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
1.3	5/Dec/2017	1.3 Protocol version changes - Update Response Codes table

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 \	Product Amount = 0	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).

Any product	Product Quantity = 0 Product Code = NULL Product Unit Price = NULL or != NULL	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 != NULL	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.
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
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
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	xxxxxx.xx
ProductTaxes	Var	Dictionary	Optional	<”[Tax Description]”, [Tax Value]\>
ProductAmount	Var	decimal	Optional	xxxxxx.xx
ProductQuantity	Var	decimal	Optional	xxxxxx.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 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
LongResponseText	200	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	xxxxxx.xx
ProductQuantity	Var	decimal	Conditional	xxxxxx.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	
ResponseCode	5	string	Required	“0” = Authorized, !”0” = Not Authorized
ResponseText	20	string	Required	Message from the Network
ReceiptData	Var	string	Conditional	
LongResponseText	200	string	Conditional	

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	xxxxxx.xx
ProductTaxes	Var	Dictionary	Optional	<”[Tax Description]”, [Tax Value]>
ProductAmount	Var	decimal	Optional	xxxxxx.xx
ProductQuantity	Var	decimal	Optional	xxxxxx.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
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

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

Code	Response Message	Long Response Message	Description
Authorized			
00000	Authorized	Authorized	
Request Validations			
10000	Invalid date	Invalid date	
10001	Invalid time	Invalid time	
10002	Invalid seq num	Invalid sequence number	
10003	Invalid acc type	Invalid account type	
10004	Invalid app type	Invalid application type	
10005	Invalid proc mode	Invalid processing mode	

10006	Invalid mess format	Invalid message format	
10007	Invalid dev type	Invalid device type	
10008	Invalid sys model	Invalid system model	
10009	Invalid sys ver	Invalid system version	
10010	Invalid entry method	Invalid entry method	
10011	Invalid unit code	Invalid unit code	
10012	Invalid unit code	Invalid datetime	
10013	Invalid pri track	Invalid primary track	
10014	Invalid prod data	Invalid product data	
10015	Prod data req	Product data required	
10016	Invalid batch number	Invalid batch number	
10017	Invalid respone code	Invalid respone code	
10018	Invalid terminal	Invalid terminal	
10019	Invalid old PIN	Invalid old PIN	
10020	Invalid new PIN	Invalid new PIN	
10021	Invalid orig data	Invalid original data	
Integrity Validations			
11000	Merch not found	Merchant not found	

11001	NetMerch not found	Network merchant not found	
11002	Site not found	Site not found	
11003	Prot not found	Protocol not found	
11004	TType not found	Terminal type not found	
11005	Fuel mapping needed	Fuel mapping needed	
11006	Fuel not found	Fuel not found	
11007	Comp not found	Company not found	
11008	NetComp not found	Network company not found	
11009	Contr not found	Contract not found	
11010	Subacc not found	Sub account not found	
11011	SecSubacc not found	Secondary sub account not found	
11012	Veh or dri not found	Vehicle or driver not found	
11013	Empty subaccount	Empty sub account	
11014	Empty sec subaccount	Empty secondary sub account	
11015	Fuel needed	Fuel needed	
11016	Fuel not in contr	Fuel not in contract	
11017	Fuel not in vehclas	Fuel not in vehicle class	
11018	Comp money needed	Company money needed	
11019	Merch money needed	Merchant money needed	

11020	Comp qty needed	Company quantity needed	
11021	Merch qty needed	Merchant quantity needed	
11022	Fleet not found	Fleet not found	
11023	Trans not found	Transaction not found	
To Review Validations			
12000	Auth amount exceeded	Authorized amount exceeded	
12001	Auth qty exceeded	Authorized quantity exceeded	
12002	Auth with diff PPU	Authorization with different PPU	
12003	Tr amount exceeded	Transaction amount exceeded	
Unlawful Validations			
13000	Term does not exist	Terminal does not exist	
13001	Netw does not exist	Network does not exist	
13002	Id does not exist	Id does not exist	
13003	SecId does not exist	Secondary Id does not exist	
13004	Both veh ids	Both are vehicle identifications	
13005	Both driv ids	Both are driver identifications	
13006	Subacc in diff cont	Sub account in different contract	
13007	Id is not active	Identification is not active	
13008	Id has expired	Identification has expired	

13009	SecId is not active	Secondary identification is not active	
13010	SecId has expired	Secondary identification has expired	
13011	Vehicle not enabled	Vehicle not enabled	
13012	Driver not enabled	Driver not enabled	
13013	Contract has expired	Contract has expired	
13014	Site not in contr	Site not in contract	
13015	Driver Needed	Driver needed	
13016	Driver not related	Driver not related	
13017	Vehicle Needed	Vehicle needed	
13018	Vehicle not related	Vehicle not related	
13019	Duplicate TSN	Duplicate TSN	
13020	Retry does not match	Retry does not match	
13021	Auth does not exist	Authorization does not exist	
13022	Auth not authorized	Authorization not authorized	
13023	Auth with diff secid	Authorization with different secondary id	
13024	Auth with diff id	Authorization with different id	
13025	Auth with diff fuel	Authorization with different fuel	
13026	Auth with diff term	Authorization with different terminal	
13027	Auth with diff netw	Authorization with different network	

13028	Auth with diff merch	Authorization with different merchant	
13029	Auth with diff nwmr	Authorization with different network-merchant	
13030	Auth with diff site	Authorization with different site	
13031	Auth with diff prot	Authorization with diff protocol	
13032	Auth with diff tt	Authorization with different terminal type	
13033	Auth with diff comp	Authorization with different company	
13034	Auth with diff nwcp	Authorization with diff network-company	
13035	Auth with dif contr	Authorization with different contract	
13036	Auth with diff subacc	Authorization with different sub account	
13037	Auth with dif sec sa	Authorization with different secondary sub account	
13038	Auth with diff vehicle	Authorization with different vehicle	
13039	Auth with diff driver	Authorization with different driver	
13040	Not contingency Auth	Not a contingency authorization	
13041	Invalid driver code	Invalid driver code	
13042	Invalid vehicle code	Invalid vehicle code	
13043	Network not enabled	Network not enabled	
13044	Company not enabled	Company not enabled	
13045	Merchant not enabled	Merchant not enabled	

13046	Invalid sec ident	Invalid secondary identification	
13047	Invalid sec ident	Invalid secondary identification	
Current Account Validations			
14000	CA movement declined	Current account movement declined	
14001	Merchant CA mov dec	Merchant current account movement declined	
Establish Limits			
20000	Unit price needed	Unit price needed	
20001	Max quota not set	Maximum quiota not set	
40000	Insufficient balance	Insufficient balance	
40001	Unable prods auth	Unable to authorize by product	
40002	Veh-class qty exc	Vahicle class tank capacity excedeed	
40003	TType qty exc	Terminal type quantity excedeed	
40004	Term qty excedeed	Terminal quantity excedeed	
40005	Term money excedeed	Terminal money excedeed	
40006	Offline lim excedeed	Offline limit excedeed	
Location Rule			
40100	Site not authorized	Site not authorized	
40101	Site not authorized	Site not authorized	
40102	Site not authorized	Site not authorized	

40103	Site not authorized	Site not authorized	
Fuel Rule			
40200	Fuel not authorized	Fuel not authorized	
40201	Fuel not authorized	Fuel not authorized	
40202	Fuel not authorized	Fuel not authorized	
40203	Fuel not authorized	Fuel not authorized	
Transaction Rule			
20300	Quota not set	Quota not set	
40300	Veh money excedeed	Vehicle money excedeed	
40301	Driv money excedeed	Driver money excedeed	
40302	Fuel money excedeed	Fuel money excedeed	
40303	Site money excedeed	Site money excedeed	
40304	Fleet money excedeed	Fleet money excedeed	
40305	Veh qty excedeed	Veh quantity excedeed	
40306	Driv qty excedeed	Driv quantity excedeed	
40307	Fuel qty excedeed	Fuel quantity excedeed	
40308	Site qty excedeed	Site quantity excedeed	
40309	Fleet qty excedeed	Fleet quantity excedeed	
Quota Rule			

20400	Quota not set	Quota not set	
40400	Veh money excedeed	Vehicle money excedeed	
40401	Driv money excedeed	Driver money excedeed	
40402	Fuel money excedeed	Fuel money excedeed	
40403	Site money excedeed	Site money excedeed	
40404	Fleet money excedeed	Fleet money excedeed	
40405	Veh qty excedeed	Veh quantity excedeed	
40406	Driv qty excedeed	Driv quantity excedeed	
40407	Fuel qty excedeed	Fuel quantity excedeed	
40408	Site qty excedeed	Site quantity excedeed	
40409	Fleet qty excedeed	Fleet quantity excedeed	
40410	Veh tran excedeed	Vehicle transactions excedeed	
40411	Driv tran excedeed	Driv transactions excedeed	
40412	Prod tran excedeed	Prod transactions excedeed	
40413	Site tran excedeed	Site transactions excedeed	
40414	Fleet tran excedeed	Fleet transactions excedeed	
Prompting Rule			
20500	Retries exceeded	Retries exceeded	
40500	Prompting needed	Prompting needed	

40501	Pri PIN needed	Primary PIN needed	
40502	Sec PIN needed	Secondary PIN needed	
40503	Invalid pri PIN	Invalid primary PIN	
40504	Invalid sec PIN	Invalid secondary PIN	
40505	Sec Track needed	Secondary track needed	
40506	Invalid odometer	Invalid odometer	
40507	Invalid eng hours	Invalid engine hours	
Days Rule			
20600	Week days not set	Week days not set	
40600	Day not authorized	Day not authorized	
40601	Day not authorized	Day not authorized	
40602	Day not authorized	Day not authorized	
40603	Day not authorized	Day not authorized	
40604	Day not authorized	Day not authorized	
DateTime Rule			
20700	DateTime not set	DateTime not set	
40700	DateTime not auth	DateTime not auth	
40701	DateTime not auth	DateTime not auth	

40702	DateTime not auth	DateTime not auth	
40703	DateTime not auth	DateTime not auth	
40704	DateTime not auth	DateTime not auth	
40705	DateTime not auth	DateTime not auth	
40706	DateTime not auth	DateTime not auth	
40707	DateTime not auth	DateTime not auth	
40708	DateTime not auth	DateTime not auth	
40709	DateTime not auth	DateTime not auth	
40710	DateTime not auth	DateTime not auth	
40711	DateTime not auth	DateTime not auth	
40712	DateTime not auth	DateTime not auth	
40713	DateTime not auth	DateTime not auth	
40714	DateTime not auth	DateTime not auth	
40715	DateTime not auth	DateTime not auth	
40716	DateTime not auth	DateTime not auth	
40717	DateTime not auth	DateTime not auth	
40718	DateTime not auth	DateTime not auth	
40719	DateTime not auth	DateTime not auth	
40720	DateTime not auth	DateTime not auth	

40721	DateTime not auth	DateTime not auth	
40722	DateTime not auth	DateTime not auth	
40723	DateTime not auth	DateTime not auth	
40724	DateTime not auth	DateTime not auth	
40725	DateTime not auth	DateTime not auth	
40726	DateTime not auth	DateTime not auth	
40727	DateTime not auth	DateTime not auth	
40728	DateTime not auth	DateTime not auth	
40729	DateTime not auth	DateTime not auth	
DaysTime Rule			
20800	Week days not set	Week days not set	
20801	Time not set	Time not set	
40800	Day not authorized	Day not authorized	
40801	Day not authorized	Day not authorized	
40802	Day not authorized	Day not authorized	
40803	Day not authorized	Day not authorized	
40804	Day not authorized	Day not authorized	
40805	DaysTime not auth	Day not authorized	
40806	DaysTime not auth	Day not authorized	

40807	DaysTime not auth	Day not authorized	
40808	DaysTime not auth	Day not authorized	
40809	DaysTime not auth	Day not authorized	
40810	DaysTime not auth	Day not authorized	
40811	DaysTime not auth	Day not authorized	
40812	DaysTime not auth	Day not authorized	
40813	DaysTime not auth	Day not authorized	
40814	DaysTime not auth	Day not authorized	
Dry Transaction Rule			
20900	Quota Not Set	Quota not set	
40900	Veh money exceeded	Vehicle money exceeded	
40901	Driv money exceeded	Driver money exceeded	
40902	Site money exceeded	Site money exceeded	
40903	Fleet money exceeded	Fleet money exceeded	
Dry Quota Rule			
21000	Quota not set	Quota not set	
41000	Veh money exceeded	Vehicle money exceeded	
41001	Driv money exceeded	Driver money exceeded	
41002	Site money exceeded	Site money exceeded	

41003	Fleet money exceeded	Fleet money exceeded	
Process Messages			
50000	App error	Application error	
50001	Not available in off	Not available in offline mode	

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