ATIONet - Native Transaction Protocol Specification v1.2

Document Information	
File:	ATIONet-Native_Transaction_Protocol-Spec-v1.2
Doc Version:	1.2
Release Date:	16, July 2014
Author:	ATIO International LLC

Change Log		
Ver.	Date	Change Summary
1.0	04/Jan/2013	Initial version. Covers Fleet Fueling Transactions. No Dry products support.
1.1	05/Jan/2013	1.1 Protocol version changes - Enhanced Limits explanation - Converted message format to JSON
1.2	17/July/2014	1.2 Protocol version changes - Add contingency request - Add sale request - Add cancellation request - Enhanced Customer Data and Product Data fields description - Add Original Data Field in TREQ messages - Enhanced Transaction Codes table - Enhanced Customer Data and Product reference tables - Add Authorization Codes table - Add Response Codes table - Add Original Data tables

Contents

Overview

Introduction

This specification is intended to document ATIONet's Native Protocol messaging format and related features required for the systems applying for integration with ATIONet. The following sections provide descriptions of the messages themselves, the expected behaviour for each supported transaction type and a common ground for the functionality of each relevant item.

Definitions

Host

A computer system that is accessed by a user working at a remote location. In this document, Host is always the ATIONet Host.

Terminal

An electronic merchant card processing device responsible for transaction capture, display output to the cashier and/or to the cardholder on screen and/or print format.

Controller

A client system that can send or receive data to and from ATIONet's Host. A Controller controls or includes one or more terminal. When there is only one Terminal connected to a Controller, Terminal and Controller are equivalent.

TREQ

Transaction Request.

TRESP

Transaction Response.

1 ATIOnet Integration Documentation Scope

Third-party systems integrate with ATIOnet via a set of APIs (Application Programming Interfaces). Each ATIOnet's API is described on a separate Protocol Specification. The complete documentation of ATIOnet API's is comprised of:

ATIOnet Native Transactions Protocol Specification:

Covers financial transactions for transaction capture systems (payment terminals, site controllers and point of sale systems), including sales and refunds.

ATIOnet Administrative Transactions Protocol Specification:

Describes a set of functions complementing the transaction-capture business, for example Batch or Shift Close. These functions enhance the capabilities of the integration but their implementation is not mandatory.

ATIOnet Native Interface Protocol Specification:

Covers system-to-system integration capabilities of ATIOnet, designed to interact with third-party back-end systems, for example downloading transactions data or sending current-accounts

movements to ATIOnet. This API is reserved and requires ATIOnet and Subscriber permissions.

ATIOnet Maintenance Interface Protocol Specification:

List a set of functions designed to help in the maintenance and support of a network of capture terminals, for example checking terminal's status via a Keep-alive message. This API is designed to support ATIOnet's own line of capture and gateway devices and thus is a reserved protocol.

In addition to one or more protocol specifications, Integration Projects must have an "Integration Scope Document" detailing the feature-set to be implemented by the capture system, which also defines the acceptance criteria for the project.

2 Scope

Version 1.2 of this document covers a particular version of ATIONet's Host protocol. Although feature's descriptions are generally not related to a particular version of the protocol, some changes may apply which would be specifically commented and identified on each feature's description paragraph.

2.1 Scope Details

Protocol: ATIONet Native Transaction Protocol

Version: Version 1.2

API URI: native.ationet.com/v1/auth

2.2 Supported Transactions

Name	Protocol Ver.		Description
	Initial	Change	
Pre-Authorization	1.0		Used to validate a sale request and obtain transaction limits before performing the sale transaction.
Completion	1.0		Informs a sale initiated with a Pre-Authorization.
Completion	1.0		Informs a sale initiated with a Pre-Authorization.
Offline Completion	1.0		Informs a Completion that was authorized locally at the site.
Satellite TAG Validation	1.1		Validates a second ID sent for an already authorized transaction. Designed for acquiring a second TAG on a master-satellite pump fueling.
Contingency Completion	1.2		Informs a Completion that was authorized in contingency.
Sales	1.2		Informs a Sale.
Cancellation	1.2		Cancels a Completion or a Sale.

3 Data Security

To validate the source of transactions and data encryption, the ATIONet Native Transaction Protocol relies on a SSL connection between the Site's Terminal or Site's Controller and the ATIONet Host.

The SSL connection is established for each request/response pair, using a certificate property of ATIONet, meaning that each request must include a system-type user and password on the Header. The user will be matched against the related ATIONet actor for each message.

Users to be used on the Transaction Protocol messaging will be created by authorized users via ATIONet Console, with the role “Controller/Terminal”.

At this time there is no provisioning to distribute or update certificates or thumbprints thru a system interface. This information will be provided at request of the Controller’s vendor during the integration project.

4 Message Structure

All Transaction API messages share the same structure, what change from message to message are the Transaction Code, which indicates the actual transaction function, the value fields sent and received, and the HTTP action (POST, GET, REQUEST) which changes depending on the Transaction Code.

Both, requests and responses use a JSON format.

Only one request is accepted on each message.

Request Format

Header:

```
Accept-Encoding: gzip

Authorization: Basic user:password
```

Body:
{“Fieldname”:”StringValue”,”FieldName”:”StringValue”,”FieldName”:Value}

Response

Header:

```
Content-Type: application/json; charset=utf-8
```

Body:
{“Fieldname”:”StringValue”,”FieldName”:”StringValue”,”FieldName”:Value}

Note: Alphanumeric fields, stated as Type “A/N” in record format tables below show the maximum possible length as the Size, although in JSON-formatted strings they will be represented with

trailing spaces trimmed.

5 Error Handling

Success/failure exits on the Native Transaction Protocol will be handled via HTTP status codes.

Successful request will get a HTTP 200 and the resulting response.

Transactions intended to post a command, for example Authorizations and Pre-Authorizations will return a single JSON-formatted item with the “Response Code” and “Response Text”. The body of these responses will never be empty.

Failure to process the request will be indicated by an HTTP 400’s range status code. The body will contain a single JSON-formatted item with the “ResponseCode”, “ResponseMessage” and “ResponseError” fields.

Refer to Response Codes Table in the Reference Tables section for a complete list of supported codes.

6 Field Descriptions

This section details the purpose and expected behavior on the Controller system for relevant items on the protocol.

System Model and System Version

Brand/Model and Firmware/Software version of the client system. Format and content will be assigned for each vendor during the Integration project.

Pump Authorization Values

Pre-sale Authorizations processed by ATIONet might be (a) Fully-authorized, (b) Partially-authorized, or (c) Declined. Full and partial authorizations may have the same Authorized codes, but a partial authorization only allows to sale a fraction of the requested amount. Controllers must always check the Authorized Amount/Quantity on approved transactions.

The second value relevant for the authorization value is the local authorization limit that can be enforced locally. On any case, the preset sent to the gas pump must be the lesser value between the Authorized Amount/Quantity and the DCR Cutoff Amount.

Terminal Identification

Terminal ID is a system-wide unique ID for the Controller or Terminal device on the capture side. Terminal ID should be configured on the client system during manual installation. The length of the TID's code depends on the controller.

Device Type Identifier

A single digit field, informed by the Controller system, that identifies the type of capture device. (Manned/Unmanned, Indoor/Outdoor). In case the Controller system doesn't have the capability to inform this distinction, "4 – Other self-service" should be always informed.

Transaction Sequence Number

The Transaction number is a fixed-length integer value from 1 to 999999 and it is assigned and incremented for each transaction sent to the Host, regardless of the result. It must be reset back to 1 every time it reaches the limit.

Entry Method

The Entry Method code indicates whether the customer identification was manually typed-in, read from a card swipe or any other automatic identification mechanism.

Processing Mode

Indicates whether the Host must apply an alternative process to the request. Regular transactions must inform "1 = Host processing required"

Track Data

This field identifies the account of the transaction.

Track Data field must contain the whole identifier's information (for example, complete Track 2 data on a magnetic card, or all the TAG's fields on an AVI capture).

There are two Tracks and PINs field pairs, the Primary and Secondary, on the Primary, the Controller must send the track of the identifier used to initiate the capture transaction (first card presented) and the Secondary should contain the complementary identification (if used). For example if the transaction is initiated presenting a Driver Card that also requires a Vehicle Identification, Primary will be the Driver and Secondary will be the Vehicle.

The Primary identification will mandate the sub-account to be charged for the transaction, the Secondary will be used for rules validation but will not be affected on its balance.

Batch Number

Optional information, if informed, ATIONet will use this field for report filtering and queries.

If used, data must be formatted as an 11 digit number: yyyymmddbbb. Year (4 digits), Month (2 digits), Day (2 digits), Batch/Shift number (3 digits, padded with zeros). Date part must be the starting date of the batch. Batch number must wrap-around to 1 after reaching 999.

If there is no batch functionality at all, the recommended format is Transaction Date plus 3 zeros.

Shift Number

Optional information, if informed, ATIONet will use this field for report filtering and queries.

The Controller application can manage the Shift Number and meaning as needed. It may be day's shift number, weekly batches, split-batches, etc., although this is a fixed length field, therefore the format must be maintained.

If there is no batch or shift functionality at all, the recommended format is the business date of the transaction followed of 3 zeros.

Product Fields

Transaction messages include a list of Product fields, plus a Product Taxes compound field plus and a Product Data compound field, the later will be used only on multi-product transactions.

On single product messages, like a simple fueling transaction, product's amount and other details must be sent on the fields included in the main body of the message. When the sale includes more than one product, the first one must be sent on the main body and rest on the Product Data structure. Fuel presets will only work for the product in the main body; therefore, first product in the list should be the fuel sale if there is any.

Refer to Product Data Structure Table on the Reference Tables Section.

Transaction authorization and register is based on ProductAmount and ProductQuantity; taxes and net amount fields are optional and are not considered by ATIOnet during transaction processing, but those fields may be used later for billing and reporting. Therefore, although optional at the protocol level, those might be required for certain integration projects for a given market or functional scope.

ATIONet expects standard NACS product codes. Although it can also map custom product codes for each site, Host-based product mapping is not recommended due to the extra administrative burden.

Product Restrictions & Authorization limits on Fuel Transactions (FCS)

Pre-Authorization Requests support different business scenarios:

Scenario	Relevant fields values	Description

Zero Authorization \ Any product	Product Amount = 0 Product Quantity = 0 Product Code = NULL Product Unit Price = NULL or != NULL	Terminal doesn't enforce any pre-auth value and the user didn't select a specific product (or the client system doesn't have product identification capabilities). ATIONet will authorize according to: 1. If there is any product restriction the host will respond with a Product Code otherwise will echo the NULL Product code, confirming any product authorization. 2. If Product Unit Price is NULL and exists any price configuration the host will respond with a Product Unit Price otherwise will echo the Product Unit Price. 3. Establish amount and quantity limits based on rules and current account. 4. If the current account is based on quantity and the Terminal supports product authorization, the Host will limit by Product Quantity (with Product Code) 5. If the current account is based on quantity and the Terminal doesn't supports product authorization, the Host will limit by Product Amount if exists Product Unit Price, otherwise the transaction will be declined 6. If the current account is based on money and the Terminal supports amount authorization, the Host will limit by Product Amount. 7. If the current account is based on amount and the Terminal doesn't supports amount authorization, the Host will limit by Product Quantity (with Product Code) if exists Product Unit Price, otherwise the transaction will be declined
Zero Authorization \ Specific product	Product Amount = 0 Product Quantity = 0 Product Code != NULL Product Unit Price = NULL or	Open value transaction for a specific product. ATIONet will authorize according to: 1. If there is any product restriction will be validated first. 2. 2) 3) 4) 5) 6) and 7) Equal than Zero Authorization any product.

	!= NULL	
Amount Authorization	Product Amount > 0 Product Quantity = 0 Product code = NULL or != NULL Product Unit Price = NULL or != NULL	Amount requested, any or specific product. ATIONet will authorize according to: 1. If there is any product restriction the host will respond with a Product Code otherwise will echo the NULL Product code, confirming any product authorization. 2. If Product Unit Price is NULL and exists any price configuration the host will respond with a Product Unit Price otherwise will echo the Product Unit Price. 3. Establish amount limits based on rules and current account. If exists quantity limits and Product Unit Price is NULL the transaction will be declined. 4. 5) 6) and 7) Equal than Zero Authorization any product.
Quantity Authorization	Product Amount = 0 Product Quantity > 0 Product code = NULL or != NULL Product Unit Price = NULL or != NULL	Quantity requested, any or specific product. ATIONet will authorize according to: 1. If there is any product restriction the host will respond with a Product Code otherwise will echo the NULL Product code, confirming any product authorization. 2. If Product Unit Price is NULL and exists any price configuration the host will respond with a Product Unit Price otherwise will echo the Product Unit Price. 3. Establish quantity limits based on rules and current account. If exists amount limits and Product Unit Price is NULL the transaction will be declined. 4. 5) 6) and 7) Equal than Zero Authorization any product.

Transaction Amount is not relevant on Authorization requests in this version of the protocol as Dry Products sale is not yet available, on future versions, the Transaction Amount will be validated against the whole balance of the sub-account while Product figures will be evaluated against fuel product rules and restrictions.

In this version of the protocol, Transaction Amount should always match the Product Amount in Completions and Sales TREQs.

As described above, on certain situations, the Host will answer a Pre-Authorization request with an unsolicited Product Code; the Terminal must enforce such product restriction, otherwise the Completion will be declined.

With commercial and industrial system that don't control fuel price, the Controller should use a $1 unit price for all available products to
avoid potential declines due to the lack of unit price to resolve amount restrictions.

Customer Data

Customer data on a TREQ contains extra information gathered from prompts to the Cardholder or Attendant. On a TRESP, it contains the list of prompts that must be presented to the Cardholder or Attendant or a list of values to be used by the Terminal at capture, transaction or receipt printing.

Prompt elements vs. Data elements

Customer Data subfields can be Prompts or Data. Values contained on a Prompt are sent by the Host to be used by the Terminal to support the local processing of the prompt's, for example minimum and maximum odometer values. ATIONet can send and receive Data elements.

Refer to Customer Data Codes Table in the Reference Tables section for a complete list of supported field names.

Re-prompting & Dual-Card Identification

ATIONet supports variable prompt-set definition for each card-type processed by the Host, allowing collection and validation of different entries for different type of cards, eventually this will allow to enforce different set of rules on different Device types.

There are three ways to implement this functionality on the Controller site:

1. Devices with fixed-behavior.

   Embedded devices, legacy devices or any other kind of equipment that implements a fixed or locally configurable behavior. In this case
   ATIONet will adapt to the device capabilities, reducing the functionality and eventually the types of cards supported.
2. Devices depending on host-based re-prompt mechanism.

   Controllers that do not have a local feature for selective prompts will prompt the User or Attendant only after receiving a TRESP requesting additional prompts. In this situation, the controller must retry the TREQ with the requested additional information after gathering the additional data from the cardholder or attendant. Is up to the Controller to handle the special flow of screens and messaging to the Host with or without showing the initial transaction as declined. From ATIONet standpoint, a

TREQ requiring additional prompts will be considered Declined.

3. Devices with pre-configured behavior based on PDL.

   Controllers with the ability to process a full parameter download from the Host, could implement selective prompting before sending the TREQ to the Host, avoiding the need to process a double request.

   It is worth to mention that a failure to submit a required prompt in this kind of devices, will cause a permanent failure to process such type of card, except if the device also has a host-based re-prompt mechanism –as in (b) type.

Authorization Code

The Host will return the Authorization Code on all approved transactions.
On Pre-Authorization/Completions message flows, the Controller must keep the Authorization Code sent on the Pre-Authorization TRESP and send it
back to the Host on the Completion TREQ. This is a mandatory feature.

Refer to Authorization Codes Table in the Reference Tables section for a complete list of supported codes.

PIN Block

The PIN entry on plain text, when the whole message or the communication themselves are encrypted.

Original Data

Original data on a TREQ contains extra information related to the original transaction that we want to cancel. Used only in zero completions without authorization code and cancellations transactions.

Refer to Original Data Table in the Reference Tables section for a complete list of supported field names.

7 Transaction Request (TREQ) Message Format

Field Name	Size	Type	Condition	Descriptions/Field Value(s)
ApplicationType	3	string	Required	Always “FCS” Fleet Control System

ProcessingMode	1	string	Required	“0” = Host Capture Only “1” = Host Processing Required “2” = Operator Assisted Capture
MessageFormatVersion	3	string	Required	Current Host Message Version = “1.2”
TerminalIdentification	Var	string	Required	Terminal Identification
DeviceTypeIdentifier	1	string	Required	“1” = Indoor Payment Terminal “2” = Outdoor Payment Terminal “3” = Card Reader in Dispenser “4” = Other Self-Service
SystemModel	10	string	Required	Refer to System Model and System Version in Field Description section
SystemVersion	10	string	Required	Refer to System Model and System Version in Field Description section

TransactionCode	3	string	Required	Refer to Transaction Codes in Reference Tables Section
AccountType	1	string	Required	Refer to Account Types in Reference Tables Section
EntryMethod	1	string	Required	“M” Manual Entry “S” Swap Card “T” Tag read
ServiceCode	1	string	Optional	Reserved for future use
PumpNumber	2	string	Optional	“00”-“99”
ProductCode	4	string	Optional	“0”-“9999”
ProductUnitPrice	Var	decimal	Optional	xxx.xxx
ProductNetAmount	Var	decimal	Optional	xxxxxxx.xx
ProductTaxes	Var	Dictionary	Optional	<”[Tax Description]”, [Tax Value]\>

ProductAmount	Var	decimal	Optional	xxxxxxx.xx
ProductQuantity	Var	decimal	Optional	xxxxxxx.xx
TransactionNetAmount	Var	decimal	Optional	xxxxxxx.xx
ProductData	Var	List	Conditional	Refer to Product Fields in Field Description section
TransactionAmount	Var	decimal	Optional	xxxxxxx.xx
UnitCode	Var	string	Optional	Refer to Measurement Unit Codes in Reference Tables Section
CurrencyCode	3	string	Optional	Refer to Currency Codes in Reference Tables Section
BatchNumber	Var	int	Optional	Refer to Batch Number in Field Description section
ShiftNumber	Var	string	Optional	Refer to Shift Number in Field Description

				Description section
TransactionSequenceNumber	Var	int	Required	Refer to Transaction Sequence Number in Field Description section
LocalTransactionDate	8	int	Required	Local Transaction Date: yyyymmdd
LocalTransactionTime	6	int	Required	Local Transaction Time: hhmmss
PrimaryTrack	Var	string	Required	Refer to Track Data in Field Description section
PrimaryPIN	Var	string	Conditional	Refer to PIN Block in Field Description section
SecondaryTrack	Var	string	Optional	Refer to Track Data in Field Description section
SecondaryPIN	Var	string	Optional	Refer to PIN Block in Field Description

				section
CustomerData	Var	Dictionary	Conditional	Refer to Customer Data in Field Description section
TransactionExtendedData	Var	string	Optional	Designed to capture OBD extended data (On board Devices)
OriginalData	Var	Dictionary	Conditional	Refer to Original Data in Field Description section
AuthorizationCode	Var	string	Conditional	Refer to Authorization Code in Field Description section
InvoiceNumber	Var	string	Optional	
ResponseCode	5	string	Conditional	Use only when informing a Response not created by the Host (for example a local authorization), otherwise it should not be echoed from

				TRESP
ResponseText	20	string	Conditional	Idem Response Code

8 Transaction Response (TRESP) Message Format

Field Name	Size	Type	Condition	Descriptions/Field Value(s)
ApplicationType	3	string	Required	Echoed from TREQ
ProcessingMode	1	string	Required	Echoed from TREQ
MessageFormatVersion	3	string	Required	Echoed from TREQ
TerminalIdentification	Var	string	Required	Echoed from TREQ
DeviceTypeIdentifier	1	string	Required	Echoed from TREQ
TransactionCode	3	string	Required	Refer to Transaction Codes in Reference Tables Section

AccountType	1	string	Required	Echoed from TREQ
EntryMethod	1	string	Required	Echoed from TREQ
PumpNumber	2	string	Optional	Echoed from TREQ
ProductCode	4	string	Conditional	Refer to Product Fields in Field Description section
ProductUnitPrice	Var	decimal	Conditional	xxx.xxx
ProductAmount	Var	decimal	Conditional	xxxxxxx.xx
ProductQuantity	Var	decimal	Conditional	xxxxxxx.xx
ProductData	Var	List	Conditional	Refer to Product Fields in Field Description section
TransactionAmount	Var	decimal	Conditional	xxxxxxx.xx
UnitCode	Var	string	Optional	Refer to Measurement Unit Codes in

				Reference Tables Section
CurrencyCode	3	string	Optional	Refer to Currency Codes in Reference Tables Section
BatchNumber	Var	int	Optional	Echoed from TREQ
ShiftNumber	Var	string	Optional	Echoed from TREQ
TransactionSequenceNumber	Var	int	Required	Echoed from TREQ
LocalTransactionDate	8	int	Required	Echoed from TREQ
LocalTransactionTime	6	int	Required	Echoed from TREQ
CustomerData	Var	Dictionary	Conditional	Refer to Customer Data in Field Description section
AuthorizationCode	Var	string	Conditional	Refer to Authorization Code in Field

				Description section
InvoiceNumber	Var	string	Optional	
ReceiptData	Var	string	Conditional	
ResponseCode	5	string	Required	“0” = Authorized, !”0” = Not Authorized
ResponseText	20	string	Required	Message from the Network

9 Satellite TAG Validation Request (VREQ) Message Format

Field Name	Size	Type	Condition	Descriptions/Field Value(s)
ApplicationType	3	string	Required	Always “FCS” Fleet Control System
ProcessingMode	1	string	Required	“0” = Host Capture Only “1” = Host processing required
MessageFormatVersion	3	string	Required	Current Host Message Version = “1.2”
TerminalIdentification	Var	string	Required	
TransactionCode	3	string	Required	Refer to Transaction Codes in Reference Tables Section
LocalDate	8	int	Required	Local Transaction Date: yyyymmdd
LocalTime	6	int	Required	Local Transaction Time: hhmmss
TagId1	Var	string	Required	First TAG’s ID or Secure ID
TagId2	Var	string	Required	Second TAG’s ID or Secure ID
AuthorizationCode	Var	string	Required	Auth code received for the ongoing transaction

10 Satellite TAG Validation Response (VRESP) Message Format

Field Name	Size	Type	Condition	Descriptions/Field Value(s)
ApplicationType	3	string	Required	Echoed from VREQ
ProcessingMode	1	string	Required	Echoed from VREQ
MessageFormatVersion	3	string	Required	Echoed from VREQ
TerminalIdentification	Var	string	Required	Echoed from VREQ
TransactionCode	3	string	Required	Refer to Transaction Codes in Reference Tables Section
LocalDate	8	int	Required	Echoed from VREQ
LocalTime	6	int	Required	Echoed from VREQ
AuthorizationCode	Var	string	Required	Echoed from VREQ
ResponseCode	5	string	Required	“0” = Authorized, !”0” = Not Authorized
ResponseText	20	string	Required	Message from the Network

11 Reference Tables

This section brings together the code tables and reference values used in messaging.

11.1 Transaction Codes

Code	Message	Description
“100”	TREQ	Pre-Authorization REQ
“101”	VREQ	Satellite TAG Validation REQ
“110”	TRESP	Pre-Authorization RESP
“111”	VRESP	Satellite TAG Validation RESP
“120”	TREQ	Completion REQ
“125”	TREQ	Offline REQ
“126”	TREQ	Contingency REQ
“130”	TRESP	Completion RESP
“200”	TREQ	Sale REQ
“210”	TRESP	Sale RESP
“400”	TREQ	Cancellation REQ
“410”	TRESP	Cancellation RESP

11.2 Account Type

Type	Description
“1”	ATIONet native track

11.3 Product Data Structure

Field Name	Size	Type	Condition	Descriptions/Field Value(s)
ServiceCode	1	string	Required	
ProductCode	4	string	Required	“0”-“9999”
ProductUnitPrice	Var	decimal	Optional	xxx.xxx
ProductNetAmount	Var	decimal	Optional	xxxxxxx.xx
ProductTaxes	Var	Dictionary	Optional	<”[Tax Description]”, [Tax Value]>
ProductAmount	Var	decimal	Optional	xxxxxxx.xx
ProductQuantity	Var	decimal	Optional	xxxxxxx.xx
UnitCode	Var	string	Optional	Refer to Measurement Unit Codes in Reference Tables Section

11.4 Customer Data

Prompt elements

Field Name
PromptOdometer
Last Odometer
Min Odometer
Max Odometer
PromptDriverId
PromptTruckUnitNumber
PromptTrailerNumber
PromptEngine Hours

PromptEngine Hours
Last Engine Hours
Min Engine Hours
Max Engine Hours
PromptMiscellaneous

Data elements

Field Name
TruckUnitNumber
TrailerNumber
Odometer

EngineHours
DriverId
Miscellaneous
DriverLicenseState
DriverLicenseNumber
TripNumber
PurchaseOrderNumber
ClientSupportsReceiptDownloading
TrailerHourMeterReading

TrailerHourMeterReading

11.5 Measurement Unit Codes

Value	Description
“usgal”	Gallon USA
“ukgal”	Gallon UK
“l”	Litro
“m3”	Metro Cúbico
“kg”	Kilogramo

11.6 Currency Codes

Refer to ISO 4217 Currency Codes standard (http://en.wikipedia.org/wiki/ISO_4217)

11.7 Authorization Codes

ResponseCode	ResponseMessage
00000	Authorized
Validations	
10000	Date Invalid

10001	Time Invalid
10002	Seq Num Invalid
10003	Term does not exist
10004	Netw does not exist
10005	Id does not exist
10006	SecId does not exist
10007	Fuel does not exist
10008	Merch not found
10009	Site not found
10010	Prot not found
10011	TType not found
10012	Comp not found
10013	Contr not found

10014	Subacc not found
10015	SecSubacc not found
10016	Empty subaccount
10017	Empty sec subaccount
10018	Ids both veh
10019	Ids both driv
10020	Subacc in diff cont
10021	Dri or Veh not found
10022	Id is not active
10023	SecId is not active
10024	Id has expired
10025	SecId has expired
10026	Vehicle not enabled

10027	Driver not enabled
10028	Contract has expired
10029	Site not in contr
10030	Fuel not in contr
10031	Fuel not in vehclas
10032	Driver not related
10033	Vehicle not related
10034	Sec Track needed
10035	Fuel needed
10036	Fuel mapping needed
10037	Already completed
10038	NetComp not found
10039	NetMerch not found

10040	Auth does not exists
10041	Auth not authorized
10042	Auth with diff fuel
10043	Auth with diff PPU
10044	Auth amount exceeded
10045	Auth qty exceeded
10046	Auth with diff id
10047	Auth with diff secid
10048	Auth with diff term
10049	Auth with diff netw
10050	Auth with diff merch
10051	Auth with diff nwmr
10052	Auth with diff site

10053	Auth with diff prot
10054	Auth with diff tt
10055	Auth with diff comp
10056	Auth with diff nwcp
10057	Auth with diff contr
10058	Auth with diff subacc
10059	Auth with diff sec sa
10060	Auth with diff vehicle
10061	Auth with diff driver
10062	Proc Code Not Supp
10063	TType qty exceded
10064	TType amount exceded

10065	Tag PIN Invalid
LocationRule	
40100	Site not authorized
40101	Site not authorized
40102	Site not authorized
40103	Site not authorized
FuelRule	
40200	Product not authorized
40201	Product not authorized
40202	Product not authorized
40203	Product not authorized
TransactionRule	
20300	Quota not set

40300	Veh money excedeed
40301	Driv money excedeed
40302	Prod money excedeed
40303	Site money excedeed
40304	Fleet money excedeed
40305	Veh fuel excedeed
40306	Driv fuel excedeed
40307	Prod fuel excedeed
40308	Site fuel excedeed
40309	Fleet fuel excedeed
QuotaRule	
20400	Quota not set
40400	Veh money excedeed

40401	Driv money excedeed
40402	Prod money excedeed
40403	Site money excedeed
40404	Fleet money excedeed
40405	Veh fuel excedeed
40406	Driv fuel excedeed
40407	Prod fuel excedeed
40408	Site fuel excedeed
40409	Fleet fuel excedeed
40410	Veh tran excedeed
40411	Driv tran excedeed
40412	Prod tran excedeed
40413	Site tran excedeed

40414	Fleet tran excedeed
PromptingRule	
20500	Retries exceded
40500	Prompting needed
40501	Pri PIN needed
40502	Sec PIN needed
40503	Pri PIN invalid
40504	Sec PIN invalid
DaysRule	
20600	Week days not set
40600	Day not authorized
40601	Day not authorized
40602	Day not authorized

40603	Day not authorized
40604	Day not authorized
DateTimeRule	
20700	DateTime not set
40700	DateTime not auth
40701	DateTime not auth
40702	DateTime not auth
40703	DateTime not auth
40704	DateTime not auth
40705	DateTime not auth
40706	DateTime not auth
40707	DateTime not auth

40708	DateTime not auth
40709	DateTime not auth
40710	DateTime not auth
40711	DateTime not auth
40712	DateTime not auth
40713	DateTime not auth
40714	DateTime not auth
40715	DateTime not auth
40716	DateTime not auth
40717	DateTime not auth
40718	DateTime not auth
40719	DateTime not auth
40720	DateTime not auth

40721	DateTime not auth
40722	DateTime not auth
40723	DateTime not auth
40724	DateTime not auth
40725	DateTime not auth
40726	DateTime not auth
40727	DateTime not auth
40728	DateTime not auth
40729	DateTime not auth
DaysTimeRule	
20800	Week days not set
20801	Time not set
40800	Day not authorized

40801	Day not authorized
40802	Day not authorized
40803	Day not authorized
40804	Day not authorized
40805	DaysTime not auth
40806	DaysTime not auth
40807	DaysTime not auth
40808	DaysTime not auth
40809	DaysTime not auth
40810	DaysTime not auth
40811	DaysTime not auth
40812	DaysTime not auth
40813	DaysTime not auth

40814	DaysTime not auth
EstablishLimits	
20900	Unit price needed
20901	Max quota not set
40900	CA quota exceeded
40901	Offline lim exceeded
Warnings	
30000	Pim Track not match
30001	Sec Track not match
30002	Fuels not match
30003	PPU not match
AplicationError	
50000	App Error

11.8 Response Codes

ResponseCode	ResponseMessage
00000	Operation Succeeded
40000	Invalid Identification Data
40001	Invalid Filter Data
40002	User not allowed to use this action
40003	Invalid Action Code
40004	Invalid user name or password
40005	Movement not allowed
50000	Internal Server Error

11.9 Original Data

Field Name
TransactionCode
TransactionSequenceNumber
LocalTransactionDate
LocalTransactionTime