totals in non-volatile memory; these totals are accumulative totals of both volume and value, for each of the hoses on the pump. These totals are zeroed when the pump is installed or when the pump electronics are reset. These totals (when present) are interrogated remotely by EZForecourt and are continually monitored to determine if deliveries are lost etc.

Mechanical totals

Most pump types also have mechanical totals for both volume and value for each hose present on the pump. Unlike the electronic totals these cannot be reset, or interrogated remotely. EZForecourt permits the manual entry of these totals. This is yet another total to be reconciled against the EZForecourt totals.

Theoretical totals

Besides the mechanical and electronic totals maintained by the pumps, EZForecourt also maintains a theoretical total for both value and volume for each of the hoses configured. This is maintained by simply accumulating the individual delivery totals as they are completed. Reconciling the theoretical, electronic and mechanical totals is ones best defense against theft or fraud.

Price level

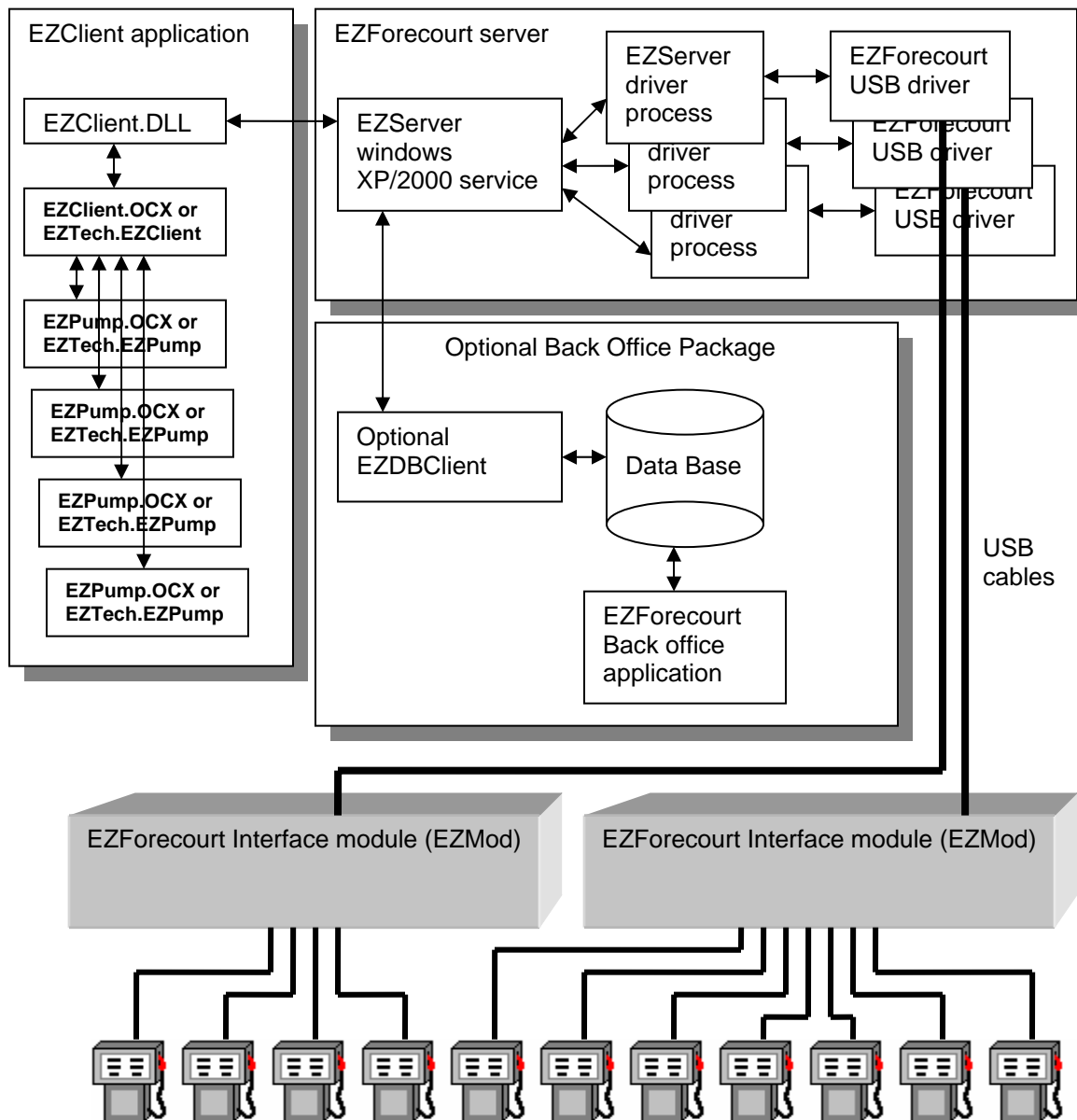
Most modern pump types today support more than one price per grade, this was originally intended to be used in credit or cash situations, however it can also be used for full vs. self service etc. Even though most pumps do not display both prices on the pump displays, they are saved internally and can be changed and selected remotely. EZForecourt supports multiple prices per grade and up to two prices per hose.



It is assumed that the reader is familiar with windows XP/2000 and general programming concepts and as such there is no explanation of this here.

3. System Architecture

The EZForecourt system architecture is as follows:





As can be seen from the diagram above, EZForecourt supports more than one EZMod module. This may be required if there is more than eight physical pumps or a mixture of pump types (different manufacturers) that cannot be hosted on the same EZMod module.

It also clearly demonstrates how the optional back office and database fit into the overall structure. If the integrator decides to use their own existing back office or develop a new one, then a similar structure will be required.

It may not be evident here but the links between the BackOffice and EZServer are also via the EZClient.DLL, and in the interests of speed and data integrity it is a requirement that the BackOffice and EZServer be hosted on the same machine. Other EZClients (POSes etc) do not need to be on the same machine, but can be anywhere on the same TCP/IP network.

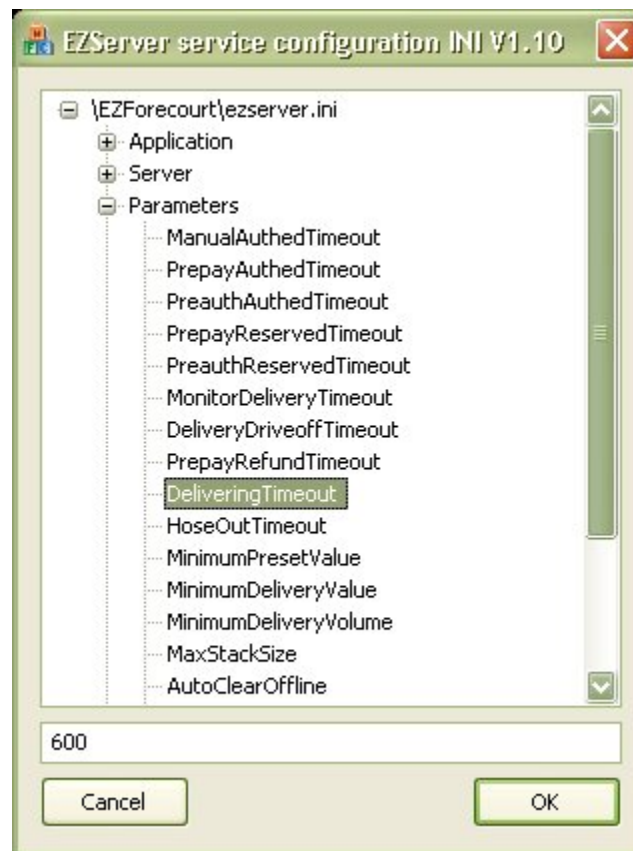
4. Installation

For a detailed description of how to install the EZForecourt product refer to the EZForecourt Installers Manual.

5. INI file configurations

Each of the EZForecourt software components is individually configured with a windows format INI file. These INI files all follow a standard format and are easily edited using the EZIniEditor utility provided. These INI files are installed per-configured and should only be edited under the direct instructions of EZTech support.

All of these INI files are located in the \EZForecourt directory. To edit an INI file simply open up a command window, change to the EZForecourt directory and then type EZIniEditor followed by the name of the INI file (with or without the INI extension). This will result in a screen similar to the following being presented:



If the INI file was located successfully, the name of the INI file will be displayed with a plus symbol beside the full file name, clicking the plus symbol will expand it to show all of the

sections contain in the INI file. Clicking the plus symbol beside each of the sections will expand it to show all of the individual configuration parameters.

The value for the currently highlighted configuration parameter is displayed in the text box at the bottom of the window. To change the currently selected configuration value, simply edit it in the text box, saving is done automatically when a different configuration parameter is selected or the application is exited by clicking the OK button.

There are some sections and configuration values which are common to all of the EZForecourt INI files these are as follows.

Section	Parameter	Description
[Application]	Name	The name of the application to which this INI file applies.
	IniVersion	The version number for this INI file
[Server]	Name	The machine name of the machine hosting the EZServer service, if the service is on the same machine then the value <local> can be used.
	CallPort	The TCP/IP port number that is used by EZClients to access the server. This number must be the same for all EZClients and the EZServer.
	EventsPort	The TCP/IP port number that is used by the EZClients to receive events. This number must be the same for all of the EZClients and the EZServer.
	CallTimeout	The timeout value in milliseconds for clients when calling the server, if the server does not respond within this time the call will return with a SERVER_TIMEOUT result.
[Log]	Screen	A 0 or 1 (0=no,1=yes) flag used to determine if diagnostic information is sent to the screen, this is only of use when running the server in debug mode and is reserved for internal EZTech use.
	File	A 0 or 1 (0=no, 1=yes) flag used to determine if diagnostic information is sent to a log file. By default this is 1 and should be left that way, unless instructed to change it by EZTech. Diagnosing faults is considerably more difficult if this flag is turned off.
	Socket	A 0 or 1 (0=no, 1=yes) flag used to determine if diagnostic information is sent to a TCP/IP socket. This feature is currently unsupported, but is reserved for future use.

5.1. EZServer INI file

The EZServer INI file contains all of the configurable parameters for the EZServer service. This INI file contains the standard parameters listed above along with the specific parameters listed below.

Section	Parameter	Description
[Parameters]	ManualAuthedTimeout	The amount of time, in seconds, that a compulsory authorize pump will remain authorized after manual authorization.
	PrepayAuthedTimeout	The amount of time, in seconds, that a pump will remain authorized for a prepay delivery, before a prepay refund is automatically generated. If the pump starts to deliver before this time is up it will be treated as a prepay delivery.
	PreauthAuthedTimeout	The amount of time, in seconds, that a pump will remain authorized for a preauth delivery, before the preauth reserve is automatically removed. If the pump starts to deliver before this time is up it will be treated as a preauth delivery.
	PrepayReservedTimeout	The amount of time, in seconds, that a pump will remain reserved for a prepay delivery before the prepay reserve is automatically removed. If the pump is prepay authorized before this time is up, it will change to the prepay authorized state.
	PreauthReservedTimeout	The amount of time, in seconds, that a pump will remain reserved for a preauth delivery before the preauth reserve is automatically removed. If the pump is preauth authorized before this time is up, it will change to the preauth authorized state.
	MonitorDeliveryTimeout	The amount of time, in seconds, that a Postpay delivery will remain on a pump, configured in monitor mode, before it is cleared automatically as a monitor delivery. It can be taken as a Postpay delivery prior to timing out.
	DeliveryDriveoffTimeout	The amount of time, in seconds, that a Postpay delivery can remain on a pump until it is flagged as a potential drive off delivery.
	PrepayRefundTimeout	The amount of time, in seconds, that a prepay refund remains on a pump before it is automatically cleared from the pump. If this delivery is cleared manually prior to this timeout expiring, the pump will remain locked out for the remainder of this timeout.

	DeliveringTimeout	The amount of time, in seconds, that a delivering pump can remain delivering before a warning is generated.
	HoseOutTimeout	The amount of time, in seconds, that a pump hose can be left out, not delivering, before a warning is generated.
	MinimumPresetValue	The minimum value accepted by the server for a preset, prepay or preauth delivery.
	MinimumDeliveryValue	The minimum delivery value accepted by the server for a valid delivery, deliveries with values smaller than this are discarded.
	MinimumDeliveryVolume	The minimum delivery volume accepted by the server for a valid delivery, deliveries with volumes smaller than this are discarded.
	MaxStackSize	The maximum number of unpaid deliveries permitted on a pump before the server will stop authorizing the pump. If the server is in standalone mode, or the pumps are in monitor mode, this limit is ignored.
	AutoClearOffline	A yes or no value, that determines whether offline deliveries are automatically cleared by the server, or must be taken manually as a Postpay delivery like any other.
	MinOfflineDeliveryVolume	The minimum volume for an offline delivery, if an offline delivery is detected with less than this volume, it is ignored.
	MaxOfflineDeliveryVolume	The maximum volume for an offline delivery, if an offline delivery very is detected with more than this it is ignored. This is used to discard offline deliveries generated when used pumps are first connected to the system.
	Standalone	A yes or no value to determine if the forecourt interface modules are allowed authorize pumps and buffer deliveries when the server is not running. This is only of use in a pump monitoring environments, and hence is of little use in a self service situation.
	NonVolObjects	A yes or no value to determine if the objects in the EZServer are non-volatile or not. If the EZBackOffice is used this should be No, otherwise it is recommended that it be yes.
	NonVolTotals	A yes or no value to determine if the hose electronic totals and tank gauge values are saved to the non-vol store or not, it is recommended that it be yes.

	LogDeliveries	A yes or no value to determine if the individual deliveries are logged to a separate log file.
	MaxDeliveryVolumeDiff	The maximum delivery volume difference between the reported delivery volume and the volume electronic totals difference before an offline delivery is generated.
	MaxDeliveryValueDiff	The maximum delivery value difference between the reported delivery value and the value electronic totals difference before an offline delivery is generated.
	MinimalPumpEvents	If set to yes this suppresses various pump events including running total events. This should be no.
	ExtendedDeliveryProperties	A flag to determine if the DeliveryEvent or DeliveryEventEx event is generated.
	RunningTotalRate	The minimum running total update rate for pumps in milli-seconds.
[DBClient]	name	The name of the data base client service extension executable run, if the back office option is activated.

5.2. EZClient INI file

The EZClient INI file contains all of the configurable parameters for all of the EZServer clients (EZClients). This INI file contains the entire standard parameters listed above but has no specific parameters. The main parameter of interest here is the name parameter in the Server section, in order to configure a remote client this parameter must be set to the machine name of EZServer service host machine.

5.3. EZDriver INI file

This INI file is also used by the EZServer service; more specifically it is use by the driver DLLs loaded at start up. The parameters in this INI are all pre-configured and should only be altered under specific instructions from EZTech. The following documentation is provided for information purposes only. Under some of the protocol and pump type sections you will find parameters which are not listed here, these are driver specific parameters, for details of these parameters please contact EZTech technical support.

This INI file contains the standard parameters listed above along with the specific parameters listed below.

Section	Parameter	Description
[Protocolnnn]	Name	The name of the protocol, nnn is from 001 to 999, and corresponds to the ProtocolID column in the Protocols table of the EZDB database.
	Driver	The name of the DLL file which is the driver for this protocol, note it must be a

		fully qualified file name.
	ProtocolType	A number used to identify the protocol as used by the forecourt interface modules EZMods
	LoopType	The physical loop type used for this protocol, the options are COM (native windows rs232 COM serial port), RS485 (Rs485 CL-RS485 EZMod), CL20 (20ma current loop, CL-RS485 EZMod), CL40 (40ma current loop, CL-RS485 EZMod), others will be added as new EZMods become available.
	Baudrate	The BPS (bits per second) communications rate use for this protocol, any value from 300 to 38400 is valid.
	DataBits	The number of data bits per character. The options are 7 or 8.
	Parity	The parity bit type. The options are O, E or N, for odd, even or none respectively.
	StopBits	The number of stop bits trailing each character. The options are 1 or 2.
	InterPollDelay	The delay in milliseconds inserted between polls by the EZMods
	InterCharDelay	The delay in milliseconds inserted between characters by the EZMods
	NotRepsondingPollingRate	The delay in milliseconds inserted between polls to devices which are currently not responding.
	MaxNoResposes	The maximum number of sequential no responses from a device, before this device is flagged as not responding.
[PumpTypennn]	Name	The pump type name, where nnn can be from 001 to 999 and corresponds to the PumpTypeID column in the PumpTypes table of the EZDB data base.
	Protocol	The ProtocolID used to communicate with this pump.
	MaxHoses	The maximum number of hoses found on this type/model of pump.
	PriceLevels	The maximum number of price levels supported by this type/model of pump.
	ResponseTimeout	The maximum number of milliseconds this pump type can take to respond to a poll.
	PollingRate	The delay in milliseconds between polls to pumps of this type.
	IntercharTimeout	The maximum number of milliseconds between characters, while this pump is responding to a poll.

5.4. EZDB INI file

This INI file is used to configure the data base client service extension, most of these parameters are pre-configured and should not be changed, specifically all of the SQL used by this application to interface to a physical database.

This INI file contains the standard parameters listed above along with the specific parameters listed below.

Section	Parameter	Description
[Datasource]	name	The name of the ODBC connection used to connect to the EZDB database.
	user	The data base user name used to make the connection.
	password	The password used in conjunction with this user.
[SelectHoses]		The SQL statement used to load all of the hoses defined in the EZDB database.
[SelectHose]		The SQL statement used to load a specific hose defined in the EZDB database.
[SelectPumps]		The SQL statement used to load all of the pumps defined in the EZDB database.
[SelectPump]		The SQL statement used to load a specific pump defined in the EZDB database.
[SelectTanks]		The SQL statement used to load all of the tanks defined in the EZDB database.
[SelectTank]		The SQL statement used to load a specific tank defined in the EZDB database.
[UpdateTank]		The SQL statement used to update a specific tank in the EZDB data base with its current state.
[SelectGrades]		The SQL statement used to load all of the grades defined in the EZDB database.
[SelectGrade]		The SQL statement used to load a specific grade defined in the EZDB database.
[SelectAllGradePrices]		The SQL statement used to load prices for all of the grades defined in the EZDB database.
[SelectGradePrices]		The SQL statement used to load prices for a specific grade defined in

		the EZDB database.
[SelectHoseDels]		The SQL statement used to load all of the hose deliveries currently stored in the EZDB database.
[SelectHoseDel]		The SQL statement used to load a specific hose delivery stored in the EZDB database.
[SelectPorts]		The SQL statement used to load all of the ports defined in the EZDB database.
[SelectPort]		The SQL statement used to load a specific port defined in the EZDB database.
[SelectAttendants]		The SQL statement used to load all of the attendants defined in the EZDB database.
[SelectAttendant]		The SQL statement used to load a specific attendant defined in the EZDB database.
[LogEvent]		The SQL statement used to log and event to the EZDB database.
[LogDelivery]		The SQL statement used to log and hose delivery to the EZDB database.
[ClearDelivery]		The SQL statement used to clear a hose delivery in the EZDB database.
[StackDelivery]		The SQL statement used to stack a hose delivery in the EZDB database.
[MaxHoseDelID]		The SQL statement used to retrieve the maximum hose delivery ID currently in the EZDB database.
[ProcessHoseETotals]		The SQL statement used to process the electronic totals for the hose as returned by the pump.
[AttendantLogon]		The SQL statement used to log a forecourt attendant onto a specific pump.
[AttendantLogoff]		The SQL statement used to log a forecourt attendant off all pumps that he/she was logged onto.

5.5. EZBackOffice INI file

The EZBackOffice INI is used by the optional EZForecourt back office. Note this application will also use the EZClient INI file in order to connect to the server. As such it does not have a [Server] or a [log] section, the application specific configuration parameters are listed below.

Section	Parameter	Description
[DataSource]	DTSOURCE	The name of the machine that hosts the EZDB database, if this is hosted on the same machine then the value of <local> should be entered
[PumpsReport]	Prepay	A value of Y or N (Yes or No) to determine if the Prepay deliveries section of the pumps report is displayed and/or printed.
	Postpay	A value of Y or N (Yes or No) to determine if the Postpay deliveries section of the pumps report is displayed and/or printed.
	Preauth	A value of Y or N (Yes or No) to determine if the Preauth deliveries section of the pumps report is displayed and/or printed.
	Monitor	A value of Y or N (Yes or No) to determine if the Monitor deliveries section of the pumps report is displayed and/or printed.
	Offline	A value of Y or N (Yes or No) to determine if the Offline deliveries section of the pumps report is displayed and/or printed.
	TestDelivery	A value of Y or N (Yes or No) to determine if the Test deliveries section of the pumps report is displayed and/or printed.
	DriveOff	A value of Y or N (Yes or No) to determine if the Drive off deliveries section of the pumps report is displayed and/or printed.
	ElecTotals	A value of Y or N (Yes or No) to determine if the electronic totals section of the pumps report is displayed and/or printed.
	TheoTotals	A value of Y or N (Yes or No) to determine if the theoretical totals section of the pumps report is displayed and/or printed.
	MechTotals	A value of Y or N (Yes or No) to determine if the mechanical totals section of the pumps report is displayed and/or printed.
[TanksReport]	Theo	A value of Y or N (Yes or No) to determine if the theoretical totals section of the tanks report is displayed and/or printed.
	Gauge	A value of Y or N (Yes or No) to determine if the electronic tank gauge totals section of the tanks report is displayed and/or printed.
	Dip	A value of Y or N (Yes or No) to determine if the manual tank dip totals section of the tanks report is displayed and/or printed.

5.6. EZLoader INI file

The EZLoader INI file contains all of the configurable parameters for all of the EZLoader utility. This INI file contains all of the standard parameters listed above but has no specific parameters.

5.7. EZLicense INI file

The EZLicense INI file is used by various EZForecourt modules, it should contain a valid license key as issued by EZTech, this will determine which of the modules are licensed to run and until which date. This INI only contains one application specific configuration key.

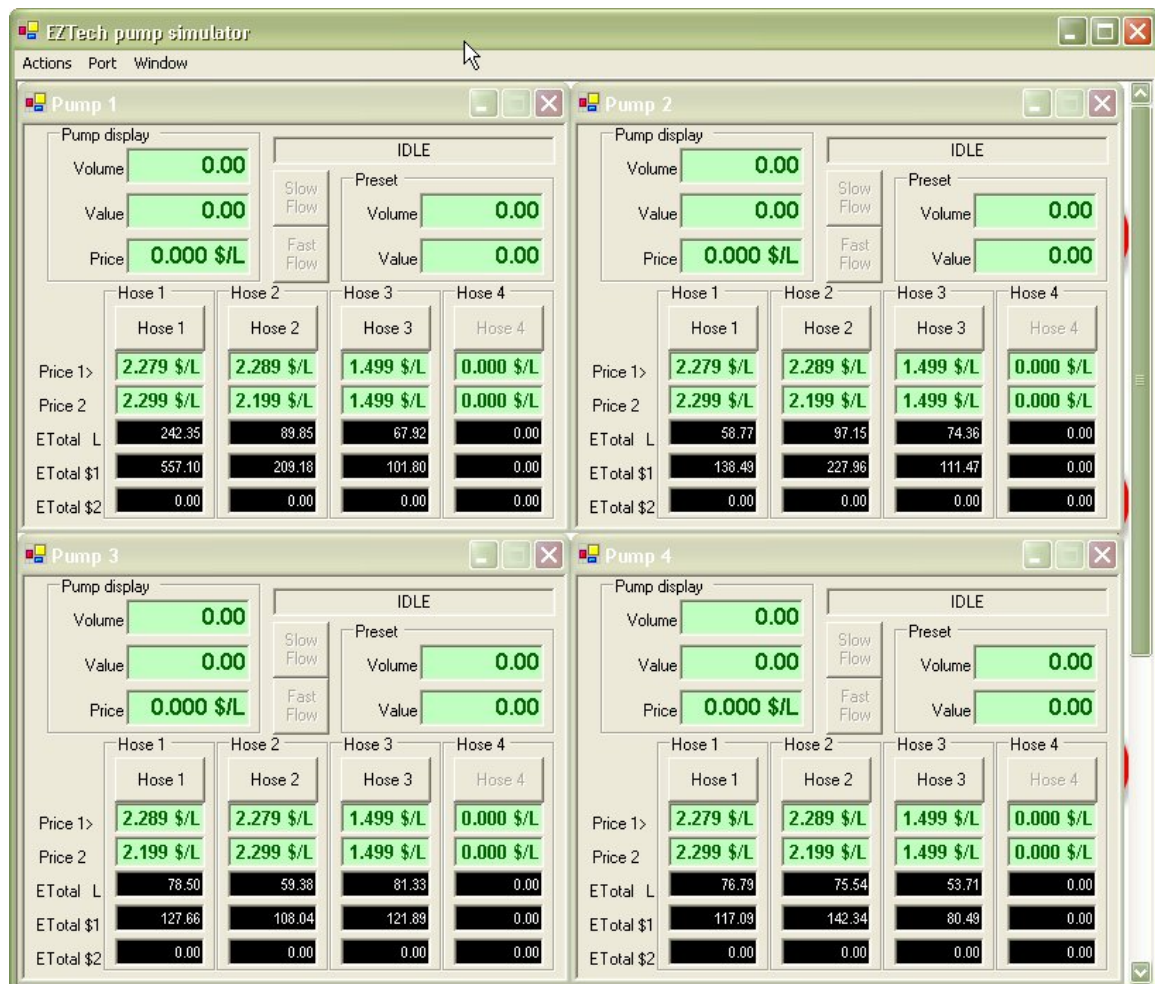
Section	Parameter	Description
[Application]	LicenseKey	A 24 hex digit license key as supplied by EZTech.

6. Developer's Kit contents

The developer's kit is shipped with a full version of EZForecourt (Server + BackOffice) as well as sample programs, pump simulator, USB RS485 adapter, and cable to connect the USB RS485 adapter to the EZMod. The pump simulator requires the USB RS485 adapter and cable in order to function. To install the USB RS485 adapter simply plug it into any available USB port and when the new hardware found dialog appears point it to the \FTDI directory on the install CD.

6.1. The EZSim pump simulator

The Pump simulator is used to assist with developing the host system, when a real pump is not available. It is a windows application that simulates a real pump. The EZSim pump appears as follows:



In order for the pump simulator to work the server must be configured to use the EZSim protocol for the EZMod in question and the pump types must all be set to EZSim.

To start a delivery simply click on any of the enabled hose, to end the delivery simply click the hose button again.

6.1.1. EZSim INI file

This EZSim INI file is used by the EZSim pump simulator, as this is not an EZClient it does not have a [Server] or [Log] section. A lot of the parameters for the EZSim application are updated by the application when it is exited; as such editing them is of little benefit. The application specific configuration parameters are as follows.

Section	Parameter	Description
[Parameters]	NumberOfPumps	The number of pumps currently configured for the simulator this can be from 1 to 16.
	PriceDecimals	The number of decimal places used for the price display and price fields in the communications protocol.
	ValueDecimals	The number of decimal places used for the value display and value fields in the communications protocol.
	VolumeDecimals	The number of decimal places used for the volume display and volume fields in the communications protocol.
	CommPort	The com port used by this simulator
	SlowRate	The amount added to the delivery volume every time a DelTimeInterval is passed when the pump is in slow flow mode.
	FastRate	The amount added to the delivery volume every time a DelTimeInterval is passed when the pump is in fast flow mode.
	DelTimeInterval	The amount of time in milliseconds that is used between updates of the volume delivered when the pump is delivering.
	PumpsPerRow	The number of pumps display per row in the EZSim window.
	HorizontalSpace	The number of pixels between each of the pump simulators horizontally.
	VerticalSpace	The number of pixels between each of the pump simulators vertically.
[PumpNNHoseY]	VolumeETot	The electronic volume total for pump NN hose Y
	Value1ETot	The electronic value total for pump NN hose Y price level 1
	Price1	The current price for price level 1
	Value2ETot	The electronic value total for pump NN hose Y price level 2
	Price2	The current price for price level 2

6.2. Sample applications

The developer's kit is shipped with three demonstration applications to assist in the integration of the EZForecourt into the host system. These applications have no copyrights applied to them and can be copied or modified as desired. EZTech are adding example applications all the time, if your requirements are not covered by the following samples, enter into contact with EZTech support.

6.2.1. EZDemoPos

The EZDemoPos is a VB.net application that acts as a POS terminal and demonstrates all of the functions that are possible with EZForecourt. It uses the EZTech.DLL .NET package.

6.2.2. EZClientVBTest

The EZClientVBTest is a more basic example and was also developed in VB.net; it demonstrates how to call the EZClient DLL directly.

6.2.3. EZDBVBClient

The EZDBVBClient application is an example developed in VB 6.0 and demonstrates how to develop a DB client; it uses the EZClient.OCX ActiveX control.

7. API reference

7.1. API Components

The API (application interface) for the EZForecourt product consists of five components:

- EZClient.DLL, a DLL which exposes the complete API to all types of applications. This DLL is not a COM DLL and as such does not require registering or require COM support capabilities of its host. The API for this is defined in the EZClient.h and EZClient.lib files included with the developer's kit.
- EZClient.OCX, an ActiveX control which exposes the complete API to those application types which can host ActiveX controls, this is typically restricted to applications with a user interface, and COM support.
- EZPump.OCX, a visual ActiveX control which exposes the pump specific parts of the API. It requires that the EZClient.OCX control also be hosted in the same application. This control is a visual control, and has a limited user interface of its own.
- EZTech.EZClient .NET control contained in the EZTech.DLL .NET package this control exposes the complete API to .NET applications such as VB.NET, C#, and C++ .NET with the managed C++ extensions.
- EZTech.EZPump a visual .NET control contained in the EZTech.DLL .NET package, this control exposes the pump specific parts of the API. It requires that the EZTech.EZClient control also be hosted in the same application. This control is a visual control, and has a limited user interface of its own.

The API calls are largely common across all five components and will be addressed just once in this document; any differences between the five interfaces will be highlighted.

7.2. Data types

The API to the EZForecourt was designed using data types that are supported across as many platforms and development environments as possible. The basic data types are as follows:-

long, Int32	32 bit Integer, range -2,147,483,648 to 2,147,483,647
short, Int16	16 bit Integer, range -32,768 to 32,767
DATE, DateTime	64 bit floating point value representing the number of days since January 1, 100. It has an approximate resolution of 1 millisecond up to December 31, 9999.
double, Double	64 bit standard floating point type
BSTR, String, String*	Variable length Unicode strings, i.e. 16 bit characters

The API is object based, with all objects having a unique identifier. These IDs are of type long and a value of -1 is designated as the NULL ID. The IDs are only unique to the specific object type; as such to identify a specific object the type and ID are required.

7.3. API Definition

7.3.1. ClientLogon/ClientLogonEx

Syntax

Visual Basic

```
Function ClientLogon( ByVal ClientID As Int32,
                    ByVal ClientType As Int16,
                    ByVal BSTR ServerName As String    **
                    ByVal short CallPortNo As Int16,   **
                    ByVal EventsPortNo As Int16,      **
                    ByVal CallTimeout As Int32,        **
                    ByVal EventHandle As Int32,        *
                    ByVal hWnd As System.IntPtr,       *
                    ByVal wParam As Int32) As Int32    *
```

C/C++/C#

```
Long ClientLogon( long ClientID ,
                  short ClientType ,
                  BSTR ServerName ,                **
                  short CallPortNo ,               **
                  short EventsPortNo ,             **
                  long CallTimeout,                 **
                  HANDLE Event ,                   *
                  HWND hWnd,                       *
                  long wParam) ;                   *
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

* EZClient.DLL only parameters

** ClientLogonEx only parameters

Parameters

Parameter	Type	Description
ClientID	long	A unique identifier to identify this client, this same value must be used by the EZClient.DLL in future calls to identify the source. This value must be between 1 and 99.
ClientType	short	Any combination of the following 0x01 = Calls client, i.e. can call all non-DB API functions 0x02 = Events client, i.e. will receive all non-DB events 0x04 = DB client, i.e. can call all DB API functions and will receive all DB related events, this identifies the client as the client who is performing all the DB operations on behalf of EZServer
ServerName	BSTR	The IP address or network name of the EZServer service host machine. This can be passed as '<local>' it is the on the same machine as the client.
CallPortNo	short	The IP Port number to use for calling the server, the default value



		for this is 5123. This value can be changed in the EZServer.ini file if required.
EventsPortNo	short	The IP Port number to use for receiving events from the, the default value for this is 5124. This value can be changed in the EZServer.ini file if required.
CallTimeout	long	The timeout in ms for the EZServer service to respond to calls. The default value for this is 10000.
EventHandle	HANDLE	A HANDLE to an event object that will be cleared when events have been received. Call ProcessEvents to retrieve the events for processing. Pass NULL or zero to disable this feature. (EZClient.DLL parameter only)
hWnd	HWND	A windows handle that will be posted with the wParam message when an event is received. Call ProcessEvents to retrieve the events for processing. Pass NULL or zero to disable this feature. (EZClient.DLL parameter only)
wMsg	long	The message type that is posted to the hWnd handle when events are received, it must be a value greater than 1024, this value is ignored if hWnd is passed as NULL.

Return value

Error code	Error description
OK_RESULT	The call was successful.
ALREADY_LOGGED_ON_RESULT	A client with this client ID is already logged on.
SERVER_TIMEOUT	The call to the server timed out, although a calls socket was opened successfully.
CONNECTION_BROKEN	A connection could not be established with the server, it is most likely not running or inaccessible.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This call is required to logon to the forecourt server (EZServer) before calling any other API function. The last three parameters are provided as means for the DLL to signal the host application that an event has occurred, either an event object handle (EventHandle) or window handle (hWnd) and message type (wMsg) are passed, never both. Calling ProcessEvents after receiving notification of the events will process the waiting events. None of these three parameters are required for the EZClient.OCX or EZTech.EZClient APIs as all of the event processing is handled internally. For ClientLogon the server address and port numbers are retrieved from the ezclient.ini file, for EZClientLogonEx these values are passed directly.

See also

ClientLogoff

7.3.2. ClientLogoff

Syntax

Visual Basic

Function ClientLogoff() As Int32

C/C++/C#

```
long ClientLogoff();
```

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is called to terminate a client session with the forecourt server (EZServer).

See also

ClientLogon/ClientLogonEx

7.3.3. ClientStatus

Syntax

Visual Basic

```
Function ClientStatus( ByRef PumpsReserved As Int16,  
ByRef DeliveriesLocked As Int16) As Int32
```

C/C++/C#

```
long ClientStatus( short* pPumpsReserved ,  
short* pDeliveriesLocked ) ;
```

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
PumpsReserved	Short output	The number of pumps currently reserved for a prepay or preauth operation by this client.
DeliveriesLocked	Short output	The number of deliveries locked by this client.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This call should be made just after logging on, or when the terminal enters its idle state (between transactions), to ensure that no previous operations have left dependencies in the forecourt server, for this client. If either of these values is returned as non-zero, it is up to the client application to locate the pump and/or delivery objects with reserves and/or locks, and take the appropriate action. As this call has no effect on the server it can also be used to verify the server connection.

See also

TestConnection, LicenseStatus

7.3.4. TestConnection

Syntax

Visual Basic

Function TestConnection () as Int32

C/C++/C#

long TestConnection () ;

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

Call this function to determine the status of the connection with the server.

See also

ClientStatus, LicenseStatus

7.3.5. LicenseStatus

Syntax

Visual Basic

Function LicenseStatus () as Int32

C/C++/C#

long LicenseStatus ();

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The license key is valid.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
SERVER_NOT_LICENSED_RESULT	The license key for the server is invalid or absent.
NO_EZMOD_RESULT	The EZModule for this license key cannot be found, it is most likely turned off or not plugged in.
LICENSE_EXPIRED_RESULT	The license key has expired.

Remarks

Call this function to determine the status of the license key for the server.

See also

ClientStatus, TestConnection

7.3.6. GetIniValue

Syntax

Visual Basic

Function GetIniValue(ByVal Section As String,
ByVal Key As String,
ByRef Value As String) As Int32

C/C++/C#

```
long GetIniValue ( BSTR Section ,  
BSTR Key,  
BSTR* pValue ) ;
```

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
Section	BSTR	The section in the EZServer.ini file where the key is located.
Key	BSTR	The key for the value being returned.
Value	BSTR	The returned value.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	The key or section names are invalid (too short).

Remarks

This call is provided so that clients can read the EZServer.ini configuration values currently in use by EZServer.

See also

SetIniValue

7.3.7. SetIniValue

Syntax

Visual Basic

```
Function SetIniValue( ByVal Section As String,  
                    ByVal Key As String,  
                    ByVal Value As String) As Int32
```

C/C++/C#

```
long SetIniValue ( BSTR Section ,  
                  BSTR Key,  
                  BSTR Value ) ;
```

Availability – EZClient.DLL, EZClient.OCX, EZTech.EZClient

Parameters

Parameter	Type	Description
Section	BSTR	The section in the EZServer.ini file where the key/value is to be added / updated.
Key	BSTR	The key for the value being updated/added.
Value	BSTR	The value to be updated/added in this section and key.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	The key or section names are invalid (too short).

Remarks

This call is provided so that clients can update the EZServer.ini configuration file. Once the update is completed the EZServer.ini values are re-loaded by the EZServer, hence alterations have immediate affect. Note that there is no validation of the section, key or values passed other than length. Passing non-existent section or key values will result in the section/key being added. Care must be taken to ensure that the configuration value does not have undesired effects on the server operation.

See also

GetIniValue

7.3.8. ProcessEvents

Syntax

Visual Basic

Function ProcessEvents() As Int32

C/C++/C#

long ProcessEvents();

Availability – EZClient.DLL for Events/DB clients only

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
CONNECTION_BROKEN	The connection with the server was lost.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call. It must include 'Events Client' and/or 'DB Client' before it will receive and events.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

ProcessEvents should be called when the EZClient.DLL posts a message to the windows handle, or when the associated event object is cleared. If neither a windows handle or and event object handle were passed to the ClientLogon, then this routine can be polled, however this is not recommended as it is inefficient.

The ProcessEvents call processes all of the waiting events and places them in an internal event queue. The contents of this queue can be examined and manipulated by using the following calls GetEventsCount, GetNextEventType, DiscardNextEvent, and GetNext????Event, where ???? is any of the various event types .

This call is only available on the EZClient.DLL because the EZClient.OCX and EZTech.EZClient controls do all of the event processing internally, firing the various events as the appropriate ActiveX or .NET events.

See also

ClientLogon, GetEventsCount, GetNextEventType, DiscardNextEvent, GetNextPumpEvent

7.3.9. GetEventsCount

Syntax

Visual Basic

Function GetEventsCount (ByRef Count As Int32) As Int32

C/C++/C#

long GetEventsCount(long* pCount);

Availability – EZClient.DLL for Events/DB clients only

Parameters

Parameter	Type	Description
Count	long output	The number of events waiting in the events queue.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.

Remarks

This call is used to determine how many events are waiting in the internal events queue.

See also

ProcessEvents, GetNextEventType, DiscardNextEvent

7.3.10. GetNextEventType

Syntax

Visual Basic

Function GetNextEventType (ByRef Type As Int16) As Int32

C/C++/C#

long GetNextEventType(short* pType) ;

Availability – EZClient.DLL for Events/DB clients only

Parameters

Parameter	Type	Description
Type	Short	The type of the event waiting at the head of the internal events queue. See Appendix N – Client event types for more information.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.

Remarks

This call is used to determine what the type of the event at the head of the internal events queue is. Once you have the type you can determine which of the GetNextEvent calls to use.

See also

ProcessEvents, Appendix N – Client event types

7.3.11. DiscardNextEvent

Syntax

Visual Basic

Function DiscardNextEvent() As Int32

C/C++/C#

long DiscardNextEvent() ;

Availability – EZClient.DLL for Events/DB clients only

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
OBJECT_DOES_NOT_EXIST_RESULT	The internal events queue is empty.

Remarks

This call removes the event from the head of the internal events queue; it is provided so that those events of types that the host application is not interested in can be discarded.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.12. GetNextPumpEvent / PumpStatusEvent / StatusEvent

Syntax

Visual Basic

```
Function GetNextPumpEvent ( ByRef PumpID As Int32, *
                          ByRef PumpNumber As Int16, *
                          ByRef State As Int16,
                          ByRef ReservedFor As Int16,
                          ByRef ReservedBy As Int32,
                          ByRef HoseID As Int32,
                          ByRef HoseNumber As Int16,
                          ByRef GradeID As Int32,
                          ByRef GradeName As String,
                          ByRef ShortGradeName As String,
                          ByRef PriceLevel As Int16,
                          ByRef Price As Double,
                          ByRef Volume As Double,
                          ByRef Value As Double,
                          ByRef StackSize As Int16 ) As Int32
```

C/C++/C#

```
long GetNextPumpEvent( long* pPumpID, *
                       short* pPumpNumber, *
                       short* pState,
                       short* pReservedFor,
                       long* pReservedBy,
                       long* pHoseID ,
                       short* pHoseNumber,
                       long* pGradeID,
                       BSTR* pGradeName ,
                       BSTR* pShortGradeName,
                       short* pPriceLevel ,
                       double* pPrice,
                       double* pVolume,
                       double* pValue,
                       short* pStackSize ) ;
```

Availability – EZClient.DLL for GetNextPumpEvent,
 EZClient.OCX and EZTech.EZClient for PumpStatusEvent with no return value,
 EZPump.OCX and EZTech.EZPump for StatusEvent with no return value, these
 values are also accessible via individual control properties.

* StatusEvent does not have these parameters

Parameters

Parameter	Type	Description
PumpID	long	The pump object identifier. This parameter is not present for the StatusEvent on the EZPump.OCX or EZTech.EZPump controls.
PumpNumber	short	The logical number of the pump. This parameter is not present

		for the StatusEvent on the EZPump.OCX or EZTech.EZPump controls.
State	short	The current state of the pump, see Appendix A – Pump states for explanation. EZPump.OCX, EZTech.EZPump control property PumpState().
ReservedFor	short	The reserve state of the pump, see Appendix B – Pump reserves for explanation. EZPump.OCX, EZTech.EZPump control property ReservedFor().
ReservedBy	long	The ID of the client that has the reserve in place or -1 if there is no reserve. EZPump.OCX, EZTech.EZPump control property ReservedBy().
HoseID	long	The ID of the currently pulled hose object or -1 if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurHoseID().
HoseNumber	short	The logical number of the currently pulled hose or 0 if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurHoseNumber().
GradeID	long	The ID of the grade for the currently pulled hose or -1 if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurGradeID().
GradeName	string	The grade name for the currently pulled hose or empty string if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurGradeName().
ShortGradeName	string	The short grade name for the currently pulled hose or empty string if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurShortGardeName().
PriceLevel	short	The current price level on the pump. EZPump.OCX, EZTech.EZPump control property CurPriceLevel().
Price	double	The selected price for the currently pulled hose or zero if there is no currently pulled hose. EZPump.OCX, EZTech.EZPump control property CurPrice().
Volume	double	The in progress volume total of the currently delivering pump, if it is not currently delivering then it is zero. EZPump.OCX, EZTech.EZPump control property CurVolume().
Value	double	The in progress value total of the currently delivering pump, if it is not currently delivering then it is zero.

		EZPump.OCX, EZTech.EZPump control property CurValue().
StackSize	short	The number of stacked deliveries waiting to be cleared on this pump.
		EZPump.OCX, EZTech.EZPump control property StackSize().

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired whenever the state of the pump changes, this could either be the pump state, delivery state, delivery values or the reserved state.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.13. GetNextDeliveryEvent[Ex] / DeliveryEvent[Ex]

Syntax

Visual Basic

```
Function GetNextDeliveryEvent(ByRef DeliveryID As Int32,  
    ByRef PumpID As Int32,  
    ByRef PumpNumber As Int16,  
    ByRef HoseID As Int32,  
    ByRef HoseNumber As Int16,  
    ByRef GradeID As Int32,  
    ByRef GradeName As String,  
    ByRef ShortGradeName As String,  
    ByRef PriceLevel As Int16,  
    ByRef Price As Double,  
    ByRef Volume As Double,  
    ByRef Value As Double,  
    ByRef DeliveryState As Int16,  
    ByRef DeliveryType As Int32,  
    ByRef LockedBy As Int32,  
    ByRef ReservedBy As Int32,  
    ByRef Age As Int32,  
    ByRef CompletedDT As Date,  
    ByRef AttendantID As Int32,  
    ByRef VolumeETot As Double, **  
    ByRef Volume2ETot As Double, **  
    ByRef ValueETot As Double ) As Int32 **
```

C/C++/C#

```
long GetNextDeliveryEvent( long* pDeliveryID ,  
    long* pPumpID, *  
    short* pPumpNumber, *  
    long* pHoseID,  
    short* pHoseNumber ,  
    long* pGradeID,  
    BSTR* pGradeName ,  
    BSTR* pShortGradeName,  
    short* pPriceLevel,  
    double* pPrice,  
    double* pVolume,  
    double* pValue,  
    short* pDeliveryState ,  
    short* pDeliveryType,  
    long* pLockedBy,  
    long* pReservedBy,  
    long* pAge,  
    DATE* pCompletedDT,  
    Long* pAttendantID ,  
    double* pVolumeETot , **  
    double* pVolume2ETot , **  
    double* pValueETot ) ; **
```



Availability – EZClient.DLL for GetNextDeliveryEvent, EZClient.OCX and EZTech.EZClient for DeliveryEvent with no return value, EZPump.OCX and EZTech.EZPump for DeliveryEvent with return value, *DeliveryEvent for EZPump.OCX and EZTech.EZPump does not have these parameters, for Events clients only.
 ** GetDeliveryEventEx and DeliveryEventEx parameters only.

Parameters

Parameter	Type	Description
DeliveryID	long	The delivery object identifier, this is allocated by the server when the delivery is completed.
PumpID	long	The pumps object identifier of the pump which dispensed this delivery. This parameter is not present for the StatusEvent on the EZPump.OCX or EZTech.EZPump controls.
PumpNumber	short	The logical pump number of the pump which dispensed this delivery. This parameter is not present for the StatusEvent on the EZPump.OCX or EZTech.EZPump controls.
HoseID	long	The hose object identifier for the hose which dispensed this delivery.
HoseNumber	short	The hose number of the hose which dispensed this delivery.
GradeID	long	The grades object identifier for the grade of this delivery.
GradeName	BSTR	The full name of the grade for this delivery.
ShortGradeName	BSTR	The short name of the grade for this delivery.
PriceLevel	short	The price level used for this delivery.
Price	double	The price used for this delivery.
Volume	double	The volume dispensed for this delivery.
Value	double	The total value for this delivery.
DeliveryState	short	The current state of this delivery, see Appendix D – Delivery states for explanation.
DeliveryType	short	The type of the delivery, see Appendix C – Delivery types for explanation
LockedBy	long	The ID of the client who has locked this delivery, if it is NULL ID then the delivery is unlocked.
ReservedBy	long	The ID of the client who has reserved this delivery, if it is NULL ID then this delivery was not reserved.
Age	long	The age of the delivery in seconds since it was completed.
CompletedDT	DATE	The date and time the delivery completed.
AttendantID	long	The ID of the Attendant that authorized this delivery or was logged onto the pump at the time the delivery was done. If neither of these was the case then this is NULL ID.
VolumeETot	double	The electronic volume total for this hose at the end of this delivery.
Volume2ETot	double	Always returned as zero. Reserved for future use.
ValueETot	double	The electronic value total for this hose at the end of this delivery.

Return value

Error code	Error description
------------	-------------------



OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired whenever the state of a delivery changes, that could be the delivery state, type, locked by, reserved by properties, or when the delivery is completed.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.14. GetNextServerEvent / ServerEvent

Syntax

Visual Basic

Function GetNextServerEvent (ByRef EventID As Int32,
ByRef EventText As String) As Int32

C/C++/C#

long GetNextServerEvent(long* pEventID,
BSTR* pEventText);

Availability – EZClient.DLL as GetNextServerEvent,
EZClient.OCX and EZTech.EZClient as ServerEvent (no return value)
for Events clients only

Parameters

Parameter	Type	Description
EventID	long	The ID of the server generated event.
EventText	BSTR	The event description string.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is generated by the server, it can be ignored or simply displayed on the screen.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.15. GetNextClientEvent/ClientEvent

Syntax

Visual Basic

```
Function GetNextClientEvent( ByRef ClientID As Int16,  
ByRef EventID As Int32,  
ByRef EventText As String) As Int32
```

C/C++/C#

```
long GetNextClientEvent( long* pClientID,  
long* pEventID,  
BSTR* pEventText );
```

Availability – EZClient.DLL for GetNextClientEvent,
EZClient.OCX and EZTech.EZClient for ClientEvent with no return value,
for Events/DB clients only

Parameters

Parameter	Type	Description
ClientID	long	The ID of the client which generated the event.
EventID	long	The ID of the client generated event.
EventText	BSTR	The event description string.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is generated when another EZServer client calls FireClientEvent. The value of the EventID and EventText are simply passed through.

See also

ProcessEvents, GetEventsCount, GetNextEventType, FireClientEvent

7.3.16. FireClientEvent

Syntax

Visual Basic

```
Function FireClientEvent( ByRef EventID As Int32,  
                          ByRef EventText As String) As Int32
```

C/C++/C#

```
long FireClientEvent( long* pEventID,  
                     BSTR* pEventText );
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
EventID	long	The ID of the client generated event
EventText	BSTR	The event description string.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

Calling this API will result in all other EZServer clients receiving a ClientEvent. This API is provided so that a client can send notifications to other clients.

See also

GetNextClientEvent/ClientEvent

7.3.17. GetNextDBLogEvent/DBLogEvent

Syntax

Visual Basic

```
Function GetNextDBLogEvent( ByRef EventType As Int32,  
ByRef DeviceID As Int32,  
ByRef EventDT As Date,  
ByRef EventText As String) As Int32
```

C/C++/C#

```
long GetNextDBLogEvent( long* pEventType ,  
long* pDeviceID ,  
DATE* pEventDT ,  
BSTR* pEventText ) ;
```

Availability – EZClient.DLL for GetNextDBLogEvent,
EZClient.OCX and EZTech.EZClient for DBLogEvent with no return value,
For DB clients only.

Parameters

Parameter	Type	Description
EventType	long	This is the type of the event being generated. See Appendix E – Event types, for more information
DeviceID	long	The ID of the device that generated this event, The event type will depend on the event type.
EventDT	DATE	The time and date the event occurred
EventText	BSTR	The event description text.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired every time the server or an attached device/module generates an exception event. As this is a DB client event, it is expected that this client will log the event to the database.



See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.18. GetNextDBLogDeliveryEvent/DBLogDeliveryEvent

Syntax

Visual Basic

Function GetNextDBLogDeliveryEvent (ByRef DeliveryID As Int32,
ByRef HoseID As Int32,
ByRef DeliveryState As Int16,
ByRef DeliveryType As Int16,
ByRef Volume As Double,
ByRef PriceLevel As Int16,
ByRef Price As Double,
ByRef Value As Double,
ByRef Volume2 As Double,
ByRef ReservedBy As Int32,
ByRef AttendantID As Int32,
ByRef DeliveryDT As Date) As Int32

C/C++/C#

```
long GetNextDBLogDeliveryEvent ( long* pDeliveryID,  
long* pHoseID,  
short* pDeliveryState,  
short* pDeliveryType,  
double* pVolume,  
short* pPriceLevel,  
double* pPrice,  
double* pValue,  
double* pVolume2,  
long* pReservedBy ,  
long* pAttendantID,  
DATE* pDeliveryDT ) ;
```

Availability – EZClient.DLL for GetNextDBLogDeliveryEvent,
EZClient.OCX and EZTech.EZClient for DBLogDeliveryEvent with no return
value.
For DB clients only.

Parameters

Parameter	Type	Description
DeliveryID	long	The ID of the delivery, as generated by the server.
HoseID	long	The ID of the hose which dispensed the delivery.
DeliveryState	short	The state of the delivery, see Appendix D – Delivery states for further explanation.
DeliveryType	short	The type of the delivery, see Appendix C – Delivery types for further explanation.
Volume	double	The volume of the delivery dispensed.
PriceLevel	short	The price level the delivery was done at.
Price	double	The price the delivery was done at.
Value	double	The total value of the delivery.

Volume2	double	The second volume of the delivery, this is a reserved field at this stage and is always zero.
ReservedBy	long	The ID of the client who reserved this delivery.
AttendantID	long	The ID of the attendant who authorized this delivery or NULL ID if not authorized by an attendant.
DeliveryDT	DATE	The date and time this delivery was stacked.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired every time a delivery is completed, as this is a DB client event, it is expected that the client will log this delivery to the database.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.19. GetNextDBCclearDeliveryEvent/DBCclearDeliveryEvent

Syntax

Visual Basic

Function GetNextDBCclearDeliveryEvent (ByRef DeliveryID As Int32,
ByRef DeliveryType As Int16,
ByRef ClearedBY As Int32,
ByRef ClearedDT As Date,
ByRef AttendantID As Int32) As Int32

C/C++/C#

```
long GetNextDBCclearDeliveryEvent( long* pDeliveryID,  
short* pDeliveryType,  
long* pClearedBY,  
DATE* pClearedDT,  
long* pAttendantID ) ;
```

Availability – EZClient.DLL for GetNextDBCclearDeliveryEvent,
EZClient.OCX and EZTech.EZClient for DBCclearDeliveryEvent with no return
value.
For DB clients only.

Parameters

Parameter	Type	Description
DeliveryID	long	The ID of the delivery that is being cleared
DeliveryType	short	The type of the delivery when it was cleared, see Appendix C – Delivery types for further information.
ClearedBY	long	The ID of the client who cleared the delivery.
ClearedDT	DATE	The time and date the delivery was cleared.
AttendantID	long	The ID of the attendant who cleared the delivery, or NULL ID if it was not cleared by an attendant.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks



This event is fired every time a delivery is cleared from the system. As this event is a DB client only event, the client is expected to flag the delivery as cleared in the database etc.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.20. GetNextDBStackDeliveryEvent/DBStackDeliveryEvent

Syntax

Visual Basic

Function GetNextDBStackDeliveryEvent (ByRef DeliveryID As Int32) As Int32

C/C++/C#

long GetNextDBStackDeliveryEvent(long* pDeliveryID) ;

Availability – EZClient.DLL for GetNextDBStackDeliveryEvent,
EZClient.OCX and EZTech.EZClient for DBStackDeliveryEvent with no return
value.
For DB clients only.

Parameters

Parameter	Type	Description
DeliveryID	long	The ID of the delivery that was stacked

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired when the state of a delivery changes from CURRENT to STACKED. As this is a DB client only event, it is expected that the DB client will update the state of the delivery in the database.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.21. GetNextDBHoseETotalsEvent/DBHoseETotalsEvent

Syntax

Visual Basic

Function GetNextDBHoseETotalsEvent (ByRef HoseID As Int32,
ByRef Volume As Double,
ByRef Value As Double,
ByRef VolumeETot As Double,
ByRef ValueETot As Double) As Int32

C/C++/C#

```
long GetNextDBHoseETotalsEvent( long* pHoseID ,  
double* pVolume ,  
double* pValue ,  
double* pVolumeETot ,  
double* pValueETot ) ;
```

Availability – EZClient.DLL for GetNextDBStackDeliveryEvent,
EZClient.OCX and EZTech.EZClient for DBStackDeliveryEvent with no return
value.
For DB clients only.

Parameters

Parameter	Type	Description
HoseID	long	The ID of the hose for which the electronics are being updated
Volume	double	The volume of the delivery which resulted in the change to the electronic totals, this value will be zero if the pump has just started responding.
Value	double	The value of the delivery which resulted in the change to the electronic totals, this value will be zero if the pump has just started responding.
VolumeETot	double	The new electronic volume total as returned by the pump for this hose.
ValueETot	double	The new electronic value total as returned by the pump for this hose.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact

	EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired at the completion of a delivery on the respective hose or when the pump starts responding. If the electronic totals are as a result of delivery completion, then the volume and value of the delivery is also passed, otherwise these values are zero.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.22. GetNextDBTriggerEvent/DBTriggerEvent

Syntax

Visual Basic

Function GetNextDBTriggerEvent (ByRef TableID As Int32,
ByRef RowID As Int32) As Int32

C/C++/C#

long GetNextDBTriggerEvent(long* pTableID ,
long* pRowID) ;

Availability – EZClient.DLL for GetNextDBTriggerEvent,
EZClient.OCX and EZTech.EZClient for DBTriggerEvent with no return value.
For DB clients only.

Parameters

Parameter	Type	Description
TableID	long	The ID of the table that has been updated
RowID	long	The object ID of the table row that has been updated

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired by the EZTrigger DLL which is called from various insert/update/delete triggers on the EZDB database. This is done so that the database client is aware of changes being made to the database by other applications, typically configuration/maintenance applications. For more information on this see the section regarding the EZTrigger DLL.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.23. GetNextDBAttendantLogonEvent / DBAttendantLogonEvent

Syntax

Visual Basic

Function GetNextDBAttendantLogon (ByRef ID As Int32,
ByRef PumpID As Int32) As Int32

C/C++/C#

long GetNextDBAttendantLogon (long* pID ,
long* pPumpID) ;

Availability – EZClient.DLL for GetDBNextAttendantLogon,
EZClient.OCX and EZTech.EZClient for DBAttendantLogon with no return value.
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant that has successfully logged on.
PumpID	long	The ID of the pump that was logged onto.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired by when an attendant successfully logs onto a specific pump.

See also

ProcessEvents, GetEventsCount, GetNextEventType, AttendantLogon

7.3.24. GetNextDBAttendantLogoffEvent / DBAttendantLogonEvent

Syntax

Visual Basic

Function GetNextDBAttendantLogoff (ByRef ID As Int32) As Int32

C/C++/C#

long GetNextDBAttendantLogoff (long* pID) ;

Availability – EZClient.DLL for GetDBNextAttendantLogoff, EZClient.OCX and EZTech.EZClient for DBAttendantLogoff with no return value, For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant that has successfully logged off.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired when an attendant successfully logs off one or more pumps.

See also

ProcessEvents, GetEventsCount, GetNextEventType, AttendantLogoff

7.3.25. GetNextDBTankStatusEvent / DBTankStatusEvent

Syntax

Visual Basic

```
Function GetNextDBTankStatusEvent ( ByRef TankID As Int32 ,  
ByRef GaugeVolume As double  
ByRef GaugeTCVolume As double  
ByRef GaugeUllage As double  
ByRef GaugeTemperature As double  
ByRef GaugeLevel As double  
ByRef GaugeWaterVolume As double  
ByRef GaugeWaterLevel As double ) As Int32
```

C/C++/C#

```
long GetNextDBTankStatusEvent ( long* pTankID ,  
double* pGaugeVolume ,  
double* pGaugeTCVolume ,  
double* pGaugeUllage ,  
double* pGaugeTemperature ,  
double* pGaugeLevel ,  
double* pGaugeWaterVolume ,  
double* pGaugeWaterLevel ) ;
```

Availability – EZClient.DLL for GetNextDBTankStatusEvent,
EZClient.OCX and EZTech.EZClient for DBTankStatusEvent with no return
value.
For DB clients only.

Parameters

Parameter	Type	Description
TankID	long	The ID of the tank that generated the status event.
GaugeVolume	double	The volume of fuel in the tank as reported by the tank gauge.
GaugeTCVolume	double	The temperature corrected volume of fuel in the tank as reported by the tank gauge.
GaugeUllage	double	The unused or available capacity in the tank as reported by the tank gauge.
GaugeTemperature	double	The temperature of the fuel in the tank as reported by the tank gauge.
GaugeLevel	double	The level of the fuel in the tank as reported by the tank gauge.
GaugeWaterVolume	double	The volume of water in the bottom of the tank as reported by the tank gauge.
GaugeWaterLevel	double	The level of the water in the bottom of the tank as reported by the tank gauge.

Return value



Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	Either the internal events queue is empty, or the event at the head of the queue is not of this type.

Remarks

This event is fired when an attendant successfully logs off one or more pumps.

See also

ProcessEvents, GetEventsCount, GetNextEventType

7.3.26. GetPumpsCount

Syntax

Visual Basic

Function GetPumpsCount (ByRef Count As Int32) As Int32

C/C++/C#

long GetPumpsCount(long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of pump objects configured in the server.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of pumps currently configured in the server. Once this is known, the IDs of the individual pump objects can be obtained using GetPumpByOrdinal.

See also

GetPumpByName, GetPumpBy, GetPumpByNumber

7.3.27. GetPumpByName

Syntax

Visual Basic

Function GetPumpByName (ByVal Name As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPumpByName(BSTR Name ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Name	BSTR	The name of the pump object for which the ID is being requested.
ID	long	The returned ID of the pump object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the ID of the pump from the pump name.

See also

GetPumpsCount, GetPumpBy, GetPumpByNumber

7.3.28. GetPumpByNumber

Syntax

Visual Basic

Function GetPumpByNumber (ByVal Number As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPumpByNumber(long Number ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Number	long	The logical number of the pump for which the ID is being requested.
ID	long	The returned ID of the pump object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the pump ID from the logical pump number.

See also

GetPumpsCount, GetPumpByOrdinal, GetPumpByName

7.3.29. GetPumpByOrdinal

Syntax

Visual Basic

Function GetPumpByOrdinal (ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPumpByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired pump object, this can be between 1 and the total number of pumps, as returned by GetPumpsCount.
ID	long	The returned ID of the pump.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the pump ID of a pump using it's ordinal value in the EZServer's internal pump objects list. The pumps are ordered in this list by ID.

See also

GetPumpsCount, GetPumpByNumber, GetPumpByName, GetPumpProperties

7.3.30. GetPumpProperties

Syntax

Visual Basic

```
Function GetPumpProperties ( ByVal ID As Int32,  
ByRef Number As Int32,  
ByRef Name As String,  
ByRef PhysicalNumber As Int16,  
ByRef Side As Int16,  
ByRef Address As Int16,  
ByRef PriceLevel1 As Int16,  
ByRef PriceLevel2 As Int16,  
ByRef PriceDspFormat As Int16,  
ByRef VolumeDspFormat As Int16,  
ByRef ValueDspFormat As Int16,  
ByRef Type As Int32,  
ByRef PortID As Int32,  
ByRef AttendantID As Int32,  
ByRef AuthMode As Int16,  
ByRef StackMode As Int16,  
ByRef PrepayAllowed As Int16,  
ByRef PreauthAllowed As Int16) As Int32
```

C/C++/C#

```
long GetPumpProperties( long ID ,  
long* pNumber,  
BSTR* pName,  
short* pPhysicalNumber,  
short* pSide,  
short* pAddress,  
short* pPriceLevel1,  
short* pPriceLevel2,  
short* pPriceDspFormat,  
short* pVolumeDspFormat,  
short* pValueDspFormat,  
short* pType,  
long* pPortID,  
long* pAttendantID,  
short* pAuthMode,  
short* pStackMode,  
short* pPrepayAllowed,  
short* pPreauthAllowed ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient,
EZPump.OCX and EZTech.EZPump via individual properties.

Parameters

Parameter	Type	Description
-----------	------	-------------

ID	long	The ID of the pump object for which the properties are being requested. EZPump.OCX and EZTech.EZPump property ID().
Number	long	The logical pump number EZPump.OCX and EZTech.EZPump property Number()
Name	BSTR	The pump name. EZPump.OCX and EZTech.EZPump property Name()
PhysicalNumber	short	The pump physical number. EZPump.OCX and EZTech.EZPump property PhysicalNumber()
Side	short	The side of the physical pump for this filling position. EZPump.OCX and EZTech.EZPump property Side()
Address	short	The electronic polling address, for the pump as configured in the pump. EZPump.OCX and EZTech.EZPump property Address()
PriceLevel1	short	The EZForecourt price level to be used as price level 1 on this pump. EZPump.OCX and EZTech.EZPump property PriceLevel1()
PriceLevel2	short	The EZForecourt price level to be used as price level 2 on this pump. EZPump.OCX and EZTech.EZPump property PriceLevel2()
PriceDspFormat	short	The format of the pump price display, see Appendix F – Pump display formats for more information. EZPump.OCX and EZTech.EZPump property PriceDspFormat()
VolumeDspFormat	short	The format of the pump volume display, see Appendix F – Pump display formats for more information. EZPump.OCX and EZTech.EZPump property VolumeDspFormat()
ValueDspFormat	short	The format of the pump value display, see Appendix F – Pump display formats for more information. EZPump.OCX and EZTech.EZPump property ValueDspFormat()
Type	short	The pump type identifier, this list is continually being added to. Consult EZTech support for the current list of supported pump types. EZPump.OCX and EZTech.EZPump property Type()
PortID	long	The ID of the communications port that this pump is connected to, the address of this pump must be unique for this port.

		EZPump.OCX and EZTech.EZPump property PortID()
AttendantID	long	The ID of the attendant currently logged onto this pump. EZPump.OCX and EZTech.EZPump property AttendantID()
AuthMode	short	The authorization mode of this pump. See Appendix G – Pump authorization modes for more information. EZPump.OCX and EZTech.EZPump property AuthMode()
StackMode	short	The pump delivery stack (or memory) mode for this pump, see Appendix H – Pump delivery stack (memory) modes for more information. EZPump.OCX and EZTech.EZPump property StackMode()
PrepayAllowed	short	Are prepay deliveries permitted on this pump. EZPump.OCX and EZTech.EZPump property PrepayAllowed()
PreauthAllowed	short	Are preauth deliveries permitted on this pump. EZPump.OCX and EZTech.EZPump property PreauthAllowed()

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API returns all the property values of a pump object for the given ID.

See also

SetPumpProperties

7.3.31. SetPumpProperties

Syntax

Visual Basic

```
Function SetPumpProperties ( ByVal ID As Int32,  
    ByVal Number As Int32,  
    ByVal Name As String,  
    ByVal PhysicalNumber As Int16,  
    ByVal Side As Int16,  
    ByVal Address As Int16,  
    ByVal PriceLevel1 As Int16,  
    ByVal PriceLevel2 As Int16,  
    ByVal PriceDspFormat As Int16,  
    ByVal VolumeDspFormat As Int16,  
    ByVal ValueDspFormat As Int16,  
    ByVal Type As Int32,  
    ByVal Port As Int32,  
    ByVal AttendantID As Int32,  
    ByVal AuthMode As Int16,  
    ByVal StackMode As Int16,  
    ByVal PrepayAllowed As Int16,  
    ByVal PreauthAllowed As Int16) As Int32
```

C/C++/C#

```
long SetPumpProperties( long ID ,  
    long Number,  
    BSTR Name,  
    short PhysicalNumber,  
    short Side, short Address,  
    short PriceLevel1,  
    short PriceLevel2,  
    short PriceDspFormat,  
    short VolumeDspFormat,  
    short ValueDspFormat,  
    short Type,  
    long Port,  
    long AttendantID,  
    short AuthMode,  
    short StackMode,  
    short PrepayAllowed,  
    short PreauthAllowed ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID for the pump object being queried.

		This property corresponds to the Pumps.PumpID column in the EZDB data model.
Number	long	The logical pump number. This property corresponds to the Pumps.PumpNumber column in the EZDB data model.
Name	BSTR	The pump name. This property corresponds to the Pumps.PumpName column in the EZDB data model.
PhysicalNumber	short	The pump physical number. This property corresponds to the Pumps.PumpPhysicalNumber column in the EZDB data model.
Side	short	The side of the physical pump for this filling position. This property corresponds to the Pumps.PumpSide column in the EZDB data model.
Address	short	The electronic polling address for the pump, as configured in the pump. This property corresponds to the Pumps.PumpAddress column in the EZDB data model.
PriceLevel1	short	The EZForecourt price level to be used as price level 1 on this pump. This property corresponds to the Pumps.GradePriceLevel1 column in the EZDB data model.
PriceLevel2	short	The EZForecourt price level to be used as price level 2 on this pump. This property corresponds to the Pumps.GradePriceLevel2 column in the EZDB data model.
PriceDspFormat	short	The format of the pump price display, see Appendix F – Pump display formats for more information. This property corresponds to the Pumps.PumpPriceDspFormat column in the EZDB data model.
VolumeDspFormat	short	The format of the pump volume display, see Appendix F – Pump display formats for more information. This property corresponds to the Pumps.PumpVolumeDspFormat column in the EZDB data model.
ValueDspFormat	short	The format of the value display see Appendix F – Pump display formats for more information. This property corresponds to the Pumps.PumpValueDspFormat column in the EZDB data model.
Type	short	The pump type identifier, this list is continually being added to. Consult EZTech support for the current list of supported pump types.

		This property corresponds to the Pumps.PumpTypeID column in the EZDB data model.
Port	long	The ID of the communications port that this pump is connected to, the address of this pump must be unique for this port. This property corresponds to the Pumps.PortID column in the EZDB data model.
AttendantID	long	The ID of the attendant currently logged onto this pump. This property corresponds to the Pumps.AttendantID column in the EZDB data model.
AuthMode	short	The authorization mode of this pump. See Appendix G – Pump authorization modes for more information. This property corresponds to the PumpProfiles.PumpProfileAuthMode column in the EZDB data model.
StackMode	short	The pump delivery stack (or memory) mode for this pump, see Appendix H – Pump delivery stack (memory) modes for more information. This property corresponds to the PumpProfiles.PumpProfileStackMode column in the EZDB data model.
PrepayAllowed	short	Are prepay deliveries permitted on this pump. This property corresponds to the PumpProfiles.PumpProfilePrepay column in the EZDB data model.
PreauthAllowed	short	Are preauth deliveries permitted on this pump. This property corresponds to the PumpProfiles.PumpProfilePreauth column in the EZDB data model.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.

Remarks

This API is provided so that pump objects can be created and maintained on the server. If the given pump ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new pump object with this ID is created and its properties set to the values passed. It is the responsibility of the caller to ensure that the pump address is unique for this port, and that the pump number etc are unique.

See also

GetPumpProperties, DeletePump

7.3.32. DeletePump

Syntax

Visual Basic

Function DeletePump(ByVal ID As Int32) As Int32

C/C++/C#

long DeletePump (long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient.
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump object to be deleted from the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove a pump object from the EZServer. If this pump has hoses or deliveries linked to it, it cannot be deleted, delete these objects first. If a pump ID of -1 is passed all the pumps will be deleted in a single operation.

See also

SetPumpProperties

7.3.33. GetPumpHosesCount/GetHosesCount

Syntax

Visual Basic

Function GetPumpHosesCount (ByVal ID As Int32, *
ByRef Count As Int32) As Int32

C/C++/C#

long GetPumpHosesCount(long ID , *
long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient for GetPumpHosesCount
EZPump.OCX and EZTech.EZPump for GetHosesCount

* EZClient.DLL, EZClient.OCX and EZTech.EZClient parameters only.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump in question. This parameter is not present for the StatusEvent on the EZPump.OCX and EZTech.EZPump controls.
Count	long	The returned number of hoses.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to determine the quantity of hoses configured on this pump. Once this is obtained GetPumpHoseByNumber can be used get the IDs for each of the individual hoses.

See also

GetPumpHoseByNumber

7.3.34. GetPumpHoseByNumber/GetHoseByNumber

Syntax

Visual Basic

Function GetPumpHoseByNumber (ByVal ID As Int32, *
ByVal Number As Int32,
ByRef HoseID As Int32) As Int32

C/C++/C#

long GetPumpHoseByNumber(long ID , *
long Number ,
long* pHoseID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient for GetPumpHoseByNumber.
EZPump.OCX and EZTech.EZPump for GetHoseByNumber.

* EZClient.DLL and EZClient.OCX and EZTech.EZClient parameters only

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that owns the hose. This parameter is not present for the StatusEvent on the EZPump.OCX or EZTech.EZPump controls.
Number	long	The logical number of the hose, ranging from 1 to the hoses count for this pump.
HoseID	long	The returned ID of the hose.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to get the hose IDs for hoses connected to a specific pump. This API along with GetPumpHosesCount can be used to get all of the hose IDs for a specific pump.

See also

GetPumpHosesCount/GetHosesCount

7.3.35. GetPumpStatus

Syntax

Visual Basic

```
Function GetPumpStatus (
    ByVal ID As Int32,
    ByRef ReservedFor As Int16,
    ByRef ReservedBy As Int32,
    ByRef HoseID As Int32,
    ByRef HoseNumber As Int16,
    ByRef GradeID As Int32,
    ByRef GradeName As String,
    ByRef ShortGradeName As String,
    ByRef PriceLevel As Int16,
    ByRef Price As Double,
    ByRef Volume As Double,
    ByRef Value As Double,
    ByRef StackSize As Int16) As Int32
```

C/C++/C#

```
long GetPumpStatus(
    long ID ,
    short* pState,
    short* pReservedFor,
    long* pReservedBy,
    long* pHoseID,
    short* pHoseNumber ,
    long* pGradeID,
    BSTR* pGradeName ,
    BSTR* pShortGradeName,
    short* pPriceLevel ,
    double* pPrice ,
    double* pVolume ,
    double* pValue ,
    short* pStackSize ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump for which the status is being requested.
State	short	The state of the pump, see Appendix A – Pump states for more information.
ReservedFor	short	The type of operation that the pump has been reserved for. See Appendix B – Pump reserves for more information.
ReservedBy	long	The ID of the client who placed the reserve on the pump.
HoseID	long	The ID of the current hose on the pump, or NULL ID if all the hoses are stowed.
HoseNumber	short	The logical number of the current hose on the pump, or zero if all the hoses are stowed.

GradeID	long	The ID of the grade for the current hose on the pump, or NULL ID if all the hoses are stowed.
GradeName	BSTR	The name of the grade for the current hose on the pump, or empty string if all the hoses are stowed.
ShortGradeName	BSTR	The short name of the grade for the current hose on the pump, or empty string if all the hoses are stowed.
PriceLevel	short	The currently selected price level on the pump.
Price	double	The current price on the pump or zero if there is no hose pulled.
Volume	double	The current volume on the pump or zero if there is no hose pulled.
Value	double	The current value on the pump or zero if there is no hose pulled.
StackSize	short	The number of untaken deliveries waiting on this pump.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This call can be used to get the current status of the pump. This call returns the same information as that supplied by the PumpStatusEvent.

See also

GetNextPumpEvent / PumpStatusEvent / StatusEvent,
GetPumpDeliveryProperties[Ex]/GetDeliveryProperties, GetAllPumpStatuses

7.3.36. GetAllPumpStatuses

Syntax

Visual Basic

Function GetAllPumpStatuses (ByRef States As String,
 ByRef CurrentHoses As String,
 ByRef DeliveriesCount As String) As Int32

C/C++/C#

```
long GetAllPumpStatuses ( BSTR* pStates ,
  BSTR* pCurrentHoses,
  BSTR* pDeliveriesCount ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient.

Parameters

Parameter	Type	Description
States	BSTR	A string containing all of the current pump states, position 1 contains the state for pump number 1 and position n contains the state for pump n, the length of the string corresponds to the highest pump number. To convert the string element to the actual pump state, subtract '0' (48 decimal, or 0x30 hex). See Appendix A – Pump states for more information.
CurrentHoses	BSTR	A string containing all of the current pump hoses, position 1 contains the current hose for pump number 1 and position n contains the current hose for pump n, the length of the string corresponds to the highest pump number. To convert the string element to the actual pump hose number, subtract '0' (48 decimal, or 0x30 hex).
DeliveriesCount	BSTR	A string containing a count of all of the available deliveries, position 1 contains the available deliveries for pump number 1 and position n contains the available deliveries for pump n, the length of the string corresponds to the highest pump number. To convert the string element to the actual number of available deliveries, subtract '0' (48 decimal, or 0x30 hex).

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.



Remarks

This API is used to quickly obtain a pump status summary for all of the installed hoses.

See also

GetPumpStatus

7.3.37. GetPumpDeliveryProperties[Ex]/GetDeliveryProperties[Ex]

Syntax

Visual Basic

```
Function GetPumpDeliveryProperties ( ByVal ID As Int32, *  
ByVal Index As Int16,  
ByRef DeliveryID As Int32,  
ByRef Type As Int16,  
ByRef State As Int16,  
ByRef HoseID As Int32,  
ByRef HoseNum As Int16,  
ByRef GradeID As Int32,  
ByRef GradeName As String,  
ByRef ShortGradeName As String,  
ByRef PriceLevel As Int16,  
ByRef Price As Double,  
ByRef Volume As Double,  
ByRef Value As Double,  
ByRef LockedBy As Int32,  
ByRef ReservedBy As Int32,  
ByRef Age As Int32,  
ByRef CompletedDT As Date,  
ByRef AttendantID As Int32,  
ByRef VolumeETot As Double, **  
ByRef Volume2ETot As Double, **  
ByRef ValueETot As Double ) As Int32 **
```

C/C++/C#

```
long GetPumpDeliveryProperties( long ID , *  
short Index ,  
long* pDeliveryID ,  
short* pType ,  
short* pState,  
long* pHoseID,  
short* pHoseNum,  
long* pGradeID ,  
BSTR* pGradeName ,  
BSTR* pShortGradeName,  
short* pPriceLevel,  
double* pPrice ,  
double* pVolume ,  
double* pValue ,  
long* pLockedBy ,  
long* pReservedBy ,  
long* pAge,  
DATE* pCompletedDT,  
Long* pAttendantID,  
double* pVolumeETot , **  
double* pVolume2ETot , **  
double* pValueETot ) ; **
```

Availability – EZClient.OCX and EZTech.EZClient for GetPumpDeliveryProperties
 EZPump.OCX and EZTech.EZPump for GetDeliveryProperties
 * EZClient.OCX parameters only
 ** GetPumpDeliveryPropertiesEx and GetDeliveryPropertiesEx parameters only

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that made the delivery. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.
Index	short	The index into the delivery stack on the pump. This can be from one to the size of the stack, see stack size parameter in GetNextPumpEvent / PumpStatusEvent / StatusEvent.
DeliveryID	long	The returned ID of the delivery.
Type	short	The type of the delivery, see Appendix C – Delivery types for more information.
State	short	The current state of the delivery, see Appendix D – Delivery states for more information.
HoseID	long	The ID of the hose which made the delivery.
HoseNum	short	The logical number of the hose which made the delivery.
GradeID	long	The ID of the grade for this delivery.
GradeName	BSTR	The name of the grade for this delivery.
ShortGradeName	BSTR	The short name of the grade for this delivery.
PriceLevel	short	The price level this delivery was taken at.
Price	double	The unit price that this delivery was taken at.
Volume	double	The total volume of this delivery.
Value	double	The total value of this delivery.
LockedBy	long	The ID of the client that has this delivery locked or NULL ID if it is unlocked.
ReservedBy	long	The ID of the client who reserved the pump for this delivery, or NULL ID if it was not reserved.
Age	long	The time in seconds since this delivery was completed.
CompletedDT	DATE	The date and time the delivery was completed.
AttendantID	long	The ID of the Attendant that authorized this delivery or was logged onto the pump at the time the delivery was done. If neither of these was the case then this is NULL ID.
VolumeETot	double	The electronic volume total for this hose at the end of this delivery.
Volume2ETot	double	Always returned as zero. Reserved for future use.
ValueETot	double	The electronic value total for this hose at the end of this delivery.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.



CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
NO_DELIVERY_AVAILABLE_RESULT	The delivery requested does exist.

Remarks

This API can be used to get all the details about a specific delivery by using the pump ID and position in the delivery stack.

See also

GetPumpStatus

7.3.38. PrepayReserve

Syntax

Visual Basic

Function PrepayReserve (ByVal ID As Int32) As Int32 *

C/C++/C#

long PrepayReserve(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump for which the reserve is being requested. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
INVALID_CLIENT_ID_RESULT	The current Client ID is not logged onto the server.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AVAILABLE_RESULT	The pump is unavailable, it is either changing prices or timing out after a prepay refund.
PREPAYS_NOT_PREMITTED_RESULT	Prepays are not permitted on this pump.
PUMP_IN_USE_RESULT	The pump is not idle.
PUMP_IS_STOPPED_RESULT	The pump is temp stopped.
PUMP_ALREADY_RESERVED_RESULT	The pump already has either a prepay or preauth reserve placed on it.
PUMP_NOT_AVAILABLE_RESULT	The pump has a current delivery which cannot be stacked automatically.

Remarks

The API is used to start a prepay delivery on a pump. This is the first step and must be done before the client is permitted to pay for the delivery. This guarantees that the pump is available and reserved for this prepay delivery.



See also

PrepayCancel, PrepayAuthorise

7.3.39. PrepayCancel

Syntax

Visual Basic

Function PrepayCancel (ByVal ID As Int32) As Int32 *

C/C++/C#

long PrepayCancel(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that is having the prepay reserve cancelled. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESERVED_FOR_PREPAY_RESULT	The pump was not reserved for a prepay delivery.
PUMP_NOT_RESERVED_BY_YOU_RESULT	The pump was not reserved by this client.
PUMP_NOT_RESPONDING_RESULT	The pump is not currently responding.
PUMP_IN_USE_RESULT	The pump is not idle.

Remarks

This API is used to cancel either a prepay reserve or prepay authorized pump. If the pump was already authorized for a prepay delivery then a prepay refund for the full amount will be generated.

See also



PrepayReserve, PrepayAuthorise

7.3.40. PrepayAuthorise

Syntax

Visual Basic

```
Function PrepayAuthorise ( ByVal ID As Int32,          *
                          ByVal LimitType As Int16,
                          ByVal Value As Double,
                          ByVal Hose As Int16,
                          ByVal PriceLevel As Int16) As Int32
```

C/C++/C#

```
long PrepayAuthorise( long ID ,                    *
                      short LimitType ,
                      double Value ,
                      short Hose ,
                      short PriceLevel ) ;
```

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
 EZTech.EZClient and EZTech.EZPump
 *EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that is being prepay authorized. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.
LimitType	short	The limit type being place on the delivery, DOLLAR_PREPAY_TYPE and VOLUME_PREPAY_TYPE are the only valid values here. See Appendix I – Pump limit types for more information
Value	double	The dollar or volume limit being placed on the delivery.
Hose	short	The permitted hose/hoses for this prepay delivery. See Appendix J – Permitted hoses mask for more information.
PriceLevel	short	The selected price level for this prepay delivery.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for

	assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
INVALID_PRESET_TYPE_RESULT	The preset type requested is not permitted for this type of authorize.
INVALID_PRESET_AMOUNT_RESULT	The preset amount is less than the configured minimum.
INVALID_HOSE_MASK_RESULT	None of the permitted hoses are configured on this pump.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_RESERVED_FOR_PREPAY_RESULT	The pump was not reserved for a prepay.
PUMP_NOT_RESERVED_BY_YOU_RESULT	The pump was not reserved by this client.
PUMP_IS_STOPPED_RESULT	The pump is currently temp stopped.

Remarks

This API is the second step in doing a prepay delivery on a pump. Once the pump has been reserved correctly, the value and grade have been confirmed and paid for, the pump can be authorized to delivery up to the volume/value purchased.

Once this delivery is completed it is cleared automatically as a prepay delivery. If the value/volume was less than the limit specified then a prepay refund is generated. If the delivery is not started within the configured prepay authorized timeout, a prepay refund will be generated for the full amount.

See also

PrepayReserve, PrepayCancel

7.3.41. PreauthReserve

Syntax

Visual Basic

Function PreauthReserve (ByVal ID As Int32) As Int32 *

C/C++/C#

long PreauthReserve(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
 EZTech.EZClient and EZTech.EZPump
 *EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump for which the reserve is being requested. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
INVALID_CLIENT_ID_RESULT	The current Client ID is not logged onto the server.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AVAILABLE_RESULT	The pump is unavailable, it is either changing prices or timing out after a prepay refund.
PREAUTHS_NOT_PREMITTED_RESULT	Preauths are not permitted on this pump.
PUMP_IN_USE_RESULT	The pump is not idle.
PUMP_IS_STOPPED_RESULT	The pump is temp stopped.
PUMP_ALREADY_RESERVED_RESULT	The pump already has either a prepay or preauth reserve on it.
PUMP_NOT_AVAILABLE_RESULT	The pump has a current delivery which cannot be stacked automatically.

Remarks

The API is used to start a preauth delivery on a pump. This is the first step and must be done before the client's credit rating is determined. This guarantees that the pump is available and reserved for this preauth delivery.



See also

PreauthCancel, PreauthAuthorise

7.3.42. PreauthCancel

Syntax

Visual Basic

Function PreauthCancel(ByVal ID As Int32) As Int32 *

C/C++/C#

long PreauthCancel(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that is having the preauth reserve cancelled. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESERVED_FOR_PREAUTH_RESULT	The pump was not reserved for a preauth.
PUMP_NOT_RESERVED_BY_YOU_RESULT	The pump was not reserved by this client.
PUMP_NOT_RESPONDING_RESULT	The pump is not currently responding.
PUMP_IN_USE_RESULT	The pump is not idle.

Remarks

This API is used to cancel either a preauth reserved or authorized pump.

See also

PreauthReserve, PreauthAuthorise

	assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
INVALID_PRESET_TYPE_RESULT	The preset type requested is not permitted for this type of authorize.
INVALID_PRESET_AMOUNT_RESULT	The preset amount is less than the configured minimum.
INVALID_HOSE_MASK_RESULT	None of the permitted hoses are configured on this pump.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_RESERVED_FOR_PREAUTH_RESULT	The pump was not reserved for a preauth.
PUMP_NOT_RESERVED_BY_YOU_RESULT	The pump was not reserved by this client.
PUMP_IS_STOPPED_RESULT	The pump is currently temp stopped.

Remarks

This API is the second step in doing a preauth delivery on a pump. Once the pump has been reserved correctly, the value and grade have been confirmed and the credit rating of the client confirmed, the pump can be authorized to delivery up to the volume/value limit.

Once this delivery is completed is must be cleared by the EZServer client who reserved/authorized this delivery. If the delivery is not started within the configured preauth authorized timeout the pump reverts back to unreserved. It is then the responsibility of the EZServer client to reverse the preauth transaction.

See also

PreauthReserve, PreauthCancel

7.3.44. LoadPreset

Syntax

Visual Basic

```
Function LoadPreset ( ByVal ID As Int32,          *
                    ByVal LimitType As Int16,
                    ByVal Value As Double,
                    ByVal Hose As Int16,
                    ByVal PriceLevel As Int16) As Int32
```

C/C++/C#

```
long LoadPreset( long ID ,          *
                 short LimitType ,
                 double Value ,
                 short Hose ,
                 short PriceLevel ) ;
```

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
 EZTech.EZClient and EZTech.EZPump
 *EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump that is being authorized with a preset. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.
LimitType	short	The limit type being place on the delivery, DOLLAR_PRESET_TYPE and VOLUME_PRESET_TYPE are the only valid values here. See Appendix I – Pump limit types for more information
Value	double	The dollar or volume limit being placed on the delivery.
Hose	short	The permitted hose/hoses for this delivery. See Appendix J – Permitted hoses mask for more information.
PriceLevel	short	The selected price level for this delivery.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

INVALID_PRESET_TYPE_RESULT	The preset type requested is not permitted for this type of authorize.
INVALID_PRESET_AMOUNT_RESULT	The preset amount is less than the configured minimum.
INVALID_HOSE_MASK_RESULT	None of the permitted hoses are configured on this pump.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AVAILABLE_RESULT	The pump is unavailable, it is either changing prices or timing out after a prepay refund.
PUMP_IN_USE_RESULT	The pump is not idle.
PUMP_CANNOT_BE_AUTHED_RESULT	The pump mode does not permit post pay deliveries.
PUMP_ALREADY_RESERVED_RESULT	The pump is already reserved for a prepay or preauth delivery.
PUMP_IS_STOPPED_RESULT	The pump is currently temp stopped.
HAS_CURRENT_DELIVERY_RESULT	The pump has a current delivery which cannot be automatically stacked.

Remarks

This API is used to manually authorize a pump with value limit, volume limit and/or hose mask. Once the delivery is completed it will need to be cleared as a Postpay delivery by any logged on EZServer client. If the delivery is not started within the configured manually authorized timeout the preset is lost and the pump reverts locked.

See also

CancelAuthorise

7.3.45. AttendantAuthorise

Syntax

Visual Basic

Function AttendantAuthorise (ByVal ID As Int32, *
ByVal AttendantID As Int32) As Int32

C/C++/C#

long AttendantAuthorise(long ID , *
long AttendantID) ;

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump being authorized. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.
AttendantID	long	The ID of the attendant authorizing the pump.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AVAILABLE_RESULT	The pump is unavailable, it is either changing prices or timing out after a prepay refund.
PUMP_IN_USE_RESULT	The pump is not idle.
PUMP_CANNOT_BE_AUTHED_RESULT	The pump mode does not permit post pay deliveries.
PUMP_ALREADY_RESERVED_RESULT	The pump is already reserved for a prepay or preauth delivery.
PUMP_IS_STOPPED_RESULT	The pump is currently temp stopped.
HAS_CURRENT_DELIVERY_RESULT	The pump has a current delivery which cannot be automatically stacked.

Remarks



This is API is used to manually authorize a pump without a limit, the resultant delivery is logged against the authorizing attendant. If the delivery is not started within the configured manually authorized timeout the pump reverts locked.

See also

CancelAuthorise

7.3.46. Authorise

Syntax

Visual Basic

Function Authorise(ByVal ID As Int32) As Int32 *

C/C++/C#

long Authorise(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump being authorized. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AVAILABLE_RESULT	The pump is unavailable, it is either changing prices or timing out after a prepay refund.
PUMP_IN_USE_RESULT	The pump is not idle.
PUMP_CANNOT_BE_AUTHED_RESULT	The pump mode does not permit post pay deliveries.
PUMP_ALREADY_RESERVED_RESULT	The pump is already reserved for a prepay or preauth delivery.
PUMP_IS_STOPPED_RESULT	The pump is currently temp stopped.
HAS_CURRENT_DELIVERY_RESULT	The pump has a current delivery which cannot be automatically stacked.

Remarks

This API is used to manually authorize a pump without a limit. If the delivery is not started within the configured manually authorized timeout the pump reverts locked.



See also

CancelAuthorise

7.3.47. CancelAuthorise

Syntax

Visual Basic

Function CancelAuthorise (ByVal ID As Int32) As Int32 *

C/C++/C#

long CancelAuthorise(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump having the manual authorization cancelled. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_AUTHED_RESULT	The pump was not manually authorized.
PUMP_ALREADY_RESERVED_RESULT	The pump has a prepay or preauth reserved placed on it.

Remarks

The API is provided to cancel any of the manual authorizations which can be done on a pump.

See also

LoadPreset, AttendantAuthorise, Authorise

7.3.48. TempStop

Syntax

Visual Basic

Function TempStop(ByVal ID As Int32) As Int32 *

C/C++/C#

long TempStop(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump being temp stopped. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_IS_STOPPED_RESULT	The pump is already temp stopped.

Remarks

This API is used to place a temp stop on a pump which is either delivering or authorized. A temp stop will stop the pump delivering or prevent it from starting. ReAuthorise is used to clear the temp stop.

See also

ReAuthorise

7.3.49. ReAuthorise

Syntax

Visual Basic

Function ReAuthorise(ByVal ID As Int32) As Int32 *

C/C++/C#

long ReAuthorise(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump being re-authorized. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_STOPPED_RESULT	The pump is not temp stopped.
DELIVERY_TERMINATED_RESULT	A delivery on the pump has been terminated, and hence the pump cannot be re-authorized.

Remarks

The API is used to re-start a pump which has been temp stopped.

See also

ReAuthorise

7.3.50. TerminateDelivery

Syntax

Visual Basic

Function TerminateDelivery(ByVal ID As Int32) As Int32 *

C/C++/C#

long TerminateDelivery(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
EZTech.EZClient and EZTech.EZPump
*EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump for which to terminate the delivery. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
PUMP_NOT_DELIVERING_RESULT	The pump is not currently delivering.

Remarks

This API is used to remotely terminate a delivery in progress. If the pump is currently delivering the pump is temp stopped first. The current value, volume, price and grade will be used to generate a delivery. Once the delivery has been terminated the pump cannot be authorized or re-authorized until the hose is returned. Once the hose is returned the electronic totals for the pump will be consulted and logged in the normal way.

See also

TempStop

7.3.51. StackCurrentDelivery

Syntax

Visual Basic

Function StackCurrentDelivery (ByVal ID As Int32) As Int32 *

C/C++/C#

long StackCurrentDelivery(long ID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX,
 EZTech.EZClient and EZTech.EZPump
 *EZClient.DLL, EZClient.OCX and EZTech.EZClient only parameter.

Parameters

Parameter	Type	Description
ID	long	The ID of the pump for which to stack the current delivery. This parameter is not present for the EZPump.OCX and EZTech.EZPump controls.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
STACK_IS_DISABLED_RESULT	Stacking deliveries is not permitted on this pump.
STACK_FULL_RESULT	The delivery stack for this pump is full.
NO_CURRENT_DELIVERY_RESULT	This pump has no current delivery.
DELIVERY_TYPE_CANNOT_BE_STACKED_RESULT	This type of delivery cannot be stacked.

Remarks

This API pushes the current delivery onto the delivery stack. The state of the delivery will change from current to stacked and a PumpStatusEvent and DeliveryEvent will be received if it was successfully stacked.

See also



GetNextPumpEvent / PumpStatusEvent / StatusEvent, GetNextDeliveryEvent[Ex] / DeliveryEvent

7.3.52. GetHosesCount

Syntax

Visual Basic

Function GetHosesCount (ByRef Count As Int32) As Int32

C/C++/C#

long GetHosesCount(long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of hoses configured in the server.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of hoses currently configured in the server. Once this is known, the IDs of the individual hose objects can be obtained using GetHoseByOrdinal.

See also

GetHoseByOrdinal

7.3.53. GetHoseByOrdinal

Syntax

Visual Basic

Function GetHoseByOrdinal(ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetHoseByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired hose object, this can be between 1 and the total number of hoses, as returned by Error! Reference source not found.
ID	long	The returned ID of the hose object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the hose ID of a hose using its ordinal value in the server's internal hose objects list. The hoses are ordered in this list by ID.

See also

GetHosesCount

7.3.54. GetHoseProperties

Syntax

Visual Basic

```
Function GetHoseProperties( ByVal ID As Int32,
  ByRef Number As Int32,
  ByRef PumpID As Int32,
  ByRef TankID As Int32,
  ByRef PhysicalNumber As Int32,
  ByRef MtrTheoValue As Double,
  ByRef MtrTheoVolume As Double,
  ByRef MtrElecValue As Double,
  ByRef MtrElecVolume As Double ) As Int32
```

C/C++/C#

```
long GetHoseProperties( long ID ,
  long* pNumber,
  long* pPumpID ,
  long* pTankID ,
  long* pPhysicalNumber,
  double* pMtrTheoValue ,
  double* pMtrTheoVolume ,
  double* pMtrElecValue ,
  double* pMtrElecVolume ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the hose object for which the properties are being requested.
Number	long	The logical number of the hose, this number is with respect to the pump, i.e. a number 1, means the first hose on this pump.
PumpID	long	The ID of the pump to which this hose is connected.
TankID	long	The ID of the tank to which this hose is connected.
PhysicalNumber	long	The physical number of the hose, this number is unique across all hoses and represents the public key for this object.
MtrTheoValue	double	The current theoretical total dollar value delivered on this hose, since it was first configured in the system.
MtrTheoVolume	double	The current theoretical total volume delivered on this hose, since it was first configured in the system.
MtrElecValue	double	The current electronic total dollar value delivered on this hose, as reported by the pump. This value is updated at the end of every delivery and when a pump starts responding.
MtrElecVolume	double	The current electronic total volume delivered on this hose as reported by the pump. This value is updated at the end of every delivery and when a pump starts responding.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API returns all the property values of a hose object for the given ID.

See also

SetHoseProperties

7.3.55. SetHoseProperties

Syntax

Visual Basic

```
Function SetHoseProperties( ByVal ID As Int32,
    ByVal Number As Int32,
    ByVal PumpID As Int32,
    ByVal TankID As Int32,
    ByVal PhysicalNumber As Int32,
    ByVal MtrTheoValue As Double,
    ByVal MtrTheoVolume As Double,
    ByVal MtrElecValue As Double,
    ByVal MtrElecVolume As Double) As Int32
```

C/C++/C#

```
long SetHoseProperties( long ID ,
    long Number,
    long PumpID,
    long TankID ,
    long PhysicalNumber,
    double MtrTheoValue ,
    double MtrTheoVolume ,
    double MtrElecValue ,
    double MtrElecVolume ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
 For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the hose object for which the properties set. This property corresponds to the Hoses.HoseID column in the EZDB data model.
Number	long	The logical number of the hose, this number is with respect to the pump, i.e. a number 1, means the first hose on this pump. This property corresponds to the Hoses.HoseNumber column in the EZDB data model.
PumpID	long	The ID of the pump to which this hose is connected. This property corresponds to the Hoses.PumpID column in the EZDB data model.
TankID	long	The ID of the tank to which this hose is connected. This property corresponds to the Hoses.TankID column in the EZDB data model.
PhysicalNumber	long	The physical number of the hose, this number is unique across

		<p>all hoses and represents the public key for this object.</p> <p>This property corresponds to the Hoses.HosePhysicalNumber column in the EZDB data model.</p>
MtrTheoValue	double	<p>The current theoretical total dollar value delivered on this hose, since it was first configured in the system.</p> <p>This property corresponds to the Hoses.HoseMtrTheoValue column in the EZDB data model.</p>
MtrTheoVolume	double	<p>The current theoretical total volume delivered on this hose, since it was first configured in the system.</p> <p>This property corresponds to the Hoses.HoseMtrTheoVolume column in the EZDB data model.</p>
MtrElecValue	double	<p>The current electronic total dollar value delivered on this hose, as reported by the pump. This value is updated at the end of every delivery and when a pump starts responding.</p> <p>This property corresponds to the Hoses.HoseMtrElecValue column in the EZDB data model.</p>
MtrElecVolume	double	<p>The current electronic total volume delivered on this hose as reported by the pump. This value is updated at the end of every delivery and when a pump starts responding.</p> <p>This property corresponds to the Hoses.HoseMtrElecVolume column in the EZDB data model.</p>

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.

Remarks

This API is provided so that hose objects can be created and maintained on the EZServer. If the given hose ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new hose object with this ID is created and its properties set to the values passed. It is the responsibility of the caller to ensure that the hose physical number is unique. The PumpID and TankID must represent objects which already exist on the server or the call will fail.

See also

GetHoseProperties, DeleteHose

7.3.56. DeleteHose

Syntax

Visual Basic

Function DeleteHose(ByVal ID As Int32) As Int32

C/C++/C#

long DeleteHose(long ID);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the hose object to be deleted from the server.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove a hose object from the server. If this hose has deliveries linked to it, it cannot be deleted, delete the delivery objects first. If a hose ID of -1 is passed all the hoses will be deleted in a single operation.

See also

SetHoseProperties

7.3.57. GetDeliveriesCount

Syntax

Visual Basic

Function GetDeliveriesCount(ByRef Count As Int32) As Int32

C/C++/C#

long GetDeliveriesCount(long* pCount);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of delivery objects currently in the server.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of deliveries currently in the server. Once this is known, the IDs of the individual delivery objects can be obtained using GetDeliveryByOrdinal.

See also

GetDeliveryByOrdinal

7.3.58. GetDeliveryByOrdinal

Syntax

Visual Basic

Function GetDeliveryByOrdinal (ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetDeliveryByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired delivery object, this can be between 1 and the total number of deliveries, as returned by GetDeliveriesCount.
ID	long	The returned ID of the hose object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the delivery ID of a delivery using its ordinal value in the server's internal deliveries objects list. The deliveries are ordered in this list by ID.

See also

GetDeliveriesCount

7.3.59. GetDeliveryProperties/GetDeliveryPropertiesEx

Syntax

Visual Basic

```
Function GetDeliveryProperties( ByVal ID As Int32,
                              ByRef HoseID As Int32,
                              ByRef State As Int16,
                              ByRef Type As Int16,
                              ByRef Volume As Double,
                              ByRef PriceLevel As Int16,
                              ByRef Price As Double,
                              ByRef Value As Double,
                              ByRef Volume2 As Double,
                              ByRef CompletedDT As Date,
                              ByRef LockedBy As Int32,
                              ByRef ReservedBy As Int32,
                              ByRef AttendantID As Int32,
                              ByRef Age As Int32,
                              ByRef VolumeETot As Double,      **
                              ByRef Volume2ETot As Double,     **
                              ByRef ValueETot As Double ) As Int32  **
```

C/C++/C#

```
long GetDeliveryProperties( long ID ,
                           long* pHoseID ,
                           short* pState ,
                           short* pType ,
                           double* pVolume ,
                           short* pPriceLevel ,
                           double* pPrice ,
                           double* pValue ,
                           double* pVolume2 ,
                           DATE* pCompletedDT ,
                           long* pLockedBy ,
                           long* pReservedBy ,
                           long* pAttendantID ,
                           long* pAge ,
                           double* pVolumeETot ,             **
                           double* pVolume2ETot ,            **
                           double* pValueETot ) ;             **
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
 ** GetDeliveryPropertiesEx parameters only

Parameters

Parameter	Type	Description
-----------	------	-------------

ID	long	The ID of the delivery object for which the properties are being requested.
HoseID	long	The ID of the hose which made the delivery.
State	short	The current state of the delivery, see Appendix D – Delivery states for more information.
Type	short	The type of the delivery, see Appendix C – Delivery types for more information.
Volume	double	The total volume for the delivery.
PriceLevel	short	The price level this delivery was taken at.
Price	double	The unit price for the delivery.
Value	double	The total value of this delivery.
Volume2	double	This property is reserved for future use and will be returned as zero.
CompletedDT	DATE	The date and time the delivery was completed.
ReservedBy	long	The ID of the client who reserved the pump for this delivery, or NULL ID if it was not reserved.
LockedBy	long	The ID of the client has locked this delivery or NULL ID if it was not reserved.
AttendantID	long	The ID of the Attendant that authorized this delivery or was logged onto the pump at the time the delivery was done. If neither of these was the case then this is NULL ID.
Age	long	The Age of the delivery in seconds since it was completed.
VolumeETot	double	The electronic volume total for this hose at the end of this delivery.
Volume2ETot	double	Always returned as zero. Reserved for future use.
ValueETot	double	The electronic value total for this hose at the end of this delivery.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API returns all the property values of a delivery object for the given ID.

See also

SetDeliveryProperties

7.3.60. SetDeliveryProperties/SetDeliveryPropertiesEx

Syntax

Visual Basic

```
Function SetDeliveryProperties( ByVal ID As Int32,  
                               ByVal HoselID As Int32,  
                               ByVal State As Int16,  
                               ByVal Type As Int16,  
                               ByVal Volume As Double,  
                               ByVal PriceLevel As Int16,  
                               ByVal Price As Double,  
                               ByVal Value As Double,  
                               ByVal Volume2 As Double,  
                               ByVal CompletedDT As Date,  
                               ByVal LockedBy As Int32,  
                               ByVal ReservedBy As Int32,  
                               ByVal AttendantID As Int32,  
                               ByVal Age As Int32 As Int32,  
                               ByVal VolumeETot As Double,      **  
                               ByVal Volume2ETot As Double,     **  
                               ByVal ValueETot As Double ) As Int32 **
```

C/C++/C#

```
long SetDeliveryProperties( long ID ,  
                            long HoselID ,  
                            short State ,  
                            short Type ,  
                            double Volume ,  
                            short PriceLevel ,  
                            double Price ,  
                            double Value ,  
                            double Volume2 ,  
                            DATE CompletedDT ,  
                            long LockedBy ,  
                            long ReservedBy ,  
                            long AttendantID ,  
                            long Age ,  
                            double VolumeETot ,                **  
                            double Volume2ETot ,              **  
                            double ValueETot ) ;                **
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

** SetDeliveryPropertiesEx parameters only

Parameters

Parameter	Type	Description
ID	long	The ID of the delivery object for which the properties are being set. This property corresponds to the HoseDeliveries.HoseDelID column in the EZDB data model.
HoseID	long	The ID of the hose which made the delivery. This property corresponds to the HoseDeliveries.HoseID column in the EZDB data model.
State	short	The current state of the delivery, see Appendix D – Delivery states for more information. This property corresponds to the HoseDeliveries. HoseDelState column in the EZDB data model.
Type	short	The type of the delivery, see Appendix C – Delivery types for more information. This property corresponds to the HoseDeliveries.HoseDelType column in the EZDB data model.
Volume	double	The total volume for the delivery. This property corresponds to the HoseDeliveries.HoseDelVolume column in the EZDB data model.
PriceLevel	short	The price level this delivery was taken at. This property corresponds to the HoseDeliveries.HoseDelPriceLevel column in the EZDB data model.
Price	double	The unit price for the delivery. This property corresponds to the HoseDeliveries. HoseDelPrice column in the EZDB data model.
Value	double	The total value of this delivery. This property corresponds to the HoseDeliveries. HoseDelValue column in the EZDB data model.
Volume2	double	This property is reserved for future use and will be returned as zero. This property corresponds to the HoseDeliveries.HoseDelVolume2 column in the EZDB data model.
CompletedDT	DATE	The date and time the delivery was completed. This property corresponds to the HoseDeliveries.HoseDelCompletedDT column in the EZDB data model.
LockedBy	long	The ID of the client who has locked this delivery or NULL ID if it was not reserved.
ReservedBy	long	The ID of the client who reserved the pump for this delivery, or NULL ID if it was not reserved.

		This property corresponds to the HoseDeliveries.HoseDelReservedBy column in the EZDB data model.
AttendantID	long	The ID of the Attendant that authorized this delivery or was logged onto the pump at the time the delivery was done. If neither of these was the case then this is NULL ID. This property corresponds to the HoseDeliveries.AttendantID column in the EZDB data model.
Age	long	The age in seconds of this delivery since it was completed.
VolumeETot	double	The electronic volume total for this hose at the end of this delivery
Volume2ETot	double	Always returned as zero. Reserved for future use.
ValueETot	double	The electronic value total for this hose at the end of this delivery.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.

Remarks

This API is provided so that delivery objects can be created and maintained on the server. If the given delivery ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new delivery object with this ID is created and its properties set to the values passed. The HoseID must represent an object which already exists on the server or the call will fail.

Normally when deliveries are completed the EZServer will automatically create a new delivery object with a generated ID. However if the EZServer has just been started up it is the responsibility of the DB Client to pre-load all the un-cleared deliveries which are still outstanding. Note that there is no DeleteDelivery API as ClearDelivery will automatically clear the deliveries from the system, and has the same effect.

See also

GetDeliveryProperties, SetNextDeliveryID

7.3.61. LockDelivery

Syntax

Visual Basic

Function LockDelivery(ByVal ID As Int32) As Int32

C/C++/C#

long LockDelivery(long ID);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the delivery that is being locked.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
DELIVERY_ALREADY_LOCKED_RESULT	The delivery has already been locked, by you or another client.
DELIVERY_IS_RESERVED_RESULT	The delivery was done as a prepay or preauth delivery and is reserved for another EZClient.
INVALID_CLIENT_ID_RESULT	The ID of this client is not a valid EZServer client.

Remarks

This API is used to lock a delivery; this will prevent other terminals from locking or clearing the delivery while this client has it included in a sale. Once the delivery is successfully locked it can then be cleared which will remove it from the EZServer.

See also

UnlockDelivery, ClearDelivery, LockAndClearDelivery

7.3.62. UnlockDelivery

Syntax

Visual Basic

Function UnlockDelivery(ByVal ID As Int32) As Int32

C/C++/C#

long UnlockDelivery(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the delivery that is being locked.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
DELIVERY_NOT_LOCKED_RESULT	The delivery is not currently locked.
DELIVERY_NOT_LOCKED_BY_YOU_RESULT	The delivery is locked by another EZServer client.

Remarks

The API performs the opposite of LockDelivery. It is provided so that a delivery already included into a sale (locked) can be voided (unlocked) making it available for other clients.

See also

LockDelivery, ClearDelivery, LockAndClearDelivery

7.3.63. ClearDelivery

Syntax

Visual Basic

Function ClearDelivery(ByVal ID As Int32,
ByVal Type As Int16) As Int32

C/C++/C#

long ClearDelivery(long ID ,
short Type) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the delivery being cleared.
Type	short	The type the delivery is being cleared as. See Appendix C – Delivery types for more information.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
DELIVERY_NOT_LOCKED_RESULT	The delivery is not currently locked.
DELIVERY_NOT_LOCKED_BY_YOU_RESULT	The delivery is locked by another EZServer client.
DELIVERY_CANNOT_BE_CLEARED_AS_THIS_TYPE_RESULT	The original type of the delivery does not permit it to be cleared as this type.

Remarks



This API is used after the delivery has been locked and the sale is finalized, this will result in the delivery being removed from the EZServer. If the original type of the delivery was POSTPAY it may be cleared as either MONITOR, TEST, DRIVEOFF or OFFLINE delivery types, otherwise it must be cleared with same type as its original delivery type.

See also

LockDelivery, UnlockDelivery, LockAndClearDelivery

7.3.64. LockAndClearDelivery

Syntax

Visual Basic

Function LockAndClearDelivery(ByVal ID As Int32,
ByVal Type As Int16) As Int32

C/C++/C#

long LockAndClearDelivery(long ID ,
short Type) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the delivery being locked and cleared.
Type	short	The type the delivery is being cleared as. See Appendix C – Delivery types for more information.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
DELIVERY_NOT_LOCKED_BY_YOU_RESULT	The delivery is locked by another EZServer client.
DELIVERY_CANNOT_BE_CLEARED_AS_THIS_TYPE_RESULT	The original type of the delivery does not permit it to be cleared as this type.

Remarks

This API is used to lock and clear a delivery as a single operation; this will result in the delivery being removed from the EZServer. If the original type of the delivery was POSTPAY it may be



cleared as either MONITOR, TEST, DRIVEOFF or OFFLINE delivery types, otherwise it must be cleared with same type as its original delivery type. This call is the equivalent to calling LockDelivery followed by a ClearDelivery. This call is intended to be used by host systems, which clear the deliveries as they are completed, passing the control/responsibility for the delivery to the host system.

See also

LockDelivery, UnlockDelivery, ClearDelivery

7.3.65. GetGradesCount

Syntax

Visual Basic

Function GetGradesCount(ByRef Count As Int32) As Int32

C/C++/C#

long GetGradesCount(long* pCount);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of grades configured in the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of grades currently configured in the EZServer. Once this is known, the IDs of the individual hose objects can be obtained using GetGradeByOrdinal.

See also

GetGradeByNumber, GetGradeByName, GetGradeByOrdinal, GetGradeProperties

7.3.66. GetGradeByNumber

Syntax

Visual Basic

```
Function GetGradeByNumber( ByVal Number As Int32,  
                          ByRef ID As Int32) As Int32
```

C/C++/C#

```
long GetGradeByNumber( long Number ,  
                      long* pID ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Number	long	The number of the grade for which the ID is being requested.
ID	long	The returned ID of the grade object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the grade ID from the grade number.

See also

GetGradesCount, GetGradeByName, GetGradeByOrdinal, GetGradeProperties

7.3.67. GetGradeByName

Syntax

Visual Basic

Function GetGradeByName(ByVal Name As String,
ByRef ID As Int32) As Int32

C/C++/C#

long GetGradeByName(BSTR Name ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Name	BSTR	The name of the grade object for which the ID is being requested.
ID	long	The returned ID of the grade object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the ID of the grade from the grade name.

See also

GetGradesCount, GetGradeByNumber, GetGradeByOrdinal, GetGradeProperties

7.3.68. GetGradeByOrdinal

Syntax

Visual Basic

Function GetGradeByOrdinal(ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetGradeByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired grade object, this can be between 1 and the total number of grades, as returned by GetGradesCount.
ID	long	The returned ID of the grade.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the grade ID of a grade using it's ordinal value in the EZServer's internal grade objects list. The grades are ordered in this list by ID.

See also

GetGradesCount, GetGradeByNumber, GetGradeByName, GetGradeProperties

7.3.69. GetGradeProperties

Syntax

Visual Basic

```
Function GetGradeProperties( ByVal ID As Int32,  
ByRef Number As Int32,  
ByRef Name As String,  
ByRef ShortName As String,  
ByRef Code As String) As Int32
```

C/C++/C#

```
long GetGradeProperties( long ID ,  
long* pNumber,  
BSTR* pName ,  
BSTR* pShortName ,  
BSTR* pCode ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the grade object for which the properties are being requested.
Number	long	The grade number.
Name	BSTR	The grade name.
ShortName	BSTR	The grade short name.
Code	BSTR	The grade code.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API returns all the property values of a grade object for the given ID.

See also

SetGradeProperties

7.3.70. SetGradeProperties

Syntax

Visual Basic

```
Function SetGradeProperties( ByVal ID As Int32,  
    ByVal Number As Int32,  
    ByVal Name As String,  
    ByVal ShortName As String,  
    ByVal Code As String ) As Int32
```

C/C++/C#

```
long SetGradeProperties( long ID ,  
    long Number,  
    BSTR Name ,  
    BSTR ShortName ,  
    BSTR Code );
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the grade object for which the properties are being requested. This property corresponds to the Grades.GradeID column in the EZDB data model.
Number	long	The grade number. . This property corresponds to the Grades. GradeNumber column in the EZDB data model.
Name	BSTR	The grade name. . This property corresponds to the Grades. GradeName column in the EZDB data model.
ShortName	BSTR	The grade short name. . This property corresponds to the Grades. GradeShortName column in the EZDB data model.
Code	BSTR	The grade code. . This property corresponds to the Grades.GradeCode column in the EZDB data model.

Return value

Error code	Error description
------------	-------------------

OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.

Remarks

This API is provided so that grade objects can be created and maintained on the server. If the given grade ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new grade object with this ID is created and its properties set to the values passed.

See also

GetGradeProperties, DeleteGrade

7.3.71. DeleteGrade

Syntax

Visual Basic

Function DeleteGrade(ByVal ID As Int32) As Int32

C/C++/C#

long DeleteGrade(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the grade object to be deleted from the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove a grade object from the EZServer. If this grade has tanks linked to it, it cannot be deleted, delete the tank objects first. This will also remove any prices associated with the grade.

See also

SetGradeProperties, SetGradePrice

7.3.72. SetGradePrice

Syntax

Visual Basic

Function SetGradePrice(ByVal ID As Int32,
 ByVal Level As Int16,
 ByVal Price As Double) As Int32

C/C++/C#

long SetGradePrice(long ID ,
 short Level ,
 double Price) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
 For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The grade ID of the grade having its price set. This property corresponds to the Grades.GradeID column in the EZDB data model.
Level	short	The price level, must be between 1 and 8. This property corresponds to the GradePrices.GradePriceLevel column in the EZDB data model.
Price	double	The price for the grade and price level. This property corresponds to the GradePrices.GradePrice column in the EZDB data model.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.

Remarks



This API is provided so that the, prices can be set/changed for each price level grade combination. To create the price level in the first place simply set a price for this level, if this grade did not have a price for this level, it will be created and added to the price list. To delete a price level, other than level 1, set the price to zero. Price level 1 cannot be deleted.

See also

GetGradePrice

7.3.73. GetGradePrice

Syntax

Visual Basic

```
Function GetGradePrice( ByVal ID As Int32,  
ByVal Level As Int16,  
ByRef Price As Double ) As Int32
```

C/C++/C#

```
long GetGradePrice( long ID ,  
short Level ,  
double* pPrice ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The grade ID of the grade having its price queried.
Level	short	The price level, must be between 1 and 8.
Price	double	The price for this grade ID and price level.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to read the price for a specific grade and price level.

See also

SetGradePrice

7.3.74. GetTanksCount

Syntax

Visual Basic

Function GetTanksCount(ByRef Count As Int32) As Int32

C/C++/C#

long GetTanksCount(long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of tanks configured in the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of tanks currently configured in the EZServer. Once this is known, the IDs of the individual tank objects can be obtained using GetTankByOrdinal.

See also

GetTankByNumber, GetTankByName, GetTankByOrdinal, GetTankProperties

7.3.75. GetTankByNumber

Syntax

Visual Basic

Function GetTankByNumber(ByVal Number As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetTankByNumber(long Number ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Number	long	The number of the grade for which the ID is being requested.
ID	long	The returned ID of the tank object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the tank ID from the tank number.

See also

GetTanksCount, GetTankByName, GetTankByOrdinal, GetTankProperties

7.3.76. GetTankByName

Syntax

Visual Basic

Function GetTankByName(ByVal Name As String,
ByRef ID As Int32) As Int32

C/C++/C#

long GetTankByName(BSTR Name ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Name	BSTR	The name of the tank object for which the ID is being requested.
ID	long	The returned ID of the tank object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the ID of the tank from the tank name.

See also

GetTanksCount, GetTankByNumber, GetTankByOrdinal, GetTankProperties

7.3.77. GetTankByOrdinal

Syntax

Visual Basic

Function GetTankByOrdinal(ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetTankByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired tank object, this can be between 1 and the total number of tanks, as returned by GetTanksCount.
ID	long	The returned ID of the tank.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the tank ID of a tank using it's ordinal value in the EZServer's internal tank objects list. The tanks are ordered in this list by ID.

See also

GetTanksCount, GetTankByNumber, GetTankByName, GetTankProperties

7.3.78. GetTankProperties

Syntax

Visual Basic

Function GetTankProperties(ByVal ID As Int32,
ByRef Number As Int32,
ByRef Name As String,
ByRef GradelID As Int32,
ByRef Type As Int16,
ByRef Capacity As Double,
ByRef Diameter As Double,
ByRef TheoVolume As Double,
ByRef GaugeVolume As Double,
ByRef GaugeTCVolume As Double,
ByRef GaugeUllage As Double,
ByRef GaugeTemperature As Double,
ByRef GaugeLevel As Double,
ByRef GaugeWaterVolume As Double,
ByRef GaugeWaterLevel As Double,
ByRef GaugeID As Int32,
ByRef ProbeNo as Int16) As Int32

C/C++/C#

long GetTankProperties(long ID ,
long* pNumber,
BSTR* pName ,
long* pGradelID ,
short* pType ,
double* pCapacity ,
double* pDiameter ,
double* pTheoVolume ,
double* pGaugeVolume ,
double* pGaugeTCVolume ,
double* pGaugeUllage ,
double* pGaugeTemperature ,
double* pGaugeLevel ,
double* pGaugeWaterVolume,
double* pGaugeWaterLevel,
long* pGaugeID ,
short* pProbeNo) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the tank object for which the properties are being requested.
Number	long	The tank number.

Name	BSTR	The tank name.
GradeID	long	The ID of the grade contained in this tank.
Type	short	The type of the tank. See Appendix K – Tank Types for more information.
Capacity	double	The total capacity of the tank in liters or gallons.
Diameter	double	The Diameter of the tank in meters or feet.
TheoVolume	double	The theoretical volume of fuel currently in the tank as calculated by the EZServer.
GaugeVolume	double	The volume in the tank as returned by the tank gauge.
GaugeTCVolume	double	The temperature corrected volume in the tank as returned by the tank gauge.
GaugeUllage	double	The unused or available capacity in the tank as returned by the tank gauge.
GaugeTemperature	double	The temperature of the fuel contained in the tank as reported by the tank gauge.
GaugeLevel	double	The level of the fuel in the tank as reported by the tank gauge.
GaugeWaterVolume	double	The volume of water in the bottom of the tank as reported by the tank gauge.
GaugeWaterLevel	double	The level of the water in the bottom of the tank as reported by the tank gauge.
GaugeID	long	The ID of the gauge which is monitoring this tank.
ProbeNo	short	The number of the probe as configured in the tank gauge.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API returns all the property values of a tank object for the given ID.

See also

SetTankProperties

7.3.79. SetTankProperties

Syntax

Visual Basic

Function SetTankProperties(ByVal ID As Int32,
ByVal Number As Int32,
ByVal Name As String,
ByVal GradeID As Int32,
ByVal Type As Int16,
ByVal Capacity As Double,
ByVal Diameter As Double,
ByVal TheoVolume As Double,
ByVal GaugeVolume As Double,
ByVal GaugeTCVolume As Double,
ByVal GaugeUllage As Double,
ByVal GaugeTemperature As Double,
ByVal GaugeLevel As Double,
ByVal GaugeWaterVolume As Double,
ByVal GaugeWaterLevel As Double,
ByVal GaugeID As Int32,
ByVal ProbeNo As Int16) As Int32

C/C++/C#

long SetTankProperties(long ID ,
long Number,
BSTR Name ,
long GradeID ,
short Type ,
double Capacity ,
double Diameter ,
double TheoVolume ,
double GaugeVolume ,
double GaugeTCVolume ,
double GaugeUllage ,
double GaugeTemperature ,
double GaugeLevel ,
double GaugeWaterVolume,
double GaugeWaterLevel,
long GaugeID,
short ProbeNo) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the tank object for which the properties are being set.

		This property corresponds to the Tanks.TankID column in the EZDB data model.
Number	long	The tank number. This property corresponds to the Tanks.TankNumber column in the EZDB data model.
Name	BSTR	The tank name. This property corresponds to the Tanks.TankName column in the EZDB data model.
GradeID	long	The ID of the grade contained in this tank. This property corresponds to the Tanks.GradeID column in the EZDB data model.
Type	short	The type of the tank. See Appendix K – Tank Types for more information. This property corresponds to the Tanks.TankType column in the EZDB data model.
Capacity	double	The total capacity of the tank in liters or gallons. This property corresponds to the Tanks.TankCapacity column in the EZDB data model.
Diameter	double	The Diameter of the tank in meters or feet. This property corresponds to the Tanks.TankDiameter column in the EZDB data model.
TheoVolume	double	The theoretical volume of fuel currently in the tank as calculated by the EZServer. This property corresponds to the Tanks.TankTheoVolume column in the EZDB data model.
GaugeVolume	double	The volume in the tank as returned by the tank gauge. This property corresponds to the Tanks. TankGaugeVolume column in the EZDB data model.
GaugeTCVolume	double	The temperature corrected volume in the tank as returned by the tank gauge. This property corresponds to the Tanks.TankGaugeTCVolume column in the EZDB data model.
GaugeUllage	double	The unused or available capacity in the tank as returned by the tank gauge. This property corresponds to the Tanks.TankGaugeUllage column in the EZDB data model.
GaugeTemperature	double	The temperature of the fuel contained in the tank as reported by the tank gauge. This property corresponds to the Tanks.TankGaugeTemperature column in the EZDB data

		model.
GaugeLevel	double	The level of the fuel in the tank as reported by the tank gauge. This property corresponds to the Tanks.TankGaugeLevel column in the EZDB data model.
GaugeWaterVolume	double	The volume of water in the bottom of the tank as reported by the tank gauge. This property corresponds to the Tanks.TankGaugeWaterVolume column in the EZDB data model.
GaugeWaterLevel	double	The level of the water in the bottom of the tank as reported by the tank gauge. This property corresponds to the Tanks.TankGaugeWaterLevel column in the EZDB data model.
GaugeID	long	The ID of the gauge which is monitoring this tank. This property corresponds to the Tanks.TankGaugeID column in the EZDB data model.
ProbeNo	short	The number of the probe as configured in the tank gauge. This property corresponds to the Tanks.TankProbeNo column in the EZDB data model.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.

Remarks

This API is provided so that tank objects can be created and maintained on the EZServer. If the given tank ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new tank object with this ID is created and its properties set to the values passed. The GradeID must represent a grade which already exists on the server or the call will fail.

See also

GetTankProperties, DeleteTank

7.3.80. DeleteTank

Syntax

Visual Basic

Function DeleteTank(ByVal ID As Int32) As Int32

C/C++/C#

long DeleteTank(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the tank object to be deleted from the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove a tank object from the EZServer. If this tank has hoses linked to it, it cannot be deleted, delete the hose objects first. If a tank ID of -1 is passed all the tanks will be deleted in a single operation.

See also

SetTankProperties

7.3.81. GetPortsCount

Syntax

Visual Basic

Function GetPortsCount(ByRef Count As Int32) As Int32

C/C++/C#

long GetPortsCount(long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of ports configured in the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of ports currently configured in the EZServer. Once this is known, the IDs of the individual port objects can be obtained using GetPortByOrdinal.

See also

GetPortByNumber, GetPortByName, GetPortByOrdinal, GetPortProperties

7.3.82. GetPortByNumber

Syntax

Visual Basic

Function GetPortByNumber(ByVal Number As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPortByNumber(long Number ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Number	long	The number of the port for which the ID is being requested.
ID	long	The returned ID of the port object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the port ID from the port number.

See also

GetPortsCount, GetPortByName, GetPortByOrdinal, GetPortProperties

7.3.83. GetPortByName

Syntax

Visual Basic

Function GetPortByName(ByVal Name As String,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPortByName(BSTR Name ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Name	BSTR	The name of the port object for which the ID is being requested.
ID	long	The returned ID of the port object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the ID of the port from the port name.

See also

GetPortsCount, GetPortByNumber, GetPortByOrdinal, GetPortProperties

7.3.84. GetPortByOrdinal

Syntax

Visual Basic

Function GetPortByOrdinal(ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetPortByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired port object, this can be between 1 and the total number of ports, as returned by GetPortsCount.
ID	long	The returned ID of the port object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the tank ID of a tank using it's ordinal value in the EZServer's internal tank objects list. The tanks are ordered in this list by ID.

See also

GetPortsCount, GetPortByNumber, GetPortByName, GetPortProperties

7.3.85. GetPortProperties

Syntax

Visual Basic

```
Function GetPortProperties( ByVal ID As Int32,  
ByRef Number As Int32,  
ByRef Name As String,  
ByRef ProtocolID As Int32,  
ByRef DeviceType As Int16,  
ByRef SerialNo As String) As Int32
```

C/C++/C#

```
long GetPortProperties( long ID ,  
long* pNumber,  
BSTR* pName ,  
long* pProtocolID ,  
short* pDeviceType ,  
BSTR* pSerialNo ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the port object for which the properties are being requested.
Number	long	The port number.
Name	BSTR	The port name.
ProtocolID	long	The ID of the protocol that is being used for this port. The list of supported protocols is being updated continually. Consult EZTech for an up-to-date list.
DeviceType	short	The device type for the devices connected to this port. See Appendix L – Device Types for more information.
SerialNo	BSTR	The serial number of the EZForecourt module that is connected to this port. If this is a standard serial port then this property is ignored.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.



Remarks

This API returns all the property values of a port object for the given ID.

See also

SetPortProperties

7.3.86. SetPortProperties

Syntax

Visual Basic

Function SetPortProperties(ByVal ID As Int32,
ByVal Number As Int32,
ByVal Name As String,
ByVal ProtocolID As Int32,
ByVal DeviceType As Int16,
ByVal SerialNo As String) As Int32

C/C++/C#

long SetPortProperties(long ID ,
long Number,
BSTR Name,
long ProtocolID ,
short DeviceType ,
BSTR SerialNo) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the port object for which the properties are being requested. This property corresponds to the Ports.PortID column in the EZDB data model.
Number	long	The port number. This property corresponds to the Ports.PortNumber column in the EZDB data model.
Name	BSTR	The port name. This property corresponds to the Ports.PortName column in the EZDB data model.
ProtocolID	long	The ID of the protocol that is being used for this port. The list of supported protocols is being updated continually. Consult EZTech for an up-to-date list. This property corresponds to the Ports.ProtocolID column in the EZDB data model.
DeviceType	short	The device type for the devices connected to this port. See Appendix L – Device Types for more information. This property corresponds to the Protocols.DeviceType column in the EZDB data model.

SerialNo	BSTR	The serial number of the EZForecourt module that is connected to this port. If this is a standard serial port then this property is ignored. This property corresponds to the Ports.SerialNo column in the EZDB data model.
----------	------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.

Remarks

This API is provided so that port objects can be created and maintained on the EZServer. If the given port ID already exists then the properties for this object will be overwritten with the values passed, otherwise a new port object with this ID is created and its properties set to the values passed. The ProtocolID must represent a protocol which is installed on the system or the call will fail.

See also

GetPortProperties, DeletePort

7.3.87. DeletePort

Syntax

Visual Basic

Function DeletePort(ByVal ID As Int32) As Int32

C/C++/C#

long DeletePort(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only. For EZTech.EZClient this API is called RemovePort.

Parameters

Parameter	Type	Description
ID	long	The ID of the port object to be deleted from the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove a port object from the EZServer. If this port has devices linked to it, it cannot be deleted, delete the device objects first. If a port ID of -1 is passed all the ports will be deleted in a single operation.

See also

SetPortProperties

7.3.88. GetAttendantsCount

Syntax

Visual Basic

Function GetAttendantsCount(ByRef Count As Int32) As Int32

C/C++/C#

long GetAttendantsCount(long* pCount) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Count	long	The returned total number of attendants configured in the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is used determine the total number of attendants currently configured in the EZServer. Once this is known, the IDs of the individual attendant objects can be obtained using GetAttendantByOrdinal.

See also

GetAttendantByOrdinal, GetAttendantByNumber, GetAttendantByName, GetAttendantProperties

7.3.89. GetAttendantByNumber

Syntax

Visual Basic

```
Function GetAttendantByNumber( ByVal Number As Int32,  
ByRef ID As Int32 ) As Int32
```

C/C++/C#

```
long GetAttendantByNumber( long Number ,  
long* pID ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Number	long	The number of the attendant for which the ID is being requested.
ID	long	The returned ID of the attendant object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the attendant ID from the attendant number.

See also

GetAttendantsCount, GetAttendantByName, GetAttendantByOrdinal, GetAttendantProperties

7.3.90. GetAttendantByName

Syntax

Visual Basic

Function GetAttendantByName(ByVal Name As String,
ByRef ID As Int32) As Int32

C/C++/C#

long GetAttendantByName(BSTR Name ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Name	BSTR	The name of the attendant object for which the ID is being requested.
ID	long	The returned ID of the attendant object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

Use this API to get the ID of the attendant from the attendant name.

See also

GetAttendantsCount, GetAttendantByNumber, GetAttendantByOrdinal, GetAttendantProperties

7.3.91. GetAttendantByOrdinal

Syntax

Visual Basic

Function GetAttendantByOrdinal(ByVal Index As Int32,
ByRef ID As Int32) As Int32

C/C++/C#

long GetAttendantByOrdinal(long Index ,
long* pID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Index	long	The index of the desired Attendant object, this can be between 1 and the total number of Attendants, as returned by GetAttendantsCount.
ID	long	The returned ID of the attendant object.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.

Remarks

This API is used to retrieve the tank ID of a tank using it's ordinal value in the EZServer's internal tank objects list. The tanks are ordered in this list by ID.

See also

GetAttendantsCount, GetAttendantByNumber, GetAttendantByName, GetAttendantProperties

7.3.92. GetAttendantProperties

Syntax

Visual Basic

Function GetAttendantProperties(ByVal ID As Int32,
ByRef Number As Int32,
ByRef Name As String,
ByRef ShortName As String,
ByRef Password As String) As Int32

C/C++/C#

long GetAttendantProperties(long ID ,
long* pNumber,
BSTR* pName ,
BSTR* pShortName,
BSTR* pPassword) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant object for which the properties are being requested.
Number	long	The attendant number.
Name	BSTR	The attendant name.
ShortName	BSTR	The attendant short name.
Password	BSTR	The attendant password used to logon to a pump.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API returns all the property values of an attendant object for the given ID.

See also

SetAttendantProperties

7.3.93. SetAttendantProperties

Syntax

Visual Basic

```
Function SetAttendantProperties( ByVal ID As Int32,  
                               ByVal Number As Int32,  
                               ByVal Name As String,  
                               ByVal ShortName As String,  
                               ByVal Password As String) As Int32
```

C/C++/C#

```
long SetAttendantProperties(long ID ,  
                           long Number,  
                           BSTR Name,  
                           BSTR ShortName,  
                           BSTR Password ) ;
```

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the Attendant object for which the properties are being requested. This property corresponds to the Attendants.AttendantID column in the EZDB data model.
Number	long	The Attendant number. This property corresponds to the Attendants.AttendantNumber column in the EZDB data model.
Name	BSTR	The Attendant name. This property corresponds to the Attendants.AttendantName column in the EZDB data model.
ShortName	BSTR	The attendant short name. This property corresponds to the Attendants.AttendantShortName column in the EZDB data model.
Password	BSTR	The attendant password used to logon to a pump. This property corresponds to the Attendants.AttendantPassword column in the EZDB data model.

Return value

Error code	Error description
------------	-------------------

OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.

Remarks

This API is provided so that attendant objects can be created and maintained on the EZServer. If the given attendant ID already exists the properties for this object will be overwritten with the values passed, otherwise a new attendant object with this ID is created and its properties set to the values passed.

See also

GetAttendantProperties, SetAttendantProperties

7.3.94. DeleteAttendant

Syntax

Visual Basic

Function DeleteAttendant(ByVal ID As Int32) As Int32

C/C++/C#

long DeleteAttendant(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant object to be deleted from the EZServer.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects in the server which have references to it.

Remarks

This API is used to remove an attendant object from the EZServer. If this attendant is currently logged onto one or more pumps, the call will fail.

See also

SetAttendantProperties

7.3.95. AttendantLogon

Syntax

Visual Basic

Function SetAttendantLogon (ByVal ID As Int32,
ByVal PumpID As Int32) As Int32 *

C/C++/C#

long SetAttendantLogon (long ID ,
long PumpID) ; *

Availability – EZClient.DLL, EZClient.OCX, EZPump.OCX, EZTech.EZClient and
EZTech.EZPump
*this parameter is not present for the EZPump.OCX or EZTech.EZPump controls

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant logging on.
PumpID	long	The ID of the pump this attendant is logging onto. This parameter is not present for the EZPump.OCX or EZTech.EZPump controls

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
PUMP_IN_WRONG_AUTH_MODE_RESULT	The pump that the attendant is attempting to logon to is not in one of the attendant authorization modes.
ATTENDANT_ALREADY_LOGGED_ON_RESULT	The pump the attendant is attempting to logon to, already has an attendant logged on to it.
PUMP_HAS_DELIVERIES_RESULT	The pump being logged on to has uncleared deliveries.

Remarks

This API is used to log an attendant onto a pump. While the attendant remains logged onto this pump, all deliveries done on this pump will be logged to this attendant.



See also

AttendantLogoff

7.3.96. AttendantLogoff

Syntax

Visual Basic

Function SetAttendantLogoff (ByVal ID As Int32) As Int32

C/C++/C#

long SetAttendantLogoff (long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
ID	long	The ID of the attendant logging off.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
ATTENDANT_NOT_LOGGED_ON_RESULT	The attendant being logged off, is not currently logged on.
PUMP_HAS_DELIVERIES_RESULT	The pump being logged off has uncleared deliveries.

Remarks

This API is used to log an attendant off a pump. This will log the attendant off all logged on pumps and close the open attendant report for this attendant.

See also

AttendantLogon, DeletePort, SetPortProperties

7.3.97. AllStop

Syntax

Visual Basic

Function AllStop() As Int32

C/C++/C#

long AllStop();

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

The API is provided to place a temp stop on all pumps with one call. The can be cleared with AllReAuthorise, or each pump can be individually ReAuthorise.

See also

ReAuthorise, TempStop, AllReAuthorise, AllStopIfIdle

7.3.98. AllStopfIdle

Syntax

Visual Basic

Function AllStop() As Int32

C/C++/C#

long AllStop();

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
PUMP_IN_USE_RESULT	One or more of the pumps is not idle, or has available deliveries.

Remarks

The API is provided to place a temp stop on all pumps with one call, if the pumps are all currently idle and do not have deliveries pending. The can be cleared with AllReAuthorise, or each pump can be individually ReAuthorise. This call is provided to assist with the shift end procedure, it is typically called when all the pumps become idle and then is reversed with the AllReAuthorise API.

See also

ReAuthorise, TempStop, AllReAuthorise, AllStop

7.3.99. AllAuthorise

Syntax

Visual Basic

Function AllAuthorise() As Int32

C/C++/C#

long AllAuthorise() ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is provided so that all can be authorized in one operation. To reverse this CancelAuthorise must be called for each individual pump.

See also

CancelAuthorise, AllStop, AllReAuthorise, AllStopIfIdle

7.3.100. AllReAuthorise

Syntax

Visual Basic

Function AllReAuthorise() As Int32

C/C++/C#

long AllReAuthorise() ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
N/A		

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

This API is provided so that AllStop can be reversed in one operation. It has the same effect as calling ReAuthorise for each individual pump.

See also

ReAuthorise, AllStop, TempStop, AllStopIfIdle

7.3.101. SetNextDeliveryID

Syntax

Visual Basic

Function SetNextDeliveryID (ByVal ID As Int32) As Int32

C/C++/C#

long SetNextDeliveryID(long ID) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient
For DB clients only.

Parameters

Parameter	Type	Description
ID	long	The ID used to save the next delivery completed.

Return value

Error code	Error description
OK_RESULT	The call was successful.
NOT_LOGGED_ON_RESULT	The client is not currently logged on.
INVALID_CLIENT_TYPE	The currently logged on client type, does not permit this call.
SERVER_TIMEOUT	The call to the server timed out.
CONNECTION_BROKEN	The connection with the server was lost.
INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.

Remarks

As the EZServer captures the deliveries done on the pumps, they are associated with a unique ID. This ID is generated by simply incrementing in internal long variable. However as the EZServer has no knowledge of the deliveries already logged into the database, it is necessary for the DB Client to give it a starting value. This would normally be the maximum logged delivery ID plus one.

See also

SetDeliveryProperties

7.3.102. ResultString

Syntax

Visual Basic

Function ResultString(ByVal Res As Int32) As String

C/C++/C#

BSTR ResultString(long Res) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Res	long	The return value from any of the API calls listed above.

Return value

An English string describing the numeric error passed.

Remarks

This API can be used to convert an error value to an error description string in English.

See also

Appendix M – Error messages

7.3.103. PumpStateString

Syntax

Visual Basic

Function PumpStateString(ByVal State As Int16) As String

C/C++/C#

BSTR PumpStateString(short State) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
State	short	The numeric value of the pump state.

Return value

An English string describing the pump state value passed.

Remarks

This API can be used to convert a pump state value to a pump state description string in English.

See also

Appendix A – Pump states

7.3.104. DeliveryStateString

Syntax

Visual Basic

Function DeliveryStateString(ByVal State As Int16) As String

C/C++/C#

BSTR DeliveryStateString(short State) ;

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
State	short	

Return value

An English string describing the delivery state value passed.

Remarks

This API can be used to convert a delivery state value to a delivery state description string in English.

See also

Appendix D – Delivery states

7.3.105. DeliveryTypeString

Syntax

Visual Basic

Function DeliveryTypeString(ByVal Type As Int16) As String

C/C++/C#

BSTR DeliveryTypeString(short Type);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Type	short	A delivery type value.

Return value

An English string describing the delivery type value passed.

Remarks

This API can be used to convert a delivery type value to a delivery type description string in English.

See also

Appendix C – Delivery types

7.3.106. ReserveTypeString

Syntax

Visual Basic

Function ReserveTypeString(ByVal Type As Int16) As String

C/C++/C#

BSTR ReserveTypeString(short Type);

Availability – EZClient.DLL, EZClient.OCX and EZTech.EZClient

Parameters

Parameter	Type	Description
Type	short	The pump reserve value.

Return value

An English string describing the pump reserve value passed.

Remarks

This API can be used to convert a pump reserve value to a pump reserve description string in English.

See also

Appendix B – Pump reserves



8. Database reference

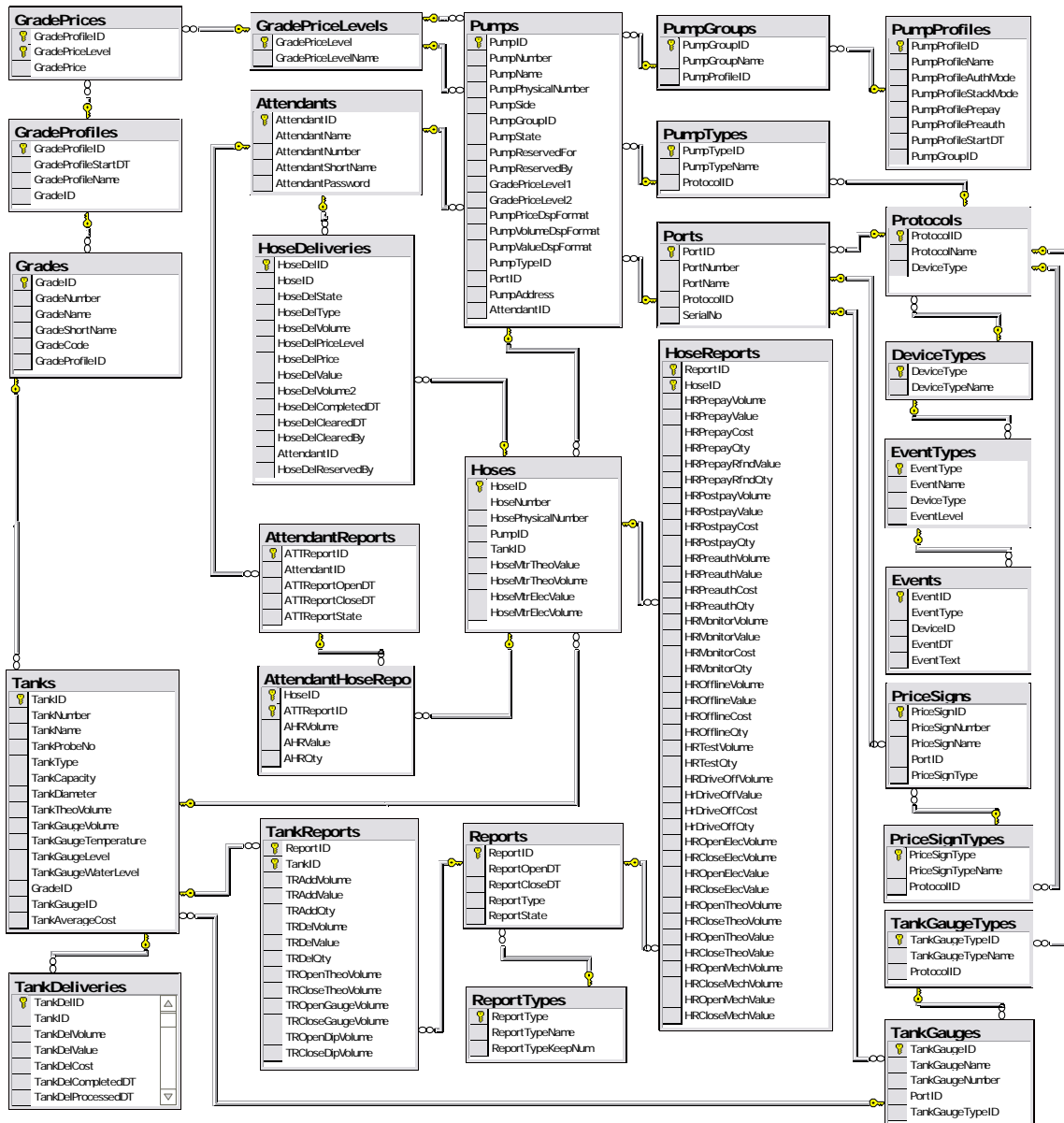
8.1. Introduction

The EZDB Database is only used if the optional database client (EZDBClient) and EZForecourt back office applications are used.

The data model used is fully relational and was built and designed with SQL Server 2000, however it does run successfully with the equivalent MSDE version.

The EZTech provided DB Client (EZDBClient) uses an ODBC connection to access the database, the back office application (EZForecourt) use ADO.net.

8.2. Data Model



8.3. Tables reference

The data types used in the this data model are as follows

- int 4 bytes , corresponds to long or int32 types.
- datetime 8 bytes , corresponds to the DATE type
- decimal(w,d) 18 bytes, numeric with width w and decimals d, corresponds to the double type.
- nchar 2 bytes * n , unicode string of length n, corresponds to the BSTR type.
- Nvarchar 2 bytes * n variable length, variable unicode string, corresponds to The BSTR type.
- Char 1 byte * n, ansii string of length n, corresponds to the BSTR type.

The column qualifiers can be any of the following

- PK Is part or all of the primary key for this table.
- ID Identity, i.e. automatically generated as rows are added.
- FK Is a foreign key, used in a relationship with another table.
- NULL Can contain a NULL value.

The individual table definitions are as follows :-

8.3.1. AttendantHoseReports

Definition

Column Name	Data type	Qualifier	Description
HoseID	int	PK	The hose that these totals belong to.
ATTReportID	int	PK	The attendant report that these totals belong to.
AHRVolume	Decimal(18,3)		The total volume delivered on this hose, by this attendant during this report period.
AHRValue	Decimal(18,2)		The total value delivered on this hose, by this attendant during this report period.
AHRQty	int		The quantity of deliveries on this hose, by this attendant during this report period.

Remarks

The table carries the summary delivery totals by hose and attendant for a specific attendant report.

See also

AttendantReports, Attendants

8.3.2. AttendantReports

Definition

Column Name	Data type	Qualifier	Description
ATTReportID	int	ID,PK	The ID for a specific attendant report.
AttendantID	int	FK	The ID of the attendant for this report.
ATTReportOpenDT	datetime		The date & time the report was opened.
ATTReportCloseDT	datetime	NULL	The date & time the report was closed.
ATTReportState	int		The state of the report, 1=Open, 2=Closed.

Remarks

This table contains a list of all the attendant reports currently in the database. The individual hose totals are contained in the AttendantHoseReports table.

See also

AttendantHoseReports, Attendants

8.3.3. Attendants

Definition

Column Name	Data type	Qualifier	Description
AttendantID	int	ID,PK	The ID for this attendant.
AttendantName	nchar(20)		The attendants full name.
AttendantNumber	int		The attendant number.
AttendantShortName	nchar(10)		The attendants short name.
AttendantPassword	nchar(10)		The attendants password used to logon to a pump.

Remarks

The table is a configuration table, and contains the current list of attendants in the system.

See also

AttendantHoseReports, AttendantReports

8.3.4. DeviceTypes

Definition

Column Name	Data type	Qualifier	Description
DeviceType	int	PK	The device type identifier.
DeviceTypeName	nchar(20)		The device type name.

Remarks

This table is pre-populated when the database is installed. Currently the supported device types and names are as follows.

- 1 – Server.
- 2 – Dispenser.
- 3 – Dispenser driver.
- 4 – Tank.
- 5 – Tank gauge.
- 6 – Tank gauge driver.
- 7 – Price sign.
- 8 – Price sign driver.

These values are used by the EventTypes table to identify the type of device which can generate an event. Also it is used to identify the type of the driver in the Protocols table.

See also

EventTypes, Protocols

8.3.5. EventTypes

Definition

Column Name	Data type	Qualifier	Description
EventType	int	PK	The event type identifier.
EventName	nchar(20)		The name of the event type.
DeviceType	int		The device type ID which can generate this event type.
EventLevel	int		The level of the event type. 1=Information, 2=Warning, 3=Error.

Remarks

The table is pre-populated when the database is initialized and contains definitions of all the event types that can be generated in the system.

See also

Events, DeviceTypes

8.3.6. Events

Definition

Column Name	Data type	Qualifier	Description
EventID	int	ID,PK	The event identifier.
EventType	int	FK	The event type identifier.
DeviceID	int	FK,NULL	The ID of the device which generated the event. The will depend on the device type.
EventDT	datetime		The date and time the event was generated.
EventText	Nvarchar(160)		Additional event description logged along with the event.

Remarks

The table is essentially an event log. And contains all events from all possible sources logged in the one place. The DeviceID value will depend on the DeviceType. The mapping is as follows:-

- 1 = always NULL
- 2 = the PumpID of the pump that generated the event.
- 3 = always NULL.
- 4 = the TankID of the tank that generated the event.
- 5 = the Tank gauge ID that generated the event.
- 6 = always NULL.
- 7 = the PriceSignID of the price sign that generated this event.
- 8 = will always be NULL.

See also

DeviceTypes, EventTypes

8.3.7. GradePriceLevels

Definition

Column Name	Data type	Qualifier	Description
GradePriceLevel	int	PK	The price level number.
GradePriceLevelName	nchar(20)		The price level name.

Remarks

This table is pre-populated when the database is installed and currently has only two rows, they are as follows:-

- 1 = Cash price level.
- 2 = Credit price level.

If is possible for system integrators to add or remove price levels from this table however there must be at least one row and the GradePriceLevel column values must start at 1 and be sequential.

See also

GradePrices, GradeProfiles, Grades

8.3.8. GradePrices

Definition

Column Name	Data type	Qualifier	Description
GradeProfileID	int	PK	The ID of the grade profile that owns this price.
GradePriceLevel	int	PK	The grade price level of this price.
GradePrice	Decimal(18,3)		The grade price.

Remarks

This table contains all the individual grade prices by level and grade profile.

See also

GradePriceLevels, GradeProfiles, Grades

8.3.9. GradeProfiles

Definition

Column Name	Data type	Qualifier	Description
GradeProfileID	int	ID,PK	The identifier of this grade profile.
GradeProfileStartDT	datetime		The start date and time for this grade profile.
GradeProfileName	nchar(20)		The name of the grade profile.
GradeID	int	FK	The ID of the grade that this grade profile applies to.

Remarks

This table is the parent for the GradePrices table, and together they define a grade price profile.

See also

GradePriceLevels, GradePrices, Grades

8.3.10. Grades

Definition

Column Name	Data type	Qualifier	Description
GradeID	int	ID,PK	The identifier for this grade.
GradeNumber	int		The number of this grade.
GradeName	nchar(20)		The long name of this grade.
GradeShortName	nchar(10)		The short name for this grade.
GradeCode	nchar(20)		The code for this grade.
GradeProfileID	int	FK	The ID of the grade profile which is the current grade profile for this grade.

Remarks

This table contains the definitions of all the grades configured in the system.

See also

GradePriceLevels, GradePrices, GradeProfiles, Tanks

8.3.11. HoseDeliveries

Definition

Column Name	Data type	Qualifier	Description
HoseDelID	int	ID,PK	The identifier allocated to this delivery when it was completed.
HoseID	int	FK	The ID of the hose which made this delivery.
HoseDelState	int		The current state of this delivery, see Appendix D – Delivery states for more information.
HoseDelType	int		The type of this delivery, see Appendix C – Delivery types for more information.
HoseDelVolume	Decimal(18,3)		The total volume of the delivery, as reported by the pump.
HoseDelPriceLevel	int		The price level the delivery was done at.
HoseDelPrice	Decimal(18,3)		The price the delivery was done at, as reported by the pump.
HoseDelValue	Decimal(18,2)		The total value of the delivery, as reported by the pump.
HoseDelVolume2	Decimal(18,3)		The second delivery volume of the delivery, this value will always be zero unless the delivery was done on a blender pump.
HoseDelCompletedDT	datetime		The time and date the delivery was completed.
HoseDelClearedDT	datetime	NULL	The time and date the delivery as cleared. I.e. paid for.
HoseDelClearedBy	int	NULL	The ID of the client that cleared the delivery.
AttendantID	int	NULL	The ID of the attendant that authorized this delivery.
HoseDelReservedBy	int	NULL	The ID of the client that reserved this delivery prior to starting.

Remarks

This table is essentially the hose delivery log, each time a delivery is completed is appended to this table and every time it is cleared or stacked etc. the corresponding row is updated.

See also

Hoses, Appendix C – Delivery types, Appendix D – Delivery states

8.3.12. HoseReports

Definition

Column Name	Data type	Qualifier	Description
ReportID	int	PK	The ID of the Report that this summary record belongs to.
HoseID	int	PK	The ID of the hose which generated this summary record.
HRPrepayVolume	decimal(18,3)		The sum of all the prepay delivery volumes done on this hose during this report period.
HRPrepayValue	decimal(18,2)		The sum of all the prepay delivery values done on this hose during this report period.
HRPrepayCost	decimal(18,3)		The sum of all the prepay delivery costs done on this hose during this report.
HRPrepayQty	int		The total number of prepay deliveries done on this hose during this report period.
HRPrepayRfndValue	decimal(18,2)		The sum of all the prepay refund delivery values done on this hose during this report period.
HRPrepayRfndQty	int		The total number of prepay refund deliveries done on this hose during this report period.
HRPostpayVolume	decimal(18,3)		The sum of all the Postpay delivery volumes done on this hose during this report period.
HRPostpayValue	decimal(18,2)		The sum of all the Postpay delivery values done on this hose during this report period.
HRPostpayCost	decimal(18,3)		The sum of all the Postpay delivery costs done on this hose during this report period.
HRPostpayQty	int		The total number of Postpay deliveries done on this hose during this report period.
HRPreauthVolume	decimal(18,3)		The sum of all the preauth delivery volumes done on this hose during this report period.
HRPreauthValue	decimal(18,2)		The sum of all the preauth delivery values done on this hose during this report period.
HRPreauthCost	decimal(18,3)		The sum of all the preauth delivery costs done on this hose during this report.
HRPreauthQty	int		The total number of preauth deliveries done on this hose during this report period.
HRMonitorVolume	decimal(18,3)		The sum of all the monitor delivery volumes done on this hose during this

			report period.
HRMonitorValue	decimal(18,2)		The sum of all the monitor delivery values done on this hose during this report period.
HRMonitorCost	decimal(18,3)		The sum of all the monitor delivery costs done on this hose during this report.
HRMonitorQty	int		The total number of monitor deliveries done on this hose during this report period.
HROfflineVolume	decimal(18,3)		The sum of all the offline delivery volumes done on this hose during this report period.
HROfflineValue	decimal(18,2)		The sum of all the offline delivery values done on this hose during this report period.
HROfflineCost	decimal(18,3)		The sum of all the offline delivery costs done on this hose during this report.
HROfflineQty	int		The total number of offline deliveries done on this hose during this report period.
HRTTestVolume	decimal(18,3)		The sum of all the test delivery volumes done on this hose during this report period.
HRTTestQty	int		The total number of test deliveries done on this hose during this report period.
HRDriveOffVolume	decimal(18,3)		The sum of all the drive off delivery volumes done on this hose during this report period.
HrDriveOffValue	decimal(18,2)		The sum of all the drive off delivery values done on this hose during this report period.
HrDriveOffCost	decimal(18,3)		The sum of all the drive off delivery costs done on this hose during this report.
HrDriveOffQty	int		The total number of drive off deliveries done on this hose during this report period.
HROpenElecVolume	decimal(18,3)		The electronic volume total reading at the beginning of this reporting period for this hose.
HRCloseElecVolume	decimal(18,3)	NULL	The electronic volume total reading at the end of this reporting period for this hose.
HROpenElecValue	decimal(18,2)		The electronic value total reading at the beginning of this reporting period for this hose.
HRCloseElecValue	decimal(18,2)	NULL	The electronic value total reading at the end of this reporting period for this hose.
HROpenTheoVolume	decimal(18,3)		The theoretical volume total at the beginning of this reporting period for this hose.
HRCloseTheoVolume	decimal(18,3)	NULL	The theoretical volume total at the end of this reporting period for this hose.
HROpenTheoValue	decimal(18,2)		The theoretical value total at the beginning of this reporting period for this

			hose.
HRCloseTheoValue	decimal(18,2)	NULL	The theoretical value total at the end of this reporting period for this hose.
HROpenMechVolume	decimal(18,3)		The mechanical volume total reading at the beginning of this reporting period for this hose.
HRCloseMechVolume	decimal(18,3)	NULL	The mechanical volume total reading at the end of this reporting period for this hose.
HROpenMechValue	decimal(18,2)		The mechanical value total reading at the beginning of this reporting period for this hose.
HRCloseMechValue	decimal(18,2)	NULL	The mechanical value total reading at the end of this reporting period for this hose.

Remarks

This table contains all the summary totals for each hose and report period.

See also

ReportTypesReports, TankReports

8.3.13. Hoses

Definition

Column Name	Data type	Qualifier	Description
HoseID	int	ID,PK	The unique identifier for this hose.
HoseNumber	int		The hose number with respect to the pump that it is connected to.
HosePhysicalNumber	int		The hose number with respect to the entire system.
PumpID	int	FK	The ID of the pump that this hose is connected to.
TankID	int	FK	The ID of the tank that this hose is connected to.
HoseMtrTheoValue	decimal(18,2)		The current theoretical total value delivered on this hose since it was configured in the system.
HoseMtrTheoVolume	decimal(18,3)		The current theoretical total volume delivered on this hose since it was configured in the system.
HoseMtrElecValue	decimal(18,2)		The current electronic total value delivered on this hose as reported by the pump.
HoseMtrElecVolume	decimal(18,3)		The current electronic total volume delivered on this hose as reported by the pump.

Remarks

This table contains the configuration/definition of all the hoses contained in the system.

See also

Pumps, Tanks

8.3.14. Ports

Definition

Column Name	Data type	Qualifier	Description
PortID	int	ID,PK	The unique identifier for this port.
PortNumber	int		The port number.
PortName	nchar(20)		The port name.
ProtocolID	int	FK	The ID of the protocol used on this port.
SerialNo	char(9)		The optional serial number of the EZMod connected to this port.

Remarks

This table contains the configuration/definition of all the ports installed in the system. This table is pre-loaded when the database is installed, however the ProtocolID and SerialNo columns can still be edited.

See also

Pumps, Protocols, TankGauges

8.3.15. PriceSignTypes

Definition

Column Name	Data type	Qualifier	Description
PriceSignType	int	PK	The price sign type identifier.
PriceSignTypeName	nchar(20)		The price sign type name.

Remarks

The table contains a list of all the price sign types supported by the EZServer product. This table is pre-populated when the database is installed.

See also

PriceSigns

8.3.16. PriceSigns

Definition

Column Name	Data type	Qualifier	Description
PriceSignID	int	ID,PK	The unique identifier for this price sign.
PriceSignNumber	int		The price sign number.
PriceSignName	nchar(20)		The price sign name.
PortID	int	FK	The ID of the port that this price sign is connected to.
PriceSignType	int	FK	The price sign type identifier.

Remarks

This table contains the configuration/definition of all the price signs contained in the system.

See also

PriceSignTypes

8.3.17. Protocols

Definition

Column Name	Data type	Qualifier	Description
ProtocolID	int	ID,PK	The unique identifier for this protocol.
ProtocolName	nchar(20)		The name of this protocol
DeviceType	int	FK	The type of device that this protocol applies to.

Remarks

This table contains the configuration/definition of all the protocols installed in the system. This table is pre-loaded when the database is installed.

See also

Ports, DeviceTypes

8.3.18. PumpGroups

Definition

Column Name	Data type	Qualifier	Description
PumpGroupID	int	ID,PK	The unique identifier for this pump group.
PumpGroupName	nchar(20)		The name for this group of pumps.
PumpProfileID	int	FK	The ID of the pump profile which is currently is use for this group of pumps.

Remarks

The table contains the definitions of all the pump groups installed in the system. The pumps are then allocated to there respective pump groups.

See also

Pumps, PumpProfiles

8.3.19. PumpProfiles

Definition

Column Name	Data type	Qualifier	Description
PumpProfileID	int	ID,PK	The unique identifier for this pump profile.
PumpProfileName	nchar(20)		The name for this pump profile.
PumpProfileAuthMode	int		The pump authorization mode for this group of pumps, see Appendix G – Pump authorization modes for more information.
PumpProfileStackMode	int		The pump delivery stacking mode for this group of pumps, see Appendix H – Pump delivery stack (memory) modes for more information.
PumpProfilePrepay	int		Whether prepay deliveries are permitted for this pump profile or not, 0 = No, 1 = Yes.
PumpProfilePreauth	int		Whether preauth deliveries are permitted for this pump profile or not, 0 = No, 1 = Yes.
PumpProfileStartDT	datetime	NULL	A start date and time for when this pump profile will be automatically applied to a group of pumps.
PumpGroupID	int	NULL	The ID of the pump group to which this pump profile is scheduled to be applied.

Remarks

This table contains the definition of each of the pump profiles, configured in the system. There will be at least as many pump profiles as pump groups; there may be some unused pump profiles configured, scheduled to be used at a later date.

See also

PumpGroups

8.3.20. PumpTypes

Definition

Column Name	Data type	Qualifier	Description
PumpTypeID	int	ID,PK	The unique identifier for this pump type.
PumpTypeName	nchar(20)		The name of this pump type.
ProcollID	int	FK	The ID of the protocol that is used for this pump type.

Remarks

This table contains a list of all the pump types supported by the EZForecourt system. This table is pre-populated at install time, and is added to as new drivers/protocols become available.

See also

Pumps, Protocols

8.3.21. Pumps

Definition

Column Name	Data type	Qualifier	Description
PumpID	int	ID,PK	The unique identifier for this pump.
PumpNumber	int		The logical number for this pump.
PumpName	nchar(20)		The pump name.
PumpPhysicalNumber	int		The pump physical number.
PumpSide	int		The side of the physical pump for this filling position.
PumpGroupID	int	FK	The ID of the pump group that this pump belongs to.
GradePriceLevel1	int		The EZForecourt price level to be used as price level 1 on this pump.
GradePriceLevel2	int		The EZForecourt price level to be used as price level 2 on this pump.
PumpPriceDspFormat	int		The format of the pump price display, see Appendix F – Pump display formats for more information.
PumpVolumeDspFormat	int		The format of the pump volume display, see Appendix F – Pump display formats for more information.
PumpValueDspFormat	int		The format of the pump value display, see Appendix F – Pump display formats for more information.
PumpTypeID	int	FK	The ID of the pump type for this pump.
PortID	int	FK	The ID of the communications port that this pump is connected to, the address of this pump must be unique for this port.
PumpAddress	int		The electronic polling address, for the pump as configured in the pump.
AttendantID	int	FK,NULL	The ID of the attendant currently logged onto this pump.

Remarks

This table contains the configuration data for each of the logical pumps (pump sides) installed in the system.

See also

Hoses, PumpGroups, PumpTypes

8.3.22. ReportTypes

Definition

Column Name	Data type	Qualifier	Description
ReportType	int	PK	The unique identifier for this report type.
ReportTypeName	nchar(20)		The name for this report type.
ReportTypeKeepNum	int		The number of days that this report type is kept in the system after it is closed.

Remarks

This table is pre-populated at install time. It contains a list of all the report types supported in the system, currently this is just day and month reports.

See also

Reports

8.3.23. Reports

Definition

Column Name	Data type	Qualifier	Description
ReportID	int	ID,PK	The unique identifier for this report.
ReportOpenDT	datetime		The date and time this report was opened / created.
ReportCloseDT	datetime	NULL	The date and time this report was closed, or NULL if it is still open.
ReportType	int	FK	The type of this report, this is a link to the ReportTypes table.
ReportState	int		The state of the report, 1 = Open, 2 = Closed.

Remarks

This table contains a list of all the reports currently contained in the system.

See also

ReportTypes, HoseReports, TankReports

8.3.24. TankDeliveries

Definition

Column Name	Data type	Qualifier	Description
TankDelID	int	ID,PK	The unique identifier for this tank delivery.
TankID	int	FK	The ID of the tank that received the delivery.
TankDelVolume	decimal(18,3)		The volume of the delivery.
TankDelValue	decimal(18,2)		The total value of the tank delivery.
TankDelCost	decimal(18,3)		The unit cost of the tank delivery.
TankDelCompletedDT	datetime		The time and date the tank delivery occurred.
TankDelProcessedDT	datetime	NULL	The time and date the tank delivery was entered into the system.
TankDelNote	nchar(10)		The delivery note number or similar to cross reference this delivery to the delivery documentation.

Remarks

This table is the current list of all tank deliveries in the system.

See also

Tanks

8.3.25. TankGaugeTypes

Definition

Column Name	Data type	Qualifier	Description
TankGaugeTypeID	int	PK	The unique identifier for this tank gauge type.
TankGaugeTypeName	nchar(20)		The name/make/model of this tank gauge type.

Remarks

This table contains a list of all the tank gauge types supported by the EZForecourt system, it is pre-populated and will be add too, as new drivers/protocols are added.

See also

TankGauges

8.3.26. TankGauges

Definition

Column Name	Data type	Qualifier	Description
TankGaugeID	int	ID,PK	The unique identifier for this tank gauge.
TankGaugeName	nchar(20)		The name for this tank gauge.
TankGaugeNumber	int		The number of this tank gauge.
PortID	int	FK	The ID of the port to which this tank gauge is connected.
TankGaugeTypeID	int	FK	The ID of the tank gauge type.

Remarks

This table contains a list of the tank gauges installed in the system. This is rarely more than one but can be.

See also

TankGaugeTypes, Tanks

8.3.27. TankReports

Definition

Column Name	Data type	Qualifier	Description
ReportID	int	PK	The ID of the report which owns this record.
TankID	int	PK	The ID of the tank which owns this record.
TRAddVolume	Decimal(18,3)		The volume added to this tank during this report period.
TRAddValue	Decimal(18,2)		The total value of fuel added to this tank during this report period.
TRAddQty	int		The number of tanks drops done during this report period.
TRDelVolume	Decimal(18,3)		The total volume of hose deliveries taken from this tank during this report period.
TRDelValue	Decimal(18,2)		The total value of the hose deliveries taken from this tank during this report period.
TRDelQty	int		The total number of hose deliveries taken from this tank during this report period.
TROpenTheoVolume	Decimal(18,3)		The theoretical volume in the tank when this report period was opened.
TRCloseTheoVolume	Decimal(18,3)	NULL	The theoretical volume in the tank when this report period was closed, or NULL if it is still open.
TROpenGaugeVolume	Decimal(18,3)		The volume in the tank, as reported by the tank gauge, when this report period was opened.
TRCloseGaugeVolume	Decimal(18,3)	NULL	The volume in the tank, as reported by the tank gauge, when this report period was closed, or NULL if it is still open.
TROpenDipVolume	Decimal(18,3)		The volume in the tank, as determined by a manual dip, when this report period was opened.
TRCloseDipVolume	Decimal(18,3)	NULL	The volume in the tank, as determined by a manual dip, when this report period was closed, or NULL if it is still open.

Remarks

This table contains all the summary totals for each tank report period.

See also

Tanks, Reports

8.3.28. Tanks

Definition

Column Name	Data type	Qualifier	Description
TankID	int	ID,PK	The unique identifier for this tank.
TankNumber	int		The number of this tank.
TankName	nchar(20)		The name of this tank.
TankProbeNo	int		The number of the tank gauge probe inserted into this tank. Or zero if there is no probe in this tank.
TankType	int	FK	The type of this tank.
TankCapacity	decimal(18,3)		The maximum capacity of the tank.
TankDiameter	decimal(18,3)		The diameter of the tank.
TankTheoVolume	decimal(18,3)		The theoretical volume currently in the tank.
TankGaugeVolume	decimal(18,3)		The current volume in the tank as reported by the tank gauge.
TankGaugeTCVolume	decimal(18,3)		The current temperature corrected volume in the tank as reported by the tank gauge.
TankGaugeUllage	decimal(18,3)		The current unused/available capacity in the tank as reported by the tank gauge. This is the capacity minus the volume.
TankGaugeTemperature	decimal(18,2)		The current temperature in the tank as reported by the tank gauge.
TankGaugeLevel	decimal(18,3)		The current level of the fuel in the tank as reported by the tank gauge.
TankGaugeWaterVolume	decimal(18,3)		The current volume of water in the tank as reported by the tank gauge.
TankGaugeWaterLevel	decimal(18,3)		The current level of the water in the tank as reported by the tank gauge.
GradeID	int	FK	The ID of the grade that is contained in this tank.
TankGaugeID	int	FK,NULL	The ID of the tank gauge that is monitoring this tank, or NULL if no probe is present.
TankAverageCost	decimal(18,3)		The average cost of the fuel contained in this tank.

Remarks

This table contains the configuration of all the tanks present in the system.

See also

TankGauges, Grades

8.4. Stored procedures reference

The EZDBClient included with the optional back office, calls several stored procedures in order to process database events. Stored procedures were used for all database update operations, instead of embedded SQL to facilitate modification of the SQL by third parties.

The stored procedures are as follows:-

8.4.1. ezspAddHoseReport

Purpose

This stored procedure creates open hose report entries for all open reports for a specific hose.

Called by

It is called by the EZBackOffice application when a new hose is added.

Parameters

Parameter Name	Data type	Description
@HoseID	int	The ID of the hose for which to create the report.

See also

ezspCloseReport

8.4.2. ezspAttendantLogoff

Purpose

This stored procedure does all the necessary processing for logging a forecourt attendant off all pumps that he/she is currently logged onto.

Called by

It is called by the EZDBClient service extension, when an attendant logs off.

Parameters

Parameter Name	Data type	Description
@AttendantID	int	The ID of the attendant logging off.
@Result	int OUTPUT	The result of the call 1 for success, 0 for failure.

See also

ezspAttendantLogon

8.4.3. ezspAttendantLogon

Purpose

This stored procedure does all the necessary processing for logging a forecourt attendant onto a specific pump.

Called by

It is called by the EZDBClient service extension.

Parameters

Parameter Name	Data type	Description
@AttendantID	int	The ID of the attendant logging on.
@PumpID	int	The ID of the pump the attendant is logging onto.
@Result	int OUTPUT	The result of the call 1 for success, 0 for failure.

See also

ezspAttendantLogoff

8.4.4. ezspClearHoseDelivery

Purpose

This stored procedure is called when a delivery is cleared; it sets the delivery status to cleared and updates all of the associated report records.

Called by

It is called by the EZDBClient service extension, when a delivery is cleared.

Parameters

Parameter Name	Data type	Description
@Result	int OUTPUT	The result of the call, 1 for success, 0 for failure.
@DeliveryID	int	The ID of the delivery being cleared.
@DeliveryType	int	The delivery type that the delivery is being cleared as.
@ClearedBy	int	The ID of the client that is clearing the delivery.
@ClearedDT	datetime	The date and time that the delivery was cleared.
@AttendantID	int	The ID of the attendant that is clearing the delivery or NULL ID if it is not being cleared by an attendant.

See also

ezspLogHoseDelivery, ezspStackHoseDelivery

8.4.5. ezspCloseReport

Purpose

This stored is called to close all open reports of a specific type and open a new report of the same type.

Called by

This stored procedure is called by the EZBackOffice application.

Parameters

Parameter Name	Data type	Description
@Result	int OUTPUT	The result of the call, 1 for success, 0 for failure.
@ReportType	int	The type of the report being closed, see ReportTypes for more explanation.

See also

ezspAddHoseReport

8.4.6. ezspGradeProfileChange

Purpose

This stored procedure is called to force the EZServer service to reload the grade information for all grades covered by this grade profile.

Called by

It is call by the grade prices table change trigger.

Parameters

Parameter Name	Data type	Description
@GradeProfileID	int	The ID of the grade profile that has changed.

See also

ezspPumpGroupChange, ezspPumpProfileChange

8.4.7. ezsplnitDB

Purpose

This stored procedure is provided so that the EZDB database can be initialized. It populates all the pre-populated tables and creates a default setup that works with the developer's kit.

Called by

The stored procedure is called by the EZForecourt BackOffice installation option.

Parameters

This stored procedure has no parameters.

8.4.8. ezspLogHoseDelivery

Purpose

This stored procedure is called when a hose delivery is completed, and is used to log all the details of the delivery to the database.

Called by

It is called by the EZDBClient service extension, when a delivery is completed.

Parameters

Parameter Name	Data type	Description
@Result	int OUTPUT	The result of the call, 1 for success, 0 for failure.
@DeliveryID	int	The ID of the completed delivery.
@HoseID	int	The ID of the hose that made the delivery.
@DeliveryState	int	The initial state of the delivery, see Appendix D – Delivery states for more information.
@DeliveryType	int	The initial type of the delivery, see Appendix C – Delivery types for more information.
@Volume	numeric(18,3)	The volume of the delivery.
@PriceLevel	int	The price level the delivery was taken at.
@Price	numeric(18,3)	The price level the delivery was taken at.
@Value	numeric(18,2)	The total value of the delivery.
@Volume2	numeric(18,3)	The second volume of the delivery, this is only relevant for mixer dispenses and is currently always passed as zero.
@ReservedBy	int	The ID of the terminal that initiated this prepay or preauth delivery, or NULL ID if not a reserved delivery.
@AttendantID	int	The ID of the attendant currently logged onto this pump.
@DeliveryDT	datetime	The date and time this delivery was completed.

See also

ezspClearHoseDelivery, ezspProcessHoseETotals, ezspStackHoseDelivery

8.4.9. ezspNightly

Purpose

This stored procedure is called once every night and is used to perform database backup and purging purposes.

Called by

The MSDE / MSSQL scheduler.

Parameters

This stored procedure has no parameters.

8.4.10. ezspProcessHoseETotals

Purpose

This routine is called when the electronic totals are successfully retrieved from a pump. If this was immediately after a hose delivery, then the value and volume of the associated delivery is also logged.

Called by

It is called by the EZDBClient service extension.

Parameters

Parameter Name	Data type	Description
@Result	int OUTPUT	The result of the call, 1 for success, 0 for failure.
@HoseID	int	The ID of the hose that these E-Totals belong to.
@Volume	numeric(18,3)	The volume of the associated delivery or zero if there is none.
@Value	numeric(18,2)	The value of the associated delivery or zero if there is none.
@VolumeETot	numeric(18,3)	The new electronic volume total retrieved from the pump for this hose.
@ValueETot	numeric(18,4)	The new electronic value total retrieved from the pump for this hose.

See also

ezspLogHoseDelivery

8.4.11. ezspProcessMeterReadings

Purpose

This stored procedure is called to process the mechanical meter readings for a specific hose, and is used to associate the entered meter readings with the close of a specific reporting period.

Called by

This stored procedure is called by the EZBackOffice application.

Parameters

Parameter Name	Data type	Description
@ReportID	Int	The ID of the report that this meter reading belongs to.
@HoseID	Int	The ID of the hose to which these readings belong.
@Volume	numeric(18,3)	The volume mechanical meter value.
@Value	numeric(18,2)	The value mechanical meter value.

See also

ezspProcessTankDip, ezspProcessTankDrop

8.4.12. ezspProcessTankDip

Purpose

This stored procedure is called to process manual tank dip readings for a specific tank, and is used to associate the volume read with the close of a specific reporting period.

Called by

This stored procedure is called by the EZBackOffice application.

Parameters

Parameter Name	Data type	Description
@ReportID	int	The ID of the report that this meter reading belongs to.
@TankID	int	The ID of the tank from which this reading was taken.
@Volume	numeric(18,3)	The volume read from the tank.

See also

ezspProcessMeterReadings, ezspProcessTankDrop

8.4.13. ezspProcessTankDrop

Purpose

This stored procedure is called to process the addition of fuel into a specific tank.

Called by

This stored procedure is called by the EZBackOffice application.

Parameters

Parameter Name	Data type	Description
@TankID	int	The ID of the tank that is having the extra fuel added to it.
@CompletedDT	datetime	The time and date of the fuel drop.
@Volume	numeric(18,3)	The volume of the fuel drop.
@Value	numeric(18,2)	The total value of the fuel drop.
@Cost	numeric(18,3)	The unit cost of the fuel drop.

See also

ezspProcessMeterReadings, ezspProcessTankDip

8.4.14. ezspPumpGroupChange

Purpose

This stored procedure is called when ever a pump group is changed; it is used to send a signal to the EZServer service to force it to reload the details for all affected pumps.

Called by

It is called by the change trigger on the PumpGroups table.

Parameters

Parameter Name	Data type	Description
@PumpGroupID	int	The ID of the pump group being changed.

See also

ezspGradeProfileChange, ezspPumpProfileChange

8.4.15. ezspPumpProfileChange

Purpose

This stored procedure is called when ever a pump profile is changed; it is used to send a signal to the EZServer service to force it to reload the details for all affected pumps.

Called by

It is called by the change trigger on the PumpProfiles table.

Parameters

Parameter Name	Data type	Description
@PumpProfileID	int	The ID of the pump profile being changed.

See also

ezspGradeProfileChange, ezspPumpGroupChange

8.4.16. ezspStackHoseDelivery

Purpose

This stored procedure is called to change the state of a delivery from current to stacked.

Called by

It is called by the EZDBClient service extension.

Parameters

Parameter Name	Data type	Description
@Result	int OUTPUT	The result of the call, 1 for success, 0 for failure.
@DeliveryID	int	The ID of the delivery being stacked.

See also

ezspLogHoseDelivery, ezspClearHoseDelivery

9. Appendices

9.1. Appendix A – Pump states

Value	Name	Description
0	INVALID_PUMP_STATE	The EZPump.OCX control has lost its connection with the server.
1	NOT_INSTALLED_PUMP_STATE	The pump not installed, it has most likely been deleted.
2	NOT_RESPONDING_PUMP_STATE	The pump is installed but is not responding to the server.
3	IDLE_PUMP_STATE	The pump is responding and is idle, all hoses are stowed, if a hose is pulled it will call for authorization, i.e. it will not start delivering without manual authorization.
4	PRICE_CHANGE_STATE	The pump is currently performing a price change. This state only exists for certain pump types.
5	AUTHED_PUMP_STATE	The pump is idle with all of its hoses stowed. If a hose is pulled it will start delivering automatically.
6	CALLING_PUMP_STATE	A hose was pulled while the pump was in the IDLE_PUMP_STATE, it will not start delivering until it receives a manual authorization.
7	DELIVERY_STARTING_PUMP_STATE	A hose has been pulled and the pump is authorized, however the volume in progress total is still zero.
8	DELIVERING_PUMP_STATE	A hose is pulled and the volume in progress is not zero.
9	TEMP_STOPPED_PUMP_STATE	The pump has been temporarily stopped while in one of the delivering states. To get it out of this state the pump will require re-authorizing or the pump delivery will need to be terminated.
10	DELIVERY_FINISHING_PUMP_STATE	The pump is delivering; however the volume in progress total has not changed in the last 5 seconds.
11	DELIVERY_FINISHED_PUMP_STATE	A hose has been returned after a delivering some fuel. The state is temporary and only exists while the server retrieves the delivery totals and pump electronic totals from the pump.
12	DELIVERY_TIMEOUT_PUMP_STATE	The pump has been in one of the delivering states for more than the 'Delivering Timeout' as configured in the EZServer.ini file.
13	HOSE_OUT_PUMP_STATE	The pump has been in the DELIVERY_FINISHING_PUMP_STATE for

		more than the 'HoseOutTimeout' as configured in the EZServer.ini file.
14	PREPAY_REFUND_TIMEOUT_STATE	A prepay delivery has finished and a refund was generated. The pump then remains locked out for 'PrepayRefundTimeout' seconds as configured in the EZServer.ini file.
15	ERROR_PUMP_STATE	The pump has entered an error state; This is pump type specific.

9.2. Appendix B – Pump reserves

Value	Name	Description
1	NOT_RESERVED	There are no reserves placed on the pump.
2	RESERVED_FOR_PREPAY	The pump has been reserved for a prepay delivery and is waiting for the prepay authorization.
3	AUTHED_FOR_PREPAY	The pump has been authorized for a prepay delivery and is waiting for the client to complete the delivery.
4	RESERVED_FOR_PREAUTH	The pump has been reserved for a preauth delivery and is waiting for the preauth authorization
5	AUTHED_FOR_PREAUTH	The pump has been authorized for a preauth delivery and is waiting for the client to complete the delivery.

9.3. Appendix C – Delivery types

Value	Name	Description
1	POSTPAY	A normal delivery which was delivered and paid for after it was completed prior to clearing it from the system
2	PREPAY	A delivery which was paid for prior to delivering.
3	PREPAY_REFUND	A refund delivery generated when a prepay delivery is taken for less than the prepaid amount.
4	PREAUTH	A pre-authorized delivery that was paid for after it was completed and before it was cleared from the system.
5	MONITOR	A normal delivery that was cleared automatically from the system after the MonitorDeliveryTimeout value was reached, as configured in the EZServier.ini file.
6	TEST	A normal delivery that was returned to the tank after completion. This is usually occurs when the pump is being calibrated or tested.
7	DRIVEOFF	A normal delivery for which the driver has driven off without paying. In order to balance the shift totals it cannot be cleared as a Postpay delivery.
8	OFFLINE	A pseudo delivery generated when the server detects that the pump has done deliveries while offline, this is detected by comparing the saved electronic totals last retrieved from the pump, with the current electronic totals. It could well be more than one delivery summed together.

9.4. Appendix D – Delivery states

Value	Name	Description
1	CURRENT	The current delivery on the pump, i.e. the delivery displayed on the pump is guaranteed to be the same as this delivery. This pump will not delivery again until this delivery is cleared or stacked
2	STACKED	This is a stacked delivery; the pump is now free to deliver again.
3	CLEARED	The delivery has been paid for and is cleared from the system.

9.5. Appendix E – Event types

Value	Name	Description
100	SERVER_INFO_EVENT	
102	SERVER_STARTED_EVENT	
103	SERVER_STOPPED_EVENT	
200	DISPENSER_INFO_EVENT	
201	DISPENSER_INSTALLED_EVENT	
202	DISPENSER_STARTED_EVENT	
203	DISPENSER_STOPPED_EVENT	
204	DISPENSER_E_TOTALS_ERROR_EVENT	
205	DISPENSER_OFFLINE_DELIVERIES_EVENT	
206	DISPENSER_REMOTE_PRICE_CHANGE_EVENT	
207	DISPENSER_LOST_DELIVERY_EVENT	
300	DISPENSER_DRIVER_INFO_EVENT	
301	DISPENSER_DRIVER_STARTED_EVENT	
302	DISPENSER_DRIVER_STOPPED_EVENT	
400	TANK_INFO_EVENT	
401	TANK_SETUP_DATA_WARNING_EVENT	
402	TANK_LEAK_ALARM_EVENT	
403	TANK_HIGH_WATER_ALARM_EVENT	
404	TANK_HIGH_WATER_ALARM_EVENT	
405	TANK_OVERFLOW_ALARM_EVENT	
406	TANK_LOW_PRODUCT_ALARM_EVENT	
407	TANK_SUDDEN_LOSS_ALARM_EVENT	
408	TANK_HIGH_PRODUCT_ALARM_EVENT	
409	TANK_INVALID_FUEL_LEVEL_ALARM_EVENT	
410	TANK_PROBE_OUT_ALARM_EVENT	
411	TANK_HIGH_WATER_WARNING_EVENT	
412	TANK_DELIVERY_NEEDED_WARNING_EVENT	
413	TANK_MAXIMUM_PRODUCT_ALARM_EVENT	
414	TANK_GROSS_LEAK_TEST_FAILED_ALARM_EVENT	
415	TANK_PERIODIC_LEAK_TEST_FAILED_ALARM_EVENT	
416	TANK_ANNUAL_LEAK_TEST_FAILED_ALARM_EVENT	
417	TANK_PERIODIC_TEST_NEEDED_WARNING_EVENT	
418	TANK_ANNUAL_TEST_NEEDED_WARNING_EVENT	
419	TANK_PERIODIC_TEST_NEEDED_ALARM_EVENT	
420	TANK_ANNUAL_TEST_NEEDED_ALARM_EVENT	
421	TANK_LEAK_TEST_ACTIVE_INFO_EVENT	
422	TANK_NO_CSLED_IDLE_TIME_WARNING_EVENT	
423	TANK_SIPHON_BREAK_ACTIVE_WARNING_EVENT	
424	TANK_CSLED_RATE_INCREASE_WARNING_EVENT	
425	TANK_ACCUCHART_CALIBRATION_WARNING_EVENT	
426	TANK_HRM_RECONCILIATION_WARNING_EVENT	
427	TANK_HRM_RECONCILIATION_ALARM_EVENT	
428	TANK_COLD_TEMPERATURE_WARNING_EVENT	
429	TANK_MISSING_DELIVERY_TICKET_WARNING_EVENT	
430	TANK_LINE_GROSS_LEAK_ALARM_EVENT	
500	TANK_GAUGE_INFO_EVENT	



501	TANK_GAUGE_INSTALLED_EVENT	
502	TANK_GAUGE_STARTED_EVENT	
503	TANK_GAUGE_STOPPED_EVENT	
600	TANK_GAUGE_DRIVER_INFO_EVENT	
601	TANK_GAUGE_DRIVER_STARTED_EVENT	
602	TANK_GAUGE_DRIVER_STOPPED_EVENT	
700	PRICE_SIGN_INFO_EVENT	
701	PRICE_SIGN_INSTALLED_EVENT	
702	PRICE_SIGN_STARTED_EVENT	
703	PRICE_SIGN_STOPPED_EVENT	
800	PRICE_SIGN_DRIVER_INFO_EVENT	
801	PRICE_SIGN_DRIVER_STARTED_EVENT	
802	PRICE_SIGN_DRIVER_STOPPED_EVENT	

9.6. Appendix F – Pump display formats

Value	Name	Description
1	PUMP_DISPLAY_4_3	9.999
2	PUMP_DISPLAY_4_2	99.99
3	PUMP_DISPLAY_4_1	999.9
4	PUMP_DISPLAY_4_0	9999
5	PUMP_DISPLAY_5_3	99.999
6	PUMP_DISPLAY_5_2	999.99
7	PUMP_DISPLAY_5_1	9999.9
8	PUMP_DISPLAY_5_0	99999
9	PUMP_DISPLAY_6_3	999.999
10	PUMP_DISPLAY_6_2	9999.99
11	PUMP_DISPLAY_6_1	99999.9
12	PUMP_DISPLAY_6_0	999999

9.7. Appendix G – Pump authorization modes

Value	Name	Description
1	NOT_AUTHABLE	The pump cannot be authorized for post pay deliveries, i.e. the pump can only be used for Prepay or Preauth deliveries.
2	COMP_AUTH	The pump must be manually authorized before it will start to deliver.
3	AUTO_AUTH	The pump will automatically authorize when the hose is pulled, however there must be a no current delivery or the stack must be set to auto stacking with space available on the stack.
4	MONITOR_AUTH	The pump will automatically authorize always, and completed deliveries will auto clear after a configured amount of time.
5	ATTENDANT_AUTH	This is the same as auto authorize with the added requirement that an attendant must be logged on to the pump.
6	ATTENDANT_MONITOR_AUTH	This is the same as monitor authorize with the added requirement that an attendant must be logged on to the pump.

9.8. Appendix H – Pump delivery stack (memory) modes

Value	Name	Description
1	STACK_DISABLED	The pump delivery stack is disabled.
2	STACK_MANUAL	Pump deliveries must be shifted onto the stack manually.
3	STACK_AUTO	Pump deliveries will automatically be shifted on the stack when a pump starts calling, if there is space available on the stack.

9.9. Appendix I – Pump limit types

Value	Name	Description
1	NO_PRESET_TYPE	There is not limit placed on the pump.
2	DOLLAR_PRESET_TYPE	There a preset value limit placed on the pump.
3	VOLUME_PRESET_TYPE	There a preset volume limit placed on the pump.
4	DOLLAR_PREPAY_TYPE	There a prepay value limit placed on the pump.
5	VOLUME_PREPAY_TYPE	There a prepay volume limit placed on the pump.
6	DOLLAR_PREAUTH_TYPE	There a preauth value limit placed on the pump.
7	VOLUME_PREAUTH_TYPE	There a preauth volume limit placed on the pump.

9.10. Appendix J – Permitted hoses mask

Value	Name	Description
0	No hoses	No permitted hoses.
1	Hose 1	Logical hose number 1 permitted.
2	Hose 2	Logical hose number 2 permitted.
4	Hose 3	Logical hose number 3 permitted.
8	Hose 4	Logical hose number 4 permitted.
16	Hose 5	Logical hose number 5 permitted.
32	Hose 6	Logical hose number 6 permitted.
64	Hose 7	Logical hose number 7 permitted.
128	Hose 8	Logical hose number 8 permitted.

To permit more than one hose simply add desired hoses together, for example 3 would permit hoses 1 and 2.

9.11. Appendix K – Tank Types

Value	Name	Description
1	MANUAL_DIP_TANK_TYPE	The tank is not gauged and has its level/volume read manually.
2	GAUGED_TANK_TYPE	The tank has a tank gauge probe fitted, and the volume and level are read automatically by a tank gauge.

9.12. Appendix L – Device Types

Value	Name	Description
1	DISPENSER_DEVICE	A fuel dispenser or pump device.
2	DISPENSER_DRIVER_DEVICE	A software driver module for a fuel dispenser or pump device.
3	TANK_GAUGE_DEVICE	An electronic tank gauge device.
4	TANK_GAUGE_DRIVER_DEVICE	A software driver module for an electronic tank gauge device.
5	PRICE_SIGN_DEVICE	An electronic price sign device.
6	PRICE_SIGN_DRIVER_DEVICE	A software driver module for an electronic price sign device.

9.13. Appendix M – Error messages

Value	Name	Description
0	OK_RESULT	The call was successful
-1	INVALID_HEADER_VERSION	An internal error has occurred contact EZTech technical support for assistance.
-2	INVALID_INTERFACE_VERSION	An internal error has occurred contact EZTech technical support for assistance.
-3	INVALID_INTERFACE_ID	An internal error has occurred contact EZTech technical support for assistance.
-4	INVALID_FUNCTION_ID	An internal error has occurred contact EZTech technical support for assistance.
-5	INVALID_SOURCE_ID	An internal error has occurred contact EZTech technical support for assistance.
-6	INVALID_DESTINATION_ID	An internal error has occurred contact EZTech technical support for assistance.
-7	INVALID_OBJECT_ID	An internal error has occurred contact EZTech technical support for assistance.
-8	INVALID_SEQUENCE_NO	An internal error has occurred contact EZTech technical support for assistance.
-9	MSG_IN_BUFFER_OVERFLOW	An internal error has occurred contact EZTech technical support for assistance.
-10	MSG_OUT_BUFFER_OVERFLOW	An internal error has occurred contact EZTech technical support for assistance.
-11	PARAMETER_TYPE_MISMATCH	An internal error has occurred contact EZTech technical support for assistance.
-12	PARAMETER_COUNT_MISMATCH	An internal error has occurred contact EZTech technical support for assistance.
-13	SERVER_TIMEOUT	The call to the server timed out, although a calls socket was opened successfully.
-14	CONNECTION_BROKEN	The connection with the server was broken.
-15	SOCKET_READ_ERROR	The read socket has been terminated.
-16	NO_MSG_ERROR	There are no waiting events/messages.

-17	SOCKET_WRITE_ERROR	The write socket has been terminated.
-18	SERVER_NONASYNC_CALL	An internal error has occurred contact EZTech technical support for assistance.
-19	SOCKET_NOT_CONNECTED	The call socket is not currently connected with the server.
-20	CLIENT_NOT_CONNECTED	The client is not currently connected/logged on to the server.
-21	OCX_NOT_CONNECTED	The OCX control is not currently connected/logged on to the server.
-22	INVALID_CLIENT_TYPE	The logged on client type does not permit this type of call.
-23	INTERNAL_SERVER_ERROR	An unspecified internal error occurred; contact EZTech technical support for assistance.
1	OBJECT_EXISTS_RESULT	The object being created already exists.
2	OBJECT_DOES_NOT_EXIST_RESULT	The object referenced does not exist.
3	OBJECT_HAS_DEPENDANCIES_RESULT	This object cannot be deleted as there are other objects which reference it
4	INVALID_INTERFACE_RESULT	An internal error has occurred contact EZTech technical support for assistance.
5	INVALID_EVENTS_SOCKET_RESULT	An internal error has occurred contact EZTech technical support for assistance.
6	INVALID_OBJECT_LINK_RESULT	One of the IDs passed representing a related object references a non-existing object.
7	INVALID_OBJECT_PARAMETER_RESULT	One of the property values passed is out of range.
8	NOT_LOGGED_ON_RESULT	The client is not currently logged on.
9	ALREADY_LOGGED_ON_RESULT	A client with this client ID is already logged on.
10	INVALID_LOGON_RESULT	An internal error has occurred contact EZTech technical support for assistance.
11	INVALID_CLIENT_TYPE_RESULT	The client type is not database client.
12	PUMP_NOT_RESPONDING_RESULT	The pump is not responding.
13	PUMP_IN_USE_RESULT	The pump is not idle.
14	PUMP_ALREADY_RESERVED_RESULT	The pump already has either a prepay or preauth reserve placed on it.
15	PUMP_NOT_AVAILABLE_RESULT	The pump has a current delivery which cannot be stacked automatically.

16	PUMP_NOT_RESERVED_RESULT	The pump does not have a preauth or prepay reserved place on it.
17	PUMP_NOT_RESERVED_FOR_PREPAY_RESULT	The pump was not reserved for a prepay.
18	PUMP_NOT_RESERVED_BY_YOU_RESULT	The pump was not reserved by this client.
19	INVALID_PRESET_TYPE_RESULT	The preset type requested is not permitted for this type of authorization.
20	INVALID_HOSE_MASK_RESULT	None of the permitted hoses are configured on this pump.
21	PUMP_NOT_RESERVED_FOR_PREAUTH_RESULT	The pump was not reserved for a preauth.
22	PREPAYS_NOT_PREMITTED_RESULT	Prepays are not permitted on this pump.
23	PREAUTHS_NOT_PREMITTED_RESULT	Preauths are not permitted on this pump.
24	PUMP_CANNOT_BE_AUTHED_RESULT	The pump mode does not permit post pay deliveries.
25	PUMP_NOT_AUTHED_RESULT	The pump was not manually authorized.
26	NO_DELIVERY_AVAILABLE_RESULT	The delivery requested does not exist.
27	STACK_IS_DISABLED_RESULT	Stacking deliveries is not permitted on this pump.
28	NO_CURRENT_DELIVERY_RESULT	This pump has no current delivery.
29	STACK_FULL_RESULT	The delivery stack for this pump is full.
30	PUMP_NOT_STOPPED_RESULT	The pump is not temp stopped.
31	PUMP_NOT_DELIVERING_RESULT	The pump is not currently delivering.
32	INVALID_PRESET_AMOUNT_RESULT	The preset amount is less than the configured minimum.
33	PUMP_IS_STOPPED_RESULT	The pump is temp stopped.
34	DELIVERY_ALREADY_LOCKED_RESULT	This delivery has already been locked, by you or another client.
35	DELIVERY_IS_RESERVED_RESULT	The delivery was done as a prepay or preauth delivery and is reserved for another EZClient.
36	DELIVERY_NOT_LOCKED_RESULT	The delivery is not currently locked.
37	DELIVERY_NOT_LOCKED_BY_YOU_RESULT	The delivery is locked by another EZServer client.
38	DELIVERY_TYPE_CANNOT_BE_STACKED_RESULT	This type of delivery cannot be stacked.
39	DELIVERY_CANNOT_BE_CLEARED_AS_THIS_TYPE_RESULT	The original type of the delivery does not permit it to be cleared as this type.
40	DELIVERY_NOT_CURRENT_RESULT	The delivery is not the current delivery.
41	INVALID_CLIENT_ID_RESULT	The current Client ID is not

		logged onto the server.
42	DELIVERY_TERMINATED_RESULT	A delivery on the pump has been terminated, and hence the pump cannot be re-authorized.
43	HAS_CURRENT_DELIVERY_RESULT	The pump has a current delivery which cannot be automatically stacked.
44	ATTENDANT_NOT_LOGGED_ON_RESULT	The attendant being logged off, is not currently logged on.
45	ATTENDANT_ALREADY_LOGGED_ON_RESULT	The pump the attendant is attempting to logon to, already has a logged on attendant.
46	PUMP_IN_WRONG_AUTH_MODE_RESULT	The pump that the attendant is attempting to logon to is not in one of the attendant authorization modes.
47	PUMP_HAS_DELIVERIES_RESULT	The pump being logged on/off has uncleared deliveries.
48	SERVER_NOT_LICENSED_RESULT	The license key for the server is invalid or absent.
49	NO_EZMOD_RESULT	The EZModule for this license key cannot be found, it is most likely turned off or not plugged in.
50	LICENSE_EXPIRED_RESULT	The license key has expired.

9.14. Appendix N – Client event types

Value	Type	Event retrieval function
0	NO_CLIENT_EVENT	**** the events queue is empty ****
1	PUMP_EVENT	GetNextPumpEvent / PumpStatusEvent / StatusEvent
2	DELIVERY_EVENT	GetNextDeliveryEvent[Ex] / DeliveryEvent
3	SERVER_EVENT	GetNextServerEvent / ServerEvent
4	CLIENT_EVENT	GetNextClientEvent/ClientEvent
5	DB_LOG_EVENT	GetNextDBLogEvent/DBLogEvent
6	DB_LOG_DELIVERY	GetNextDBLogDeliveryEvent/DBLogDeliveryEvent
7	DB_CLEAR_DELIVERY	GetNextDBCclearDeliveryEvent/DBCclearDeliveryEvent
8	DB_STACK_DELIVERY	GetNextDBStackDeliveryEvent/DBStackDeliveryEvent
9	DB_LOG_ETOTALS	GetNextDBHoseETotalsEvent/DBHoseETotalsEvent
10	DB_TRIGGER	GetNextDBTriggerEvent/DBTriggerEvent
11	DB_ATTENDANT_LOGON_EVENT	GetNextDBAttendantLogonEvent / DBAttendantLogonEvent
12	DB_ATTENDANT_LOGOFF_EVENT	GetNextDBAttendantLogoffEvent / DBAttendantLogonEvent
13	DELIVERY_EX_EVENT	GetNextDeliveryEvent[Ex] / DeliveryEventEx